Oxford International Lower Secondary

8

Computing

Student Book

OXFORD

Contents

Introduction

Delivering computing to young learners

Oxford International Primary and Lower Secondary Computing is a complete syllabus for computing education for ages 5–14 (Years 1–9). By following the program of learning set out in this series, teachers can feel reassured that their students have access to the computing skills and understanding that they need for their future education.

Find out more at:
www.oxfordsecondary.com/computing.

Structure of the book

This book is divided into six chapters, for Year 8 (ages 12–13).

1. **The nature of technology:** Introduction to computer networks and internet services
2. **Digital literacy:** Using online research for learning and discovery
3. **Computational thinking:** Writing a program using a data structure
4. **Programming:** Using procedures or functions and comparing algorithms
5. **Multimedia:** Using technology creatively to make a video
6. **Numbers and data:** Using technology to analyse data

What you will find in each unit

- Introduction: An unplugged activity and a class discussion help students to start thinking about the topic.
- Lessons: Six lessons guide students through activity-based learning.
- Check what you know: A test and activities allow you to measure students' progress.

What you will find in the lessons

Although each lesson is unique, they have common features: learning outcomes for each lesson are set out at the start; learning content delivers skills and develops understanding.

Activity Every lesson involves a one or more learning activities for the students.

Extra challenge Activities to extend students who are able to do more.

Test A short test of four questions, of progressive difficulty, to check students' understanding of the lesson.

Additional features

You will also find these features throughout the book:

Word cloud The word cloud builds vocabulary by identifying key terms from the unit.

Be creative Suggestions for creative and artistic work.

Explore more Extra tasks that can be taken outside the classroom and into the home.

Digital citizen of the future Advice on using computers responsibly in life.

Glossary Key terms are identified in the text and defined in the glossary at the end.

Assessing student achievement

The final pages in each unit give an opportunity to assess student achievement.

- Developing: This acknowledges the achievement of students who find the content challenging but have made progress.
- Secure: Students have reached the level set out in the programme for their age group. Most should reach this level.
- Extended: This recognises the achievement of students who have developed above-average skills and understanding.

Questions and activities are colour-coded according to achievement level. Self-evaluation advice helps students to check their own progress.

Software to use

We recommend Python for writing programs at this age. For other lessons, teachers can use any suitable software, for example: Microsoft Office; Google Drive software; LibreOffice; any web browser.

Source files

You will see this symbol on some of the pages.

This means that there are extra files you can access to help with the learning activities. For example, half-completed Python programs or spreadsheet files.

To access the files, go to **www.oxfordowl.co.uk** and navigate to the 'Oxford International Secondary Programme' page then 'Oxford International Secondary Computing'.

Teacher's Guides

For more on these topics, look at the Teacher's Guide that accompanies this book.

1 The nature of technology: Understanding networks

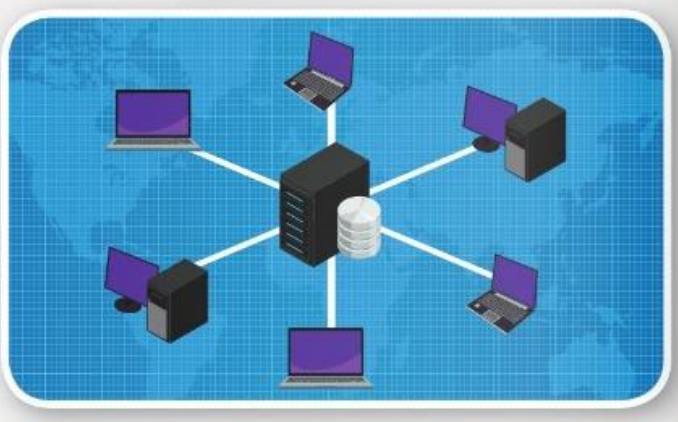
You will learn

- about network hardware and how the components work together
- how messages are sent over the internet
- about packet switching
- how to connect to a network and how to solve problems with network connections
- about cloud storage and other services in the cloud.

In Student Book 5 you learned how to look for clues that tell you there is a network in your school. You learned that networks use special hardware such as hubs and routers. In this unit you will learn more about networks. You will learn about the hardware components of a network and how they are connected. You will learn how messages are sent over the internet. You will also learn how to connect to a network and how to solve some of the problems you might experience along the way. Finally, you will learn about cloud storage and other services in the cloud.

Unplugged

What do you know about network connections? Look around the room you are in and make a note of everything you can see that tells you there is a network in the room. If there is time, take pictures or make drawings.



Learning outcomes: Explain how computers communicate; Describe internet services (for example, cloud storage)

network hub
switch server router packets
packet switching cloud
data centre wireless access point

Talk about...

In 2019, more than half (around 56%) of the world's population had access to the internet.

The number of people with access to the internet varies in different parts of the world. The rates are high in Europe, North America and parts of Asia. In parts of Africa, access to the internet is still very low.

Is it fair that some people are unable to access the internet? What do you think people miss out on? Is there anything the rest of the world could do to make the internet easier to access?

Did you know?

A new improved mobile phone service called 5G was introduced in 2019. 5G provides an internet connection that is faster than home internet links that run over cables. Some people think that 5G phone networks will become the way that we access the internet at home in the future.

1.1 Network hardware

In this lesson

You will learn:

- about the hardware needed in a network.

Spiral back

In Student Book 5 you learned about two types of network. There are local area networks, like the one in your school, and wide area networks like the internet. Networks use special equipment. You learned how to look for clues that tell you there is a network in your school.

Using networks

When you use a computer, you are usually using a **network**.

You may use your school network. If you are at home, you may have a **broadband network** that connects you to the internet.

When you are sending an email, saving a school assignment or posting your thoughts on a social media site, you use a network. You often don't think about using a network. You save a document and expect it to be available next time you need it. But what happens to the file and where does it go?

When we send or receive a message over a network, we use many pieces of computer hardware. In this lesson you will learn more about network hardware.

Network hardware

Servers and storage

One of the most important pieces of hardware in a network is a **server**. A server is a computer.

The clue to what a server does is in the name. When you go to a restaurant, a waiter serves you food. The job of a waiter is to make sure everyone gets the meal they ordered.

A server stores and sends computer files and messages to users and to devices like printers. The job of a server is to make sure files go to the person or device that they are intended for.

Here are some examples of what a server can do.

It can:

- receive a word-processed file and save it to a network storage drive
- find a spreadsheet file you ask for and send it to your computer
- receive an email from you and pass it on to someone else
- receive a request from you to print a file and send the file to a printer.

Different types of servers do different jobs. For example, there are file servers, print servers and email servers.

Activity

What type of server is used for each of the examples?

What job do you think a web server does?

All computers need to store data. Servers are no different. They have storage devices attached to them. The storage devices used by servers are larger and faster than the ones in personal computers. This is because **network storage** is shared by many people who use the network.

Hubs and switches

When messages are sent out across a network, they need to be sent to the correct place. Hubs and switches do this job. They work in different ways.

When a **hub** receives a message, it sends it to all the computers that are connected to it. Every computer checks the message. The computer that the message is meant for keeps the message. The other computers ignore the message.

When a **switch** receives a message, it decides which computer the message is meant for. It sends the message only to that computer.

So, for example, if a hub is connected to 20 devices, it sends a message to all 20 devices. A switch only sends the message to the device it is meant for.

Hubs and switches: advantages and disadvantages

When a network is busy, the extra messages a hub sends out can slow the network down. A hub is also less secure than a switch. A hacker has more chance of stealing messages.

A switch is more expensive than a hub and is more difficult to set up. But it is more secure and works faster.

Activity

Imagine you work in a computer department. Your manager sends you this email.

I am planning to upgrade the network in our accounts department. The computer manager wants to use a switch for the network. I have been told a hub would be cheaper. What is your advice?

Write a short reply to the email.

Routers and modems

A **router** joins two networks together. Routers are usually used to connect a LAN or a home network to the internet.

When two networks are joined together, they sometimes use different ways to send data. It is like two people talking in different languages. They can communicate if a translator works with them. A translator understands both languages. A translator changes words from one language to the other. This is the job of a **modem** in a network. It takes data sent by one network and translates it into a form that can be understood by the second network.

A wireless access point

Most modern networks allow wireless connection. This means that devices like laptops and tablets can connect to the network without having to use a cable. A wireless connection needs a special piece of hardware called a **wireless access point** (WAP).

A WAP is connected to the network by a cable. But computers can connect to the WAP wirelessly, without a cable. Several computers can connect to a single WAP. WAPs are put on ceilings or high on walls.

A network interface card

A **network interface card** (NIC) allows a computer or other device to connect to a network. Most devices come with a NIC fitted.

A standard desktop or laptop computer will be fitted with two NICs.

- One allows a network cable to be plugged in.
- The other allows a wireless connection.

NICs are fitted inside the computer case but you will see the network cable socket on the outside of the case.

Activity

Look for network hardware at school. Take photos of the hardware you find.

Make a document and label the photos.

If you have a home network, take a photo of your home router and add it to the document.

Extra challenge

Search the web for images of the types of network hardware described in this lesson.

Paste the images at the end of the document you created in the activity. Label them.

Test

1. What does a server do in a network?
2. What piece of network hardware is used to give a wireless connection?
3. How do a router and modem work together in a network?
4. Explain why you would use a switch in a network instead of a hub.

1.2 How a network fits together

In this lesson

You will learn:

- how the hardware components of a network are connected
- how the hardware components work together.

Home network

Many people have internet connections at home. An internet signal reaches your home through telephone or special broadband cables. When the internet connection reaches your home, it is connected to a device called a home router.

In Lesson 1.1 you learned about the devices used in a network. A home router combines the main devices needed for a network into a single small case that sits on a shelf or desk. The small home router contains four main components.

- A **router** connects your home to the internet.
- A **modem** converts the signals that pass along telephone or broadband cables into digital data your home network can use.
- A **switch** makes sure that messages coming into your home are sent to the correct computer.
- A **wireless access point** lets you connect to the network wirelessly, from anywhere in your home.

The photo shows the back of a home router. The yellow cable links the internet to the router. The four empty sockets allow devices to be plugged into the router using a cable. The sockets are sometimes used to connect other devices like a printer.

The three aerials (which look like black sticks in this photo) are connected to the wireless access point. Not all WAPs have aerials, but aerials can improve the wireless signal. Most people connect computers to the router using a wireless connection.

Activity

Which components of a network listed in Lesson 1.1 are not included in a home router? List them.

Explore more

If you have a home broadband connection, examine the home router carefully. Can you identify the features described in this lesson?

Don't touch or remove any cables. It is an electrical device so can be dangerous.

Local area network

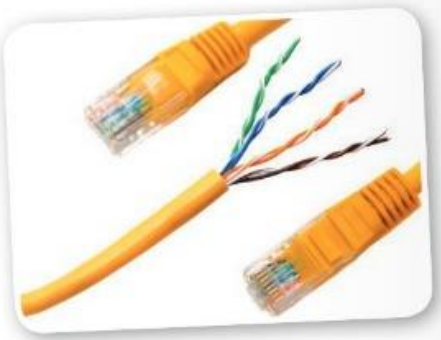
A **local area network** (LAN) is much bigger than a home network, but it contains the same components. The components in a LAN are larger and there are many more of them. A LAN is not stored in a single case like a home network. LAN equipment is spread around a building. The equipment is stored in cabinets that either stand on the floor or are attached to walls.

Connecting LAN components

Network components like servers and switches are connected by cables.

Two types of cable connect network equipment.

- **Copper cable** is the most common cable in a network. Networks use **twisted pair cables**, made up of pairs of thin copper wires twisted together. Data is sent along copper cables as pulses of electricity.
- **Fibre-optic cable** is made up of thin strands of clear fibre. Data is sent along **fibre-optic cables** as pulses of light.



There are three important differences in the two types of cable.

1. More data can be sent at faster speed along fibre-optic cables. Data is sent as light. Nothing travels faster than light.
2. Fibre-optic cables can be used over longer distances than copper. Electrical pulses fade as they travel. Copper cables can only be used over distances of 100 metres.
3. It is cheaper to use copper cable.

The server room

At the centre of a network is a **server room**. The room is usually air conditioned. This is because computer equipment makes a lot of heat. If it gets too hot, the equipment may break down.

The server room contains all the servers in the network. It also contains hubs and switches so that the rest of the network can be connected to the servers.

All servers must be connected to a hub or switch. Data passes along a single cable between the server and a hub or switch. All devices must be connected to a hub or switch. A single hub or switch can connect to many devices. Each connection uses its own cable.

When a computer is plugged into a network socket, it is connected to the network. It sends messages along a cable to the hub or switch. The hub or switch passes the messages from many devices to and from a server.

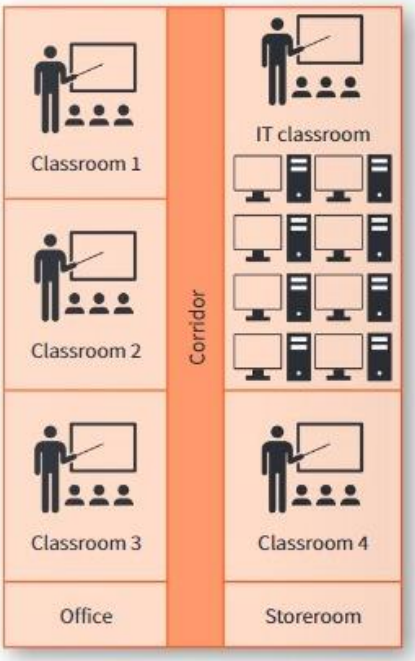
The hub room

Sometimes you need to connect an area of a building to a network that is more than 100 metres away from the server room. This can't be done with copper cable. The electrical signal won't transmit that far. So a **hub room** is set up close to the area that is going to be connected to the network.

A hub room does not contain servers. It only has switches and hubs. Every network socket in the distant area is connected to a hub or switch in the hub room. The hubs or switches in the hub room are connected to the main server room using a fibre-optic cable. Using a fibre-optic cable means that the network can be more than 100 metres from the server room.

Activity

The diagram shows an area in a school that is going to be connected to the school LAN. The area contains four classrooms, an IT classroom containing 20 computers, an office and a storeroom.



Each of the classrooms (including the IT classroom) must have:

- two network connections on the teacher's desk
- a network connection for a WAP installed on the ceiling.

The IT classroom should also have:

- a single network connection for each of the 20 computers
- a network connection for a printer.

The office will be used by four teachers. Each will have their own desk.

- Each desk will have two network connections.
- There will be an extra network connection for a printer in the office.

Draw a copy of the plan. In each room, write the number of network connections needed.

How many network connections are needed in total?

Extra challenge

Hubs or switches for the network will be kept in a cabinet on the wall of one of the rooms. Suggest two rooms that the cabinet could go in.

The school has two spare switches it can use. Each switch has 24 input connections. Does the school need to buy any more switches?

Test

1 List four network components that are contained in a home router.
2 What are the two types of cable used in LANs?
3 Give an example of where a fibre-optic cable is used in a LAN.
4 Explain why hub rooms are used in LANs.

1.3 Sending messages over the internet

In this lesson

You will learn:

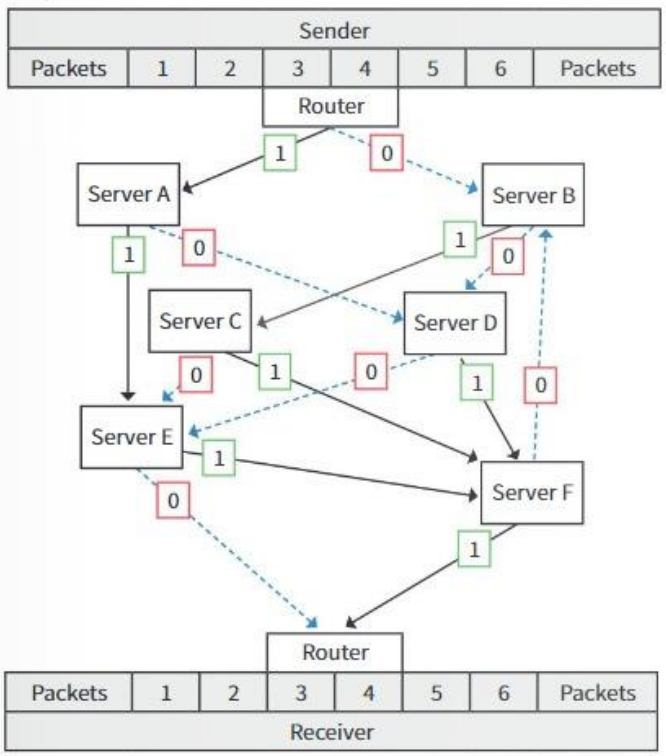
- the rules the internet uses to send messages.

Being connected

Being connected to the internet means you can send and receive messages. There are many types of messages you can send. These include simple text messages, audio, video and photographs.

To a computer, a text message and a video are treated the same – they are both **digital data**. A computer stores all files as digital data made up of zeros and ones. It sends every file across the internet in the same way.

In this lesson you are going to build a model. You will use it to learn how messages are sent over the internet. When you have completed the model, it will look like the one in the diagram. It looks complicated but you will build it step by step before using it in the next lesson.



Activity

Get ready to build your model. You will need a large sheet of paper, two or three coloured pencils or pens, and a ruler. If you don't have a large sheet of paper, stick two sheets of A4 printer paper together along the long edge.

First, add a sender section to the top of your sheet. There are three parts to it:

- the word 'Sender'
- a row labelled 'Packets'. This row should contain six boxes, numbered 1 to 6.
- a box labelled 'Router'.

Add a similar section at the bottom of your sheet. This section is called 'Receiver' and is a mirror image of the 'Sender' section you have drawn at the top of the page.

Do not draw the sections too large. Leave plenty of room in the centre of the sheet.

What are packets?

A computer does not send messages as a single section of data over the internet. Any file that you send over the internet is split into smaller sections before it is sent. The small sections are called **packets**. Whether the message is an email, a photograph or a word-processed file, it is split into small packets before being sent. This makes sending a message easier for the equipment in the network to handle. A packet is about the size of this paragraph (about 550 characters). If you could look inside a packet you would see digital data – 0s and 1s.

When the packets reach their destination, they must be put back together into the original message. You don't have to worry about creating packets or putting them back together. Your computer does it automatically whenever you send a message over the internet.

Many thousands of packets are sent if you are sending a photograph or another large message. Each packet is addressed to the person receiving the message. Each packet is numbered so that the message can be put back together in the right order.

What does a router do?

In Lesson 1.1, you learned that a router is used to connect your school or home network to the internet. Every packet in your message will pass through the router one after the other. When you use the model, you will move packets onto the router box before they are sent.

Activity

Go back to the model you started to draw. In the space in the centre of your page, do the following:

- Draw six boxes. Don't make the boxes too large. Make sure there is plenty of space between them.
- Label the boxes 'Server A' to 'Server F'. Look back at the diagram on page 14 if you're unsure how the boxes should look.

What does a server do?

As discussed in Lesson 1.1, the job of a server is to send files over a network. When you send a message over the internet, it is split into packets. Each packet travels in steps across the internet, jumping from one server to another until it reaches its destination.

Activity

In this activity you will add links, joining servers in your model together. In the real world, each server will be linked to many other servers. You will add just two links from each server in your model. This keep things simple.

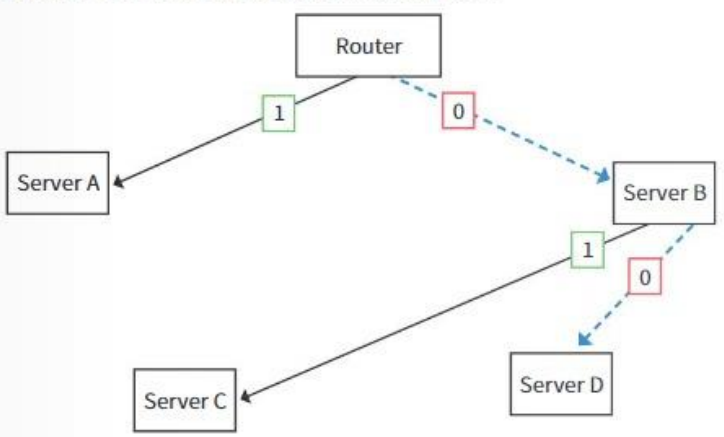
To make your model easy to follow, choose two different line styles.

- Use two different colours.
- Label one style of line '0' and the other '1'.

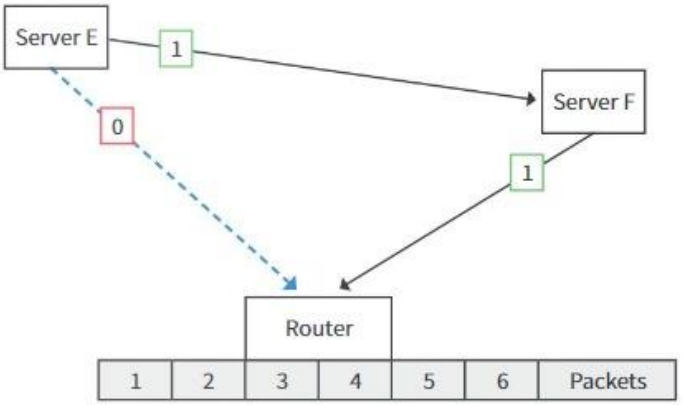
Add an arrow to indicate the direction of the line.

Start from the router at the top of the page. Draw connections to two servers. Use different line styles.

Work through each of the servers, adding two links to the other servers in your model. Make sure each server is linked to at least once.



Make sure two lines (one of each style) link to the router in the receiver, as shown in the image.



Extra challenge

Look back at the completed model from the start of this lesson. Check the model you have drawn to make sure it looks the same. If it doesn't, make sure you understand where you have made mistakes. Update your model.

How a packet travels over the internet

You have learned that a message is passed from server to server as it finds its destination. There is no set route for the message. When a server receives a packet, it must decide where to send it to next. The server will always send a message along the clearest route, avoiding slow or blocked routes.

Test

1 What is a router used for?
2 How does a packet travel across the internet?
3 Why are internet messages sent as packets?
4 Explain why packets sent over the internet have to be numbered.

1.4 Sending messages: a simulation

In this lesson

You will learn:

- how messages are sent over the internet using the packet switching method.

A simulation

In this lesson you will use the model you drew in Lesson 1.3 to show how messages are sent from one computer to another over the internet. The model shows how a message is sent step by step. You are going to use it in a simulation. A simulation copies something that happens in real life.

What you know so far

- All data sent over the internet is digital data.
- A message reaches its destination by being passed from server to server over the internet.
- A message is broken up into small packets before being sent over the internet.
- A router is used to connect a local computer to the internet. All messages sent over the internet will pass through the router.

Preparing for the simulation

You will need:

- your completed model
- a tracking table, which you will create in this lesson
- a coin
- six squares of paper, the same size as the numbered squares in the packets row in your model. Number the squares 1 to 6. These will be your packets of data.

Activity

Create a table like this.

Packet tracker								
	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Packet 1								
Packet 2								
Packet 3								
Packet 4								
Packet 5								
Packet 6								

Playing the simulation

Set up your simulation board. Start by putting the six numbered squares in the packets row of the 'Sender' section. The objective of the simulation is to move the message through the network of servers to the receiver section.

You are going to flip the coin during the simulation to decide which path each packet will take between servers. Decide which side of the coin represents '0' and which represents '1'. Make a note so that you remember.

Move Packet 1 to the router. Now you are ready to start.

Rules

- Flip the coin at the start of each round. The result will determine which path all packets will take in that round.
- Move each packet that is in play to the next server using the path (0 or 1) decided by the coin flip. A packet is 'in play' if it is on a server or waiting in the sender's router box. It is easier if you move the packets in number sequence from the lowest to the highest.
- In the table, record the server letter each packet has moved to. When a packet reaches the receiver router, write an 'R' in the table.

Packet tracker								
	Step 1	Step 2	Step 3	Step 4	Step 5	Step 6	Step 7	Step 8
Packet 1	A	D	E	F	R			
Packet 2	B	D	F	R				
Packet 3	B	C	F					
Packet 4	A	E						
Packet 5	A							
Packet 6								

- Move any pieces that have arrived at the receiver router onto the next available slot in the packets row. Do not put the packets on their matching slot in the receiver's packet row. Put them on the next available slot in the order they arrive. So if Packet 6 arrives fourth in sequence, it goes on slot 4, not slot 6.
- Move the next sender packet onto the router.

Repeat the instructions until all packets have moved from the sender to the receiver. It will take around eight or nine rounds to complete the simulation. There will be up to three or four packets 'in play' at any time.

Activity

Play the simulation, then answer the questions below. You might want to play the simulation in a small group.

Look at the tracking table.

- Did all the packets take the same number of steps to reach the receiver?
- Did all the packets take the same route through the network?
- Did *any* of the packets take the same route through the network?
- Did the packets arrive with the receiver in the same order that they set off from the sender?

It is possible, in fact likely, that the packets you sent arrived out of sequence. There is a final step that needs to be added to the model to make sure that the receiver can understand the message. Do you know what that step is?

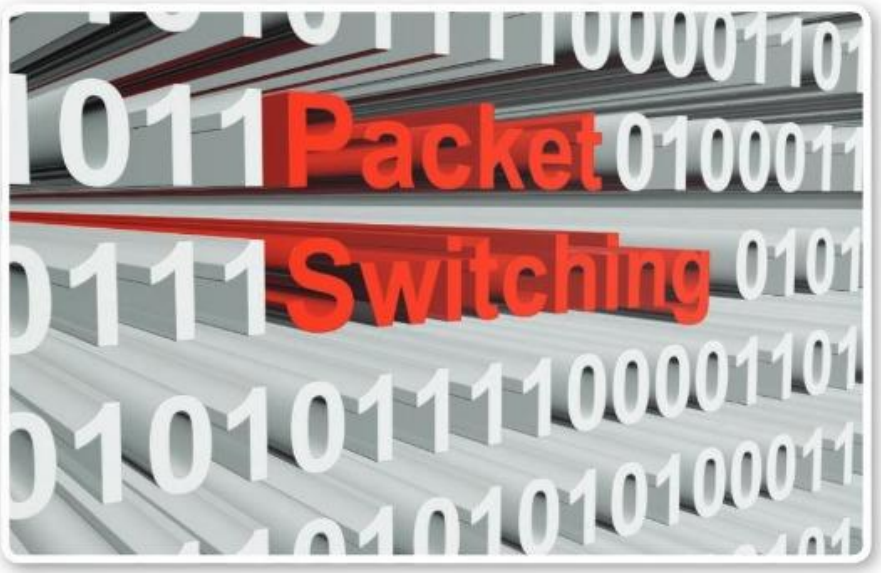
Packet switching

The simulation you have just played is a simple way of showing **packet switching**.

- A message is split into small sections called packets.
- As a packet travels across the internet, it is switched onto the clearest route.

Packet switching is a reliable way of sending messages over the internet. Almost all messages reach their destination correctly. Packet switching is used to send data across local area networks as well as the internet.

Packet switching is carried out automatically by your computer and other hardware in a network. You use packet switching every time you log on to a network.

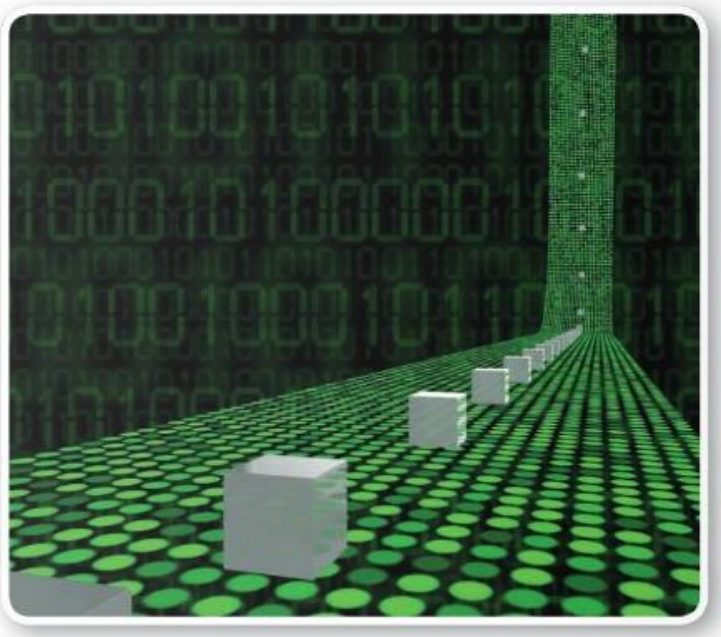


What a packet contains

You already know that a packet contains a small part of a complete message. The data in the packet is stored as digital data.

Each packet contains some extra data.

- The **address** of the person the message is being sent to. When you send an email, you must always add the address of the person you are sending it to (for example, a.friend@gmail.com). When your message is split into packets, each packet must contain the address.
- The **sequence number** of the packet. You saw in the simulation that packets can arrive at their destination in the wrong order. A sequence number allows the receiving computer to sort the packets into the right order.
- The **total number of packets** in the message. The receiving computer uses this information to check if it has received all the packets. If any packets are missing, the computer sends a message back to the sender and asks for them to be re-sent. This is done automatically so you don't have to re-send the whole message.



Extra challenge

Sometimes packets get lost as they are sent across the internet. What problem does that create for the computer receiving a message?

Research the web to find out how packet switching deals with lost packets.

Test

1. What happens if a packet gets lost when being sent across a network?
2. Name three pieces of information that are included in a packet when it is sent over a network.
3. Why can packets arrive at their destination in the wrong order?
4. Describe the term 'packet switching' in your own words.

1.5 Connecting to a network

In this lesson

You will learn:

- how to connect to a network
- basic troubleshooting methods for network connections.

Types of network connection

There are three main ways to connect to a network.

1 **Wired** Your computer is connected to a data point on a wall with a cable. You can only connect to a network if you have a network point nearby. You cannot move your computer. Wired connections are used for desktop computers and sometimes for laptops.

2 **Wireless (Wi-Fi)** A cable isn't needed for a wireless connection. Signals are sent between your computer and a wireless access point (WAP). Lesson 1.1 has more information about WAPs.

Laptop computers, tablets and smartphones connect to networks using Wi-Fi. Using Wi-Fi means you can move around without easily losing your connection.

Wireless networks are sometimes available in public places like restaurants or shops. A WAP in a public place is called a **hotspot**.

3 **Mobile phone** Smartphones can connect to the internet using a mobile phone connection. A connection can be made anywhere there is a phone signal.

How to connect to a network

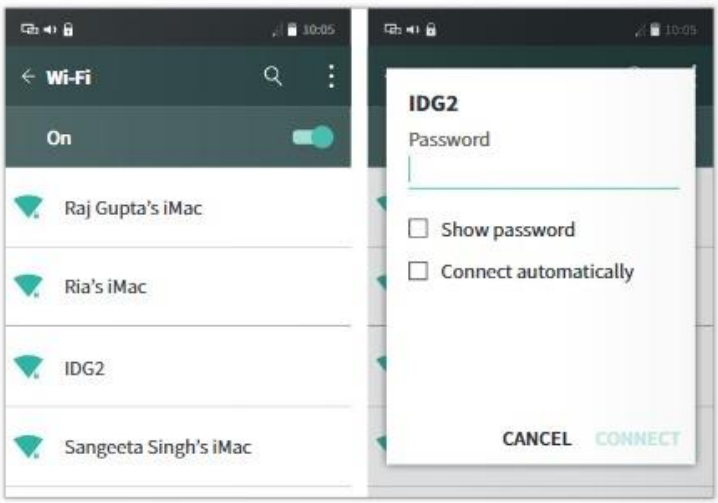
School network

Everyone who is allowed to use the school network is given a username and password. You use the username and password to connect to the network. The login process is the same whether you connect by a wired or wireless connection.

Home network

You do not need a username and password for your home network. Everyone uses the same login details. To log in to the network, you need to know the network name and **network security key** for the router. This information is on a sticker on your router. The network security key works like a password.

When you switch on your computer, you will be shown a list of available networks. Your computer will detect your home network and any other networks close to you. It will detect your neighbour's network, if they are close enough, and any public hotspots.



When you click on the name of your network, you are asked to enter the network security key. Entering it correctly will connect you to the network. When you have entered the key once, your computer will remember it. If you tick the 'Connect automatically' box, you won't need to log on in future. Your computer will automatically connect to the network.

Solving network problems

When you are solving computer problems, you will be working with electrical equipment. Your priority is to stay safe. Always follow your school computer use safety rules. Always check with your parents before trying to deal with problems with your home network.

Stay safe

Always follow these safety rules.

- Switch off power points at the wall before connecting or disconnecting a mains electricity cable.
- Do not open the case of your computer or any other device.
- Never use force when connecting or disconnecting a cable or component.
- Make sure all cables are disconnected before attempting to move a device (for example, a printer).
- Do not lift heavy equipment on your own.
- If in doubt, STOP and ask for help.

Be creative

Create a poster to encourage people to act responsibly and safely when solving computer problems.

Make notes

When you are trying to solve network problems, record:

- the location of the computer
- the steps you have followed
- any error messages you see on-screen.

Username and password

If you are having problems logging on to a network, check you are using the correct username and password.

A password is case-sensitive. That means it matters if you use upper case letters instead of lower case ones. Check you haven't left 'Caps Lock' on.

If you have forgotten your password, use the 'Forgot password?' link to change your password. You will need an email address that a new password can be sent to. At school, you will need to ask your IT technician to change your password.

Wired networks

If you are having network problems with a wired connection, check that the cable connecting your computer to a network point is connected securely at both ends.

The connection socket on your computer will usually have a small green light. If the light is flashing, your computer is connected.

There are two things you can check if the light isn't flashing.

- The cable connecting your computer to the network may be faulty. Changing it for one you know is not faulty will solve the problem.
- The network point you are connected to may be faulty. Try another network point.

Only try to solve problems with your school network if your school IT policy allows it. Report the problem to the IT technician or your teacher and use another computer until the problem is fixed.

Wireless networks

If you cannot make a wireless connection to a network, check if you have wireless enabled on your computer. Find the Wi-Fi icon on the tool bar at the bottom of your screen. If it has a red cross on it, wireless on your computer may be turned off.

The picture on the right shows the wireless network symbol.

Click the Wi-Fi icon to open the wireless control box. You can turn wireless on by clicking the button marked 'Wi-Fi'.

If you are using a tablet or smartphone. you will find the same Wi-Fi icon on your device. If the icon is grey, wireless is turned off. Tap the icon to turn wireless on.

Home network

If you are having problems connecting to your home network, check:

- you are entering your password correctly
- that wireless is turned on
- whether other people in the house have a connection. If they do, the problem is likely to be with your device, not the router.

There are some basic checks you can carry out on your home network router.

- Is the router plugged into an electricity socket? Is the socket switched on?
- Check that any cables connected to the back of your router are secure.

If everything else fails, reset your router. Turn off the power switch or turn off the electricity socket. Wait 30 seconds then turn the router back on again. Do not press any button marked 'reset' on the router.

Activity

Write a list of instructions called 'How to connect to a network'.

Design the instructions for students who are about to start using computers, to help them study at school and at home.

Extra challenge

Work in a small group. Use your multimedia skills to create an audio or video guide to network connectivity. You can use a phone or recorder. Record yourself reading the instructions you wrote, or make a short video of someone connecting to the school network and explaining how to do it.

Test

1. List three ways to connect to a network.
2. What is a hotspot?
3. You are unable to connect your computer to your home network. List four problems that could be preventing you from connecting.
4. You want to connect a laptop to your school network. Why is a wireless connection better than a wired connection?

1.6 In the cloud

In this lesson

You will learn:

- about cloud storage
- about other services that are available in the cloud.

Spiral back

In Student Book 4 you learned about computer storage. You can save files using a storage device on your computer. Files can also be saved on a network storage device. You probably use network storage at school.

What is the cloud?

The **cloud** is a term used to describe the internet. The cloud is a metaphor for the internet. When you use a metaphor, you are saying that the internet is like a cloud, without using the word 'like'.

Saving a file to the cloud means saving the file to the internet.

Cloud storage

Everything you do 'in the cloud' involves data storage. When you save file in the cloud, it is stored on large storage devices. The storage devices are held in a **data centre**.

A data centre is a very large computer system. It contains many servers and storage devices. A data centre often has its own building.

Who provides cloud storage?

There are many providers of cloud storage. Your internet provider may offer cloud storage. Software providers like Microsoft and Google provide cloud storage. Microsoft's cloud storage is called OneDrive.

You often get free cloud storage when you buy a computer, or other devices like cameras and smartphones. Some companies specialise in providing cloud storage. Dropbox is an example.

Usually, a small amount of cloud storage space is provided free of charge. Businesses and people who have a lot of data can pay for more space if they need it.

Advantages of cloud storage

Here are the main advantages of cloud storage.

- You can access your files from anywhere you have an internet connection. If you save a file on your own computer, you can only access the file if you have the computer with you. If your file is in the cloud you can access it anywhere, using any computer.
- You can share your files with other people. For example, when you are working on team projects.
- Your work is safe. Cloud storage providers back up your files so that you don't have to do it. It is unlikely your data will be lost or stolen.

Other cloud services

Storage isn't the only reason for using the cloud. There are many other services.

Collaboration

As you saw in Student Book 7, collaboration means working with others.

Collaboration software makes working together easier. If files are stored in the cloud, any team member can work on them.

Collaboration software also lets people in different locations meet using web conferencing software. **Web conferencing** shows a video of the people taking part in the meeting. It also lets people show files on-screen and make notes. A web conference saves travel time for the people taking part.

Software applications

Many software applications (apps) work in the cloud, for example, Google Apps and Microsoft 365. The apps make it easy to work with other people. They make sharing documents and files easy.

Web hosting

If you want to build a website, you need a web server to store your web pages. A web server can be expensive, and most people do not have the skills to set up and run one.

Web hosting companies help people who want to set up a website. A web hosting service provides technical knowledge and web editing software. That makes creating a website easy.

Music, videos and games

Music and videos have been stored online for many years. When **digital** music was first stored online, people bought and downloaded the music. When downloading, you save a copy of the digital music file on your computer. When you have saved the copy, you can play the music on your computer, or transfer it onto a special MP3 player, such as an iPod.

Today, music and videos are often streamed to a computer or device like a smartphone. **Streaming** means that a music track is played as it is being downloaded to your computer. A copy of the file does not have to be saved before the music is played. Streaming is a cloud service.

Playing games in the cloud means that you can compete or collaborate with other players. This type of game is called a **multi-player game**. Multi-player games can be more interesting than single-player games. They often have social media tools included to allow players to chat about the game.

Advantages and disadvantages of cloud services

Here are the advantages.

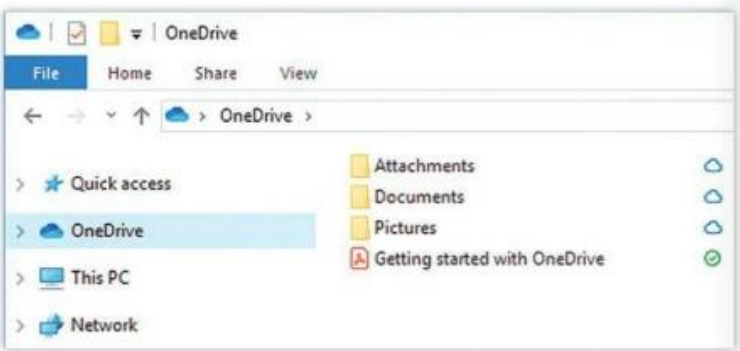
- Files are available anywhere and on any computer.
- Collaborating and sharing data is easy.
- Data is safe.
- Low cost to start with.
- Technical help is available from the company providing the service.

Here are the disadvantages.

- The service might not be exactly what you need.
- The owner of the service can change how it works.
- The owner of the service can increase costs.
- Can be expensive over time.

Saving to the cloud

When you save a file, you must choose where you save it. You can save it on a storage device on your computer. You can save it on a network storage device. You can save it to the cloud. The places will be listed for you when you save a file.



When you save a file to the cloud, a copy of the file is also saved to your own computer. This means you can open the file and work on it even if you have no internet connection.

You can choose not to make a copy of a file on your computer. The file will only be saved to the cloud. You will not be able to use the file if there is no internet connection. You can save large files like videos and photos to the cloud to save space on your computer.

Cloud storage uses icons to tell you how a file has been saved.

- A cloud status icon shows that the file is only saved in the cloud.
- A clock status icon tells you the file is saved to the cloud and your computer.
- An icon that looks like a person tells you the file has been shared with someone.

Activity

Your company is about to launch a new cloud data storage service.

Write an advert to attract as many new customers as possible. Make sure your advert highlights the advantages of cloud storage.

Extra challenge

Do you use any cloud services? For example, multi-player games, or streaming music and videos.

Make a list of the services you use. Which are your favourite services and why?

Is there anything you don't like about the cloud services you use?

Test

1. Describe three services that can be accessed in the cloud.
2. Where is data saved when it is saved to the cloud?
3. Explain why it is a good idea to store your data files in the cloud.
4. List three disadvantages of using cloud services.

Check what you know

You have learned

- about network hardware and how the components work together
- how messages are sent over the internet
- about packet switching
- how to connect to a network and how to solve problems with network connections
- about cloud storage and other services in the cloud.

Try the test and activities. They will help you to see how much you understand.

Test

1 List two items of information you need to log on to a network.

2 Where is a data file stored if it is 'in the cloud'?

3 What are the two types of cable used to connect network equipment? Describe the advantages of each type.

4 Describe three computer services that can be used in the cloud.

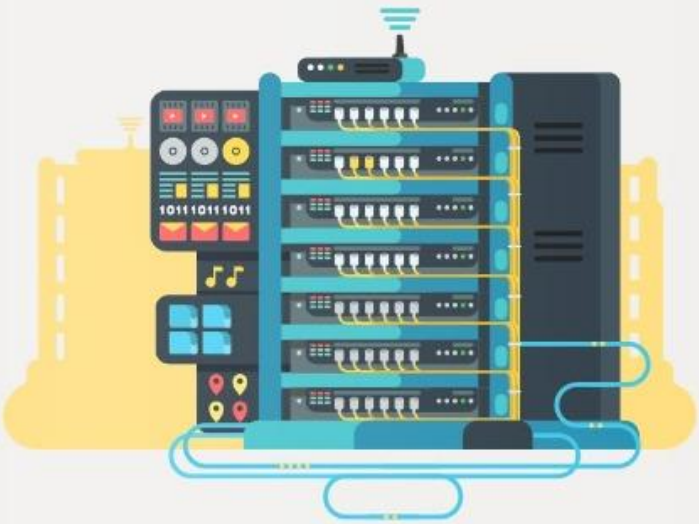
5 You are having difficulties logging on to your home network. Describe three checks you can carry out to try to solve the problem.

6 Why can it be better to store data files in the cloud rather than on your own computer? Give three examples.

Activities

You have been asked to create a presentation. It will be used to explain to other students how the components of a network are connected to allow computers to communicate.

1. Create one or two slides that tell students how to connect to your school network and use the internet. You can include screenshots or pictures to help explain.
2. Create one or two slides about the items of network hardware you might find at home or in schools. You can include pictures or photos that you have taken yourself or found in internet research.
3. Add one or two slides that explain what to do if you have problems connecting to the school network. Give some hints about how to fix the problem.



Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1.
- I answered test questions 1–4.
- I completed activities 1 and 2.
- I answered all the test questions.
- I completed all the activities.

Re-read any parts of the unit that you feel unsure about. Try the test and activities again – can you do more this time?

2 Digital literacy: Computers and learning

You will learn

- how computers are used to help people learn
- how computers are used to make discoveries
- how to use online research to help complete a project.

You can use computers to help you in any subject you study. The skills you learn in your computing classes do not just help you to learn while you are at school. They will be important throughout your life. When you leave school, computers will be important in your career.

Computers have been used to make important discoveries. You will learn about some of these discoveries in space exploration, medicine, and in understanding natural disasters such as earthquakes. Perhaps in the future you will contribute to a major discovery.

Later in the unit, you will work in small groups to create a short e-learning course. You will use online research to help you complete the project.

Unplugged

How do you use computers to help you do your schoolwork? Make a list. What software and hardware do you use?

Give examples of times you have used a computer to make work you are proud of.

Learning outcomes: Carry out an online research project; Explain how computers can help with learning and discovery

Did you know?

Scientists in a group of British universities created a 'robot scientist'. They claim it is the first computer to have discovered new scientific knowledge. The robot is called Adam. Adam uses **artificial intelligence** to carry out research. Artificial intelligence is a technique that programs computers to think and work like humans. Adam made a discovery about yeast. The discovery was simple. But it was important because it was made without much human help.

Talk about...

Computers can support your work in many subjects, not just computing. In which subjects do you use computers most?

Are there any subjects where you don't use computers at all? Why do you think that is?

artificial intelligence
assistive technology
computer models
e-learning
guidance computer
interactive video
multimedia
simulation
supercomputer
virtual reality
speech generator

2.1 Using computers for learning

In this lesson

You will learn:

- how computers can help you learn and discover information in all the subjects you study.

Spiral back

Computers are not just for computer studies. You can use computers to help you learn in any subject.

You have already learned some skills that will help you in your studies. You have learned how to:

- find information, to help you make the most of the web
- present information using word-processing, spreadsheet and presentation software.

In Student Book 5 you learned how to use a web browser to find information on the web. The skills you have gained in writing search strings should help you with other subjects like geography and history.

What do you already know?

Here is a reminder of some of the skills you have already learned.

You already know how to:

- **identify key words to use in a search**

  Think about the question you are trying to find answers to. Write your question down and underline the key words. This will help you identify what must be included in your web search.

- **use an age-appropriate web browser**

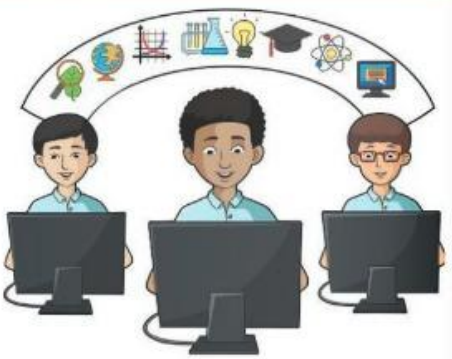
  Using a browser like Kiddle will show you information that is suitable for your age. It will filter out adverts.

- **use special search words and characters**

  You can use special characters in a web browser to make searching easier. For example, you can use the – (minus) sign to exclude terms from the search

- **check the reliability of information you find**

  In Student Book 5, you learned how important it is to check the sources of your information. Can you identify who wrote the information? Does the site publishing the information have a good reputation? Can you check key facts in the information on another site?



- **use bookmarks**

  Bookmarks make sure you can find useful sites again when you need them.

- **give credit for another person's work.**

  If you use content that belongs to another person, you should always indicate where the content comes from.

Now you have started to develop these skills, you can use the web to improve your work in all the subjects you study.

Activity

Choose two subjects that you study other than computing. For each subject, pick a topic that you are studying or have recently studied. Find one or two websites that provide useful information on these topics.

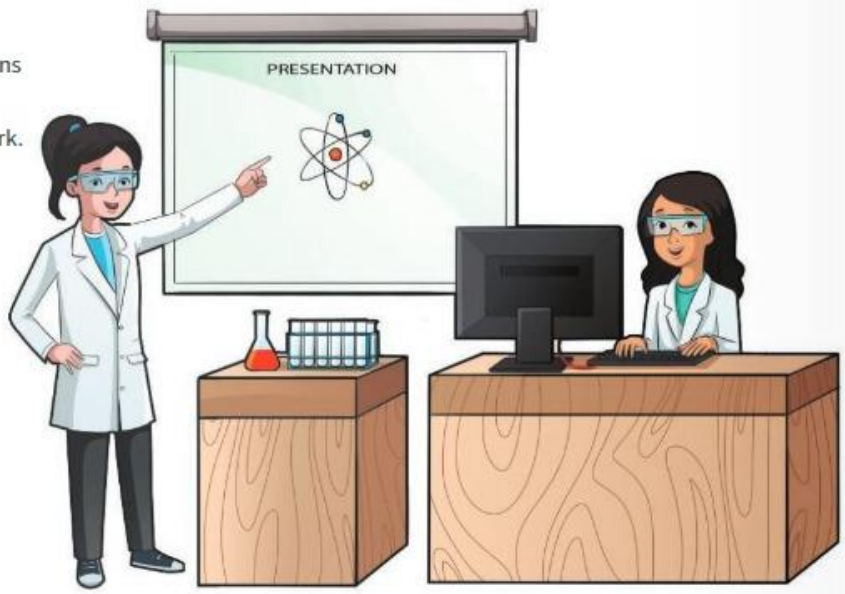
Share the sites with a small group of classmates. Did any of your group suggest other sites you will find useful?

Presenting your work

Think about the skills you have developed using application software. You have used a word processor, a spreadsheet and presentation software. You have also used graphics software to make images. All these skills are useful in any subject you are studying.

You can use application software to:

- write important course notes
- write assignments and homework
- create posters
- make presentations
- add images to illustrate your work.



Explore more

What are the advantages of using a word processor instead of writing your work by hand?

How much do your parents use word-processing apps in their work/at home rather than writing things out by hand? Discuss with them the advantages and disadvantages of using a word processor.

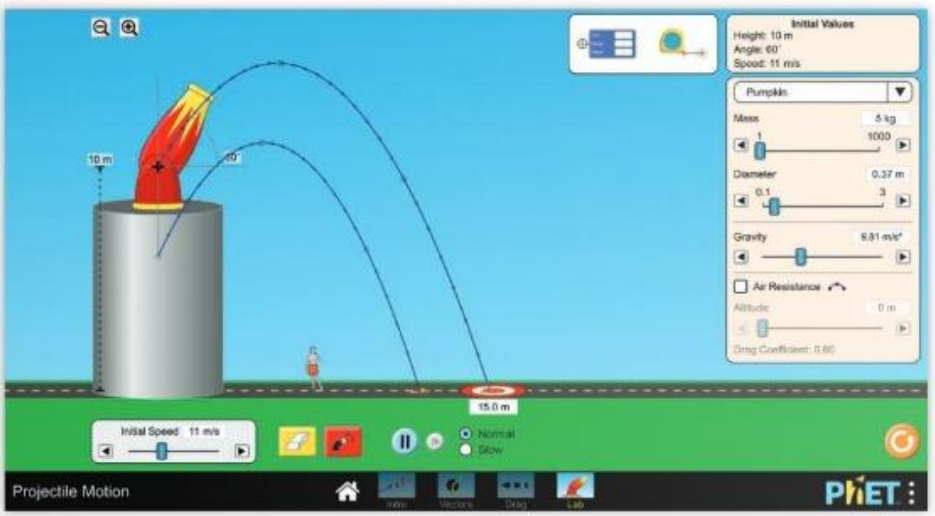
Other uses of technology for learning

Simulations

A computer **simulation** is a program that imitates something you can see or experience in real life. A simulation is like a game you play on your computer. You are in control of objects in a game. You can change how fast they move and the direction they move in.

You can change the way a simulation works. Simulations and games are interactive. This makes simulations a very useful tool for learning.

In science, simulations can be used for experiments. Chemistry experiments with dangerous chemicals can be simulated without students being at risk of injury. In geography, students use simulations to explore how glaciers are formed, or how greenhouse gas emissions affect global warming.



Here are some of the advantages of using simulations.

- They can be done anywhere.
- Experiments can be done many times.
- They speed up processes that take a long time.
- You don't have to set up complicated and expensive equipment.
- They can be used in situations where a real experiment is difficult to do.

Activity

Search the web to find a computer simulation. Use a search string like: 'free computer simulation science elementary school'.

Did you like using the simulation? Why or why not? Was it easy to use?

Virtual reality

Some schools have started to use **virtual reality** (VR) headsets for learning. You may be familiar with virtual reality from computer games. Wearing a VR headset gives you the feeling that you are in a different place. A virtual world is created around you. The world is interactive, like a simulation.

If you are learning about a place in geography or history, you read a description of the place. You can look at pictures or watch videos. With a VR headset, you can step inside the place. You can walk around and explore it.

Assistive technology

Assistive technology helps people with disabilities to learn.

Software can make on-screen text larger and clearer to help students with poor vision. Screen readers read out on-screen text for those with little or no vision. Braille converters translate on-screen text into a pattern of dots called braille. Braille is used as a way of reading by people who have no sight.

Some students have disabilities that make it difficult for them to speak. A **speech generator** turns text that you type into speech, using a simulated voice. The device saves phrases and words that you use often. This saves you from having to re-type everything.

Activity

A mouse can be difficult to use for some people with physical disabilities.

Search the web and find three devices that you can use instead of a mouse.

Extra challenge

Choose one of the devices from the activity.

Make an information sheet for the device you have chosen. Include a picture and write one or two paragraphs to describe how the device works.

Test

1. What is a computer simulation?
2. Give two examples of how assistive technology helps people with disabilities.
3. How can virtual reality be used to help people learn about geography?
4. How do simulations help people learn?

2.2 Using computers for discovery

In this lesson

You will learn:

- how computers can help make discoveries.

Space discovery

The Eagle has landed

On July 20, 1969 the Apollo Lunar Module called Eagle made the first landing on the moon. Neil Armstrong and Buzz Aldrin walked on the moon before returning safely to Earth.

Six Apollo missions to the moon returned safely to Earth. One of the reasons the Apollo missions were safe and successful was the Apollo **Guidance Computer** (AGC).

The AGC was built specially to guide spacecraft to the moon and back. It was the first computer used to guide flights. Today, all aircraft use guidance computers.

Compared with modern computers, the AGC was heavy, slow and difficult to use. It weighed as much as ten modern laptops. The computer chip in a car satellite navigation system is many times more powerful than the AGC.

Activity

There are programs that simulate the Apollo Guidance Computer. They show you how the computer looked and how to enter instructions.

Search the web for 'Apollo Guidance Computer Simulator'. What are the input and output devices it uses? How is it different from the computers you use?

Life on Mars?

Robots are used to explore planets. In January 2004, two identical robot explorers called Spirit and Opportunity landed on Mars. The explorers were supposed to work for 90 days. Opportunity worked for 15 years.

The robot explorers were designed to study the rocks on Mars in the same way that a geologist studies rocks on Earth.

The explorers had:

- two video cameras acting as eyes
- a flexible robot arm to examine rocks
- a hammer to break rocks open
- instruments to analyse rocks
- a microscope to look closely at rocks.

Spirit and Opportunity did not discover life on Mars. They did discover that ancient Mars had water and hot springs that could have supported simple microscopic life.

Activity

Look at the picture of the robot explorer. It seems to have a wing on each side. It doesn't fly.

What are the wings for?

Predicting natural disasters

Weather forecasting

Weather events like hurricanes can be devastating and life-threatening. Understanding how weather events form and move lets scientists called meteorologists predict where bad weather will occur. Early warnings about bad weather gives people time to prepare, or to evacuate an area.

There are two parts to predicting weather.

1. Satellites and radar record real weather conditions. Data about the weather is fed into a computer.
2. The weather data is processed using algorithms that predict how weather conditions will change over the coming hours and days.

This allows meteorologists to predict the course of hurricanes. They can warn people days before the hurricane arrives. The predictions are compared with what really happens. This is so meteorologists can discover more about the way storms work. They change their algorithms to help improve predictions in the future.

Very fast and powerful computers called **supercomputers** are used for this work. A supercomputer can process huge amounts of data very quickly.

Earthquakes

Another type of natural disaster is an earthquake. Parts of the earth move until rocks under the surface break. Huge forces cause the earth to shake violently. Buildings are damaged. Sometimes this causes them to collapse.

Scientists who study earthquakes are called seismologists. Seismologists use sensors to detect movement in the earth. They use satellite images of the earth. They feed this data into a computer model. Algorithms in the model help seismologists discover how earthquakes work.

Scientists now know that an earthquake is going to happen, but they don't know when. They hope that supercomputers will be able to predict earthquakes in the future. Predicting earthquakes will allow people to prepare for them. It will save many lives.

Activity

What preparations could you make if you knew in advance that a hurricane or earthquake was going to happen in your town? How could people be warned?

Medical discovery

DNA

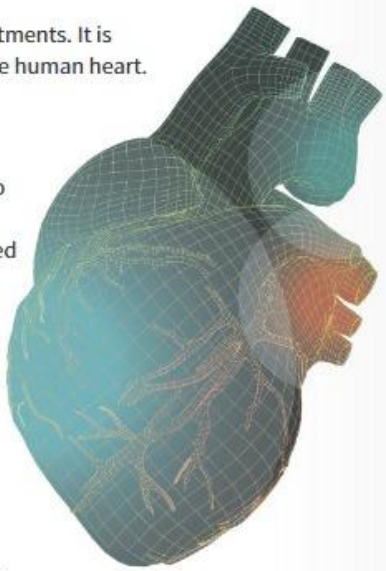
Understanding DNA has led to many important medical discoveries in the 21st century. DNA research involves millions of calculations on large amounts of data. This has only been possible using computers.

Scientists believe that DNA computers will be developed in the future. A microscopic computer made of DNA will be able to travel through the body and repair cells that are damaged by cancer or other diseases.

Computer models and medicine

Meteorologists and seismologists use **computer models** to make discoveries and to predict events. Other scientists use models too. Doctors and medical scientists use computer models. One use for models is to predict the effect that new treatments will have on patients.

Computer models are used to develop new drugs and other treatments. It is important to know if a new medicine will have a bad effect on the human heart. A computer model can help doctors to find out.



Animal and human testing

Drugs are usually tested on animals before permission is given to use them on humans. Before a new drug can be used, it must be tested on a small group of humans to check it is safe. This is called a clinical trial.

Many people think testing new medicines on animals is cruel. Clinical trials are usually safe but can be dangerous for the people involved. Doctors would like to develop more computer models so that they can test drugs using computers. Some medicines are already tested without using animals.

Activity

Why do you think it is important to use computers to understand and predict natural disasters? Give three reasons.

Extra challenge

Do you think testing medicines on animals is the right thing to do? Would you trust a medicine that has only been tested using a computer?

Outline the advantages and disadvantages of testing medicines using just a computer.

Test

1. Why does a meteorologist use computers?
2. Name one major space discovery that used computers.
3. What are computer models used for in medicine?
4. Name three pieces of equipment used by Mars explorer robots. Can you think of another piece of equipment that would be useful for researching the surface of a planet?

2.3 What is e-learning?

In this lesson

You will learn:

- how computers are used for e-learning.

What is e-learning?

An important use of computers in education is for **e-learning**. The 'e-' in e-learning stands for electronic. E-learning is like classroom learning in many ways, but it takes place over a local area network or the internet.

With e-learning, you can decide when and where you want to study. E-learning doesn't have a timetable, although there are usually set dates to submit your work. In the classroom, materials may be printed on paper or projected on the whiteboard. In e-learning, all the learning is done on a computer screen.

Who uses e-learning?

E-learning is often used in the workplace for training people. It is used to help people get new skills, for example, IT skills. It is also used when people need to know about new rules or ways of doing things, for example, if a government passes a new law that people in a particular job must know about.

E-learning is never the main type of learning in schools. It may be used by teachers to give you extra help. E-learning can also be used to give you extra challenges. Students can use e-learning at home to do extra study.

Multimedia and e-learning

In your classroom you have books to learn from. Your teacher gives you information. Sometimes, this information is given to the whole class using the board. At other times, you might work in small groups around a table. Your teacher may show videos or give a presentation using a projector.

There are many ways you can learn in the classroom. E-learning is no different. Many different types of media are used for learning. When different types of media, such as video and text, are used together, it is called **multimedia**.

Here is some of the media used in e-learning.

- **Video and interactive video**

  Video can be used for demonstrations, for example, how to use a piece of software. Or it can be used to show what an animal is like, while a commentary or words on-screen give facts about the animal.

  Sometimes you can decide how a video is played. For example, a video clip might be followed by a question. The next clip you see depends on what your answer to the question is. What you see depends on the action you take. This is called **interactive video**.

- **Presentations and audio**

  You have created presentations in earlier books. Some e-learning lessons are shown as a presentation. Presentations show information on a series of slides. An audio file is sometimes used to explain the content on each slide.

- **Text and images**

  Some e-learning material is made up of simple text with images, like the books in your classroom. Text and image content can be cheap and easy to make. It can be created quite quickly by teachers who do not have the IT skills to make videos, games and simulations. Books that you read on-screen are called e-books.

- **Games and simulations**

  You must complete learning challenges to progress in an e-learning game. Games can make learning fun.

  As we saw in Lesson 2.1, a simulation is a model of a real-life situation, for example, a scientific experiment. In a simulation of a chemistry experiment, you can safely see what happens if you increase the temperature or change the number or amount of chemicals.

Some students learn best if they watch a video. Others prefer to read, or to use a simulation. E-learning uses a mix of media types. That allows students to learn using the materials that suit them best.

Activity

How do you prefer to learn things? Do you prefer to read or watch a video? Do you think an educational game would help you learn?

How important is it for your teacher to explain something to you?

Tests and assignments

When you study at school, you do tests and exams. You do assignments and complete homework.

There are also tests and assignments in e-learning.

Tests are made up of short questions. For each question, you choose an answer from a list of options, or provide a single-word answer. This type of question can be marked automatically by a computer. This saves time and provides you with instant results.

Assignments are longer pieces of work that a trainer or teacher must mark. You send the assignment using the computer and get your mark back in the same way.

How is e-learning taught?

Technology is changing the way that people work. Trainers and teachers are still important in e-learning but their job is not the same as in classroom teaching.

In e-learning, teachers and trainers spend less time:

- teaching classes of students
- marking work
- working face to face with students.

This gives them more time to:

- help students get better results
- talk to students online and answer messages
- help individual students
- create e-learning media.

More learning is delivered on-screen, so a trainer or teacher does not spend as much time teaching in front of a full class. This gives them more time to spend helping individual students.

Advantages and disadvantages of e-learning

Advantages

Here are some of the advantages of e-learning.

- You can learn where and when you want to.
- If you live in a remote area, you can learn without having to travel long distances.
- You can use the type of media you prefer.
- Teachers have more time to help individual students.

Disadvantages

And here are a couple of disadvantages.

- Some people prefer to learn in a class with other people.
- It can be difficult to work without a strict timetable.

Digital citizen of the future

Technology is changing quickly. This means you will need to learn new skills throughout your working life. E-learning is one way for you to keep your skills and knowledge up to date.

Creating e-learning materials is a job that combines creative work with computing skills. Is it a career you would consider in the future?

Activity

Plan a lesson for a subject you are interested in.

- State the topic of the lesson.
- Outline the information you will include in your lesson.
- Write down one or two questions you could ask to see if the learner has understood the multimedia content.

Extra challenge

Do online research to find an example of multimedia content you could use for the lesson you planned.

Test

1. Explain what multimedia means in the context of e-learning.
2. Give three examples of media types that can be used for e-learning.
3. List three ways that a teacher's job will change if e-learning is used in your school.
4. Describe three ways that e-learning could help you to learn.

2.4 Starting your e-learning project

In this lesson

You will learn:

- how to plan your e-learning project.

Planning your e-learning project

For the rest of this unit, you will work in small groups to create a short e-learning course. There are some suggested topics below, but you can also choose your own topic.

Each member in the group will produce:

- a slide that covers one lesson
- a test question.

You will make the e-learning content using presentation software. The examples in this unit use Microsoft PowerPoint. If you use different presentation software, your teacher will tell you the differences between your software and PowerPoint.

First, plan how your team will work together. Here are some things you need to think about.

Choose a topic

It's up to your team to choose a topic. You could choose something you have studied in your computing course. Or choose a topic from another school subject. Here are some computing ideas.

- Network hardware (Unit 1 of this book)
- Computers and space discovery (see Lesson 2.2)
- Computer hardware (input, output and storage devices)
- Cybercrime (Unit 2, Student Book 7)
- Staying safe on the internet (Unit 2, Student Book 7).

Agree who will do what

When you have chosen a topic, decide how you will split it up into lessons. You need one lesson for each member of the group. For example, if you choose computer output devices, you could decide to divide your course like this:

- What is an output device?
- Computer screens
- Printers
- Speakers.

You need one heading for each member of your team.

Design your presentation

When you have completed your e-learning lesson, you will put it together with the lessons made by other members of your team. The completed presentation needs to look good. Each lesson needs to have the same style so that it looks like one person has written the whole course.

Here are some ways you can help ensure you have a consistent style.

- Agree a **theme** to use for your completed course. Your presentation software will have backgrounds you can choose from.
- Agree a **template** to use for your content. Your presentation software will have templates you can use. A layout like the one below will be suitable. There is room for a heading and an image with text next to it.

The themes that you can apply to your presentation are in the 'Design' tab. Choose one with a white background. It will make your text easier to read.

You can see what page templates are available if you click the 'Layout' button in the 'Slides' section in the 'Home' tab.

Your teacher will show you an example of a completed lesson.

Starting your project

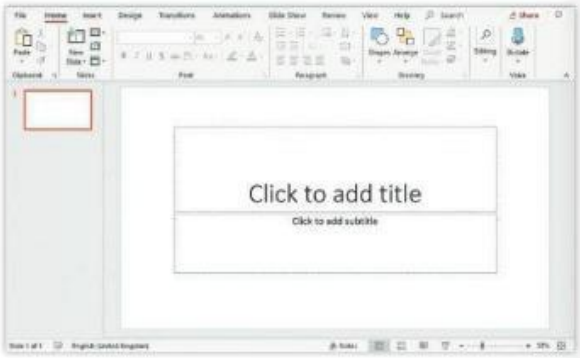
In Lesson 2.5, you will create your individual e-learning slides.

First, you will complete the title slide and contents page for your e-learning course.

Add a title slide

To add a title slide, open a new, blank presentation in your presentation software.

A blank title slide should be automatically inserted. If not, add a title slide.

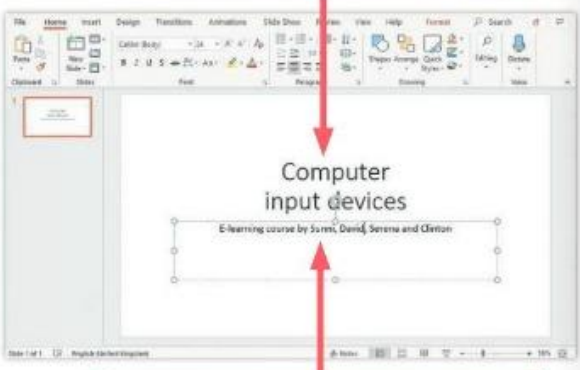


Add a title and credits

Now you are going to add a title and credits to your e-learning course.

Agree a suitable title for your presentation. It needs to describe what your e-learning course is about. It must not be too long.

1 Add a title to your presentation.



2 Add the names of your team members.

Choose a theme

Adding a theme to your presentation will make it look more professional. The theme you choose will appear on every slide, so try a few background designs before you make your final choice. Pick a theme that leaves plenty of space in the centre of your slides. A white background will make your text easier to read.

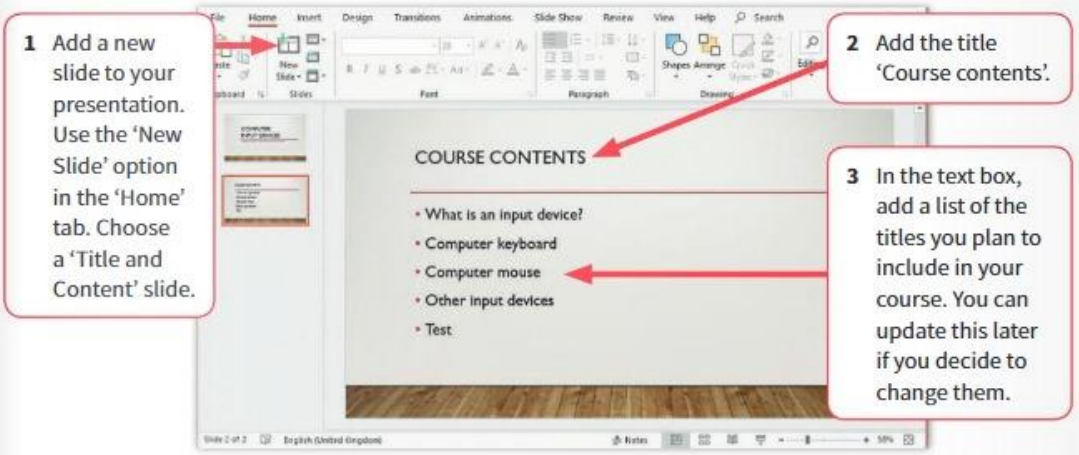
1 Select the 'Design' tab in your presentation software.

2 Browse the 'Themes' menu and select a theme.

Add a contents slide

Create a contents slide for your e-learning course. A contents page tells students what they will learn about when they study a course.

The contents slide should list the titles of the lessons your team members are going to create.



Save your work. You will use the presentation again in the next lesson.

When each member of your team has finished their slides, you will need to bring them all into a single presentation. Discuss how your team will do that.

Activity

Create the title and content slides for the e-learning course you have planned.

Extra challenge

Add an image to the title page. Choose an image that is relevant to your presentation. You can select an image using the 'Online Pictures' option in the 'Insert' tab.

Test

1 Why does your e-learning course need a title slide?
2 How do you add a new slide to a presentation?
3 Give reasons why you chose the theme for your presentation.
4 How will you carry out research for your e-learning lesson?

2.5 Create an e-learning slide

In this lesson

You will learn:

- use a presentation package to create a lesson.

In the last lesson, your team decided on a topic for the e-learning course. You each agreed to complete a slide for your group's e-learning course.

In this lesson you will create the lesson slides. Before you open your presentation software, do some planning and research for your lesson.

Activity

Write down the title of the e-learning course.

Underneath, write the title of your lesson.

Spiral back

In Student Book 5, you learned how to use key words to search for information on the web. Here is a reminder of the guidelines for using key words.

Make sure you understand the question you need to answer. Before you enter anything into the search engine:

- make a list of key words
- underline the most important key words
- make sure that the question you type in the search engine contains the important key words. Your question does not need to include punctuation or short joining words like 'is' and 'the'.

Research your content

Before you do any research on the web, write down what you already know about the lesson topic.

Activity

Make a list of short points that you already know about the lesson topic.

Add them under the lesson title you wrote down in the last activity.

Once you have written down what you know, it is time to do some research. There may be other facts that you want to include in the lesson.

Search the web

Before deciding what you will include in your lesson, search the web.

Look for:

- new facts. Your search may find facts you had forgotten to include in your list and some facts you weren't aware of.
- an image to illustrate your lesson. The image currently used in this lesson (a computer keyboard) isn't very helpful. In this case, it may have been more useful to include an image where the main parts of the keyboard are labelled.

Remember to check facts that you find on the web before you include them in your lesson. One way to check facts is to see if you can find the same fact on another website.

Remember that you will need to provide a credit in your presentation for the image that you choose. Make a note of the website you took the image from. You can use that information to credit the image.

Activity

Search the web for facts about your lesson topic. Choose two or three facts. Add them to the list of points you created in the earlier activity.

Search the web for an image that you can use in your lesson. Save the image.

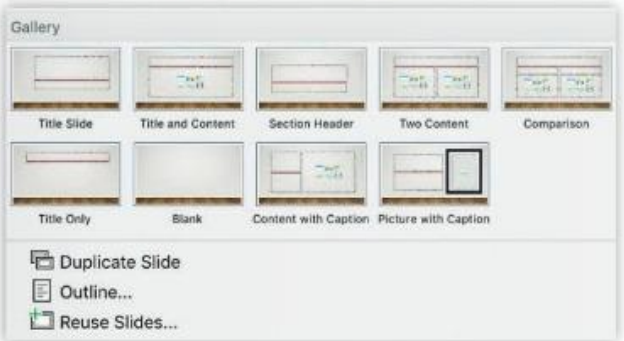
If you find more than one good image, save them all. You can decide which one to use later.

Create your slide

When you have all your lesson information, copy it into a slide. Remember to work as a team to make sure your slides all look the same.

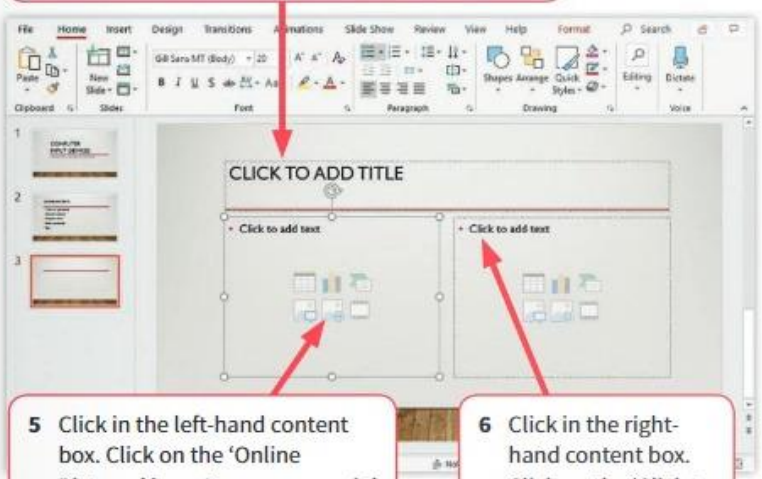
1 Open your presentation software.

2 Open a new slide. Select a template. The example in this lesson uses a template called 'Two Content'. The template has a space for a title. Below the title are two content boxes. You will add text into the right-hand content box and an image in the left-hand content box.



3 Add the theme you agreed with your team. You can then see how your content will look. Your page is ready for you to enter content.

4 Click in the title box. Type the title of your lesson. In our example, the title is 'Computer keyboard'.



5 Click in the left-hand content box. Click on the 'Online Pictures' icon. It opens a special browser that lets you search for pictures. Add an image.

6 Click in the right-hand content box. Click on the 'Click to add text' link. Add the text for the slide.

7 Save your file.

Your completed slide should look a bit like this.

Activity

Use the information you have gathered to create a slide with facts and an image. Use the layout shown in this example.

Work with the other students in your group to make a presentation which includes all the slides you made.

Extra challenge

Look through the whole presentation. Say what you like best about the other slides in the presentation. Finally, think if there are any ways that the presentation could be improved. If you have time, make those changes.

Test

1. Why is it important to identify key words before you do a web search?
2. What is a template used for in presentation software?
3. Why do you need to include a credit for images or content you use from websites?
4. Why is it important to check facts you find on the web before using them?

2.6 Add a test question

In this lesson

You will learn:

- how to add a test to your e-learning lesson.

Making a test

In Lesson 2.3 you learned that e-learning courses usually contain tests. This is so that students can check that they have understood the course content. Most e-learning tests are marked automatically by the computer. This means that you know straight away if you have given a correct answer. It also means that the teacher or trainer doesn't have to spend time marking the test.

Multiple-choice tests

The test you are going to write is called a multiple-choice test. Each multiple-choice question has more than one possible answer (usually four). Only one answer is correct. Here is an example.

The answer is B, Paris.

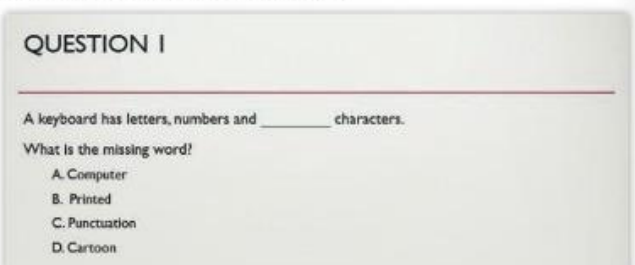
There is usually only one correct answer to a multiple-choice question. Simple questions like this one are easy for a computer to mark. If the answer is B, it is correct. Any other answer is incorrect.

Making an example question

Now you will make a test to find out whether students have understood the information in your lesson.

Work as a group to make a one-question test. It can be about any topic in your e-learning presentation. A student must be able to answer the question from information in the presentation.

The example lesson was about the computer keyboard. Here is an example test question. In this case, the question tests information from the third bullet point in the lesson.


QUESTION 1

A keyboard has letters, numbers and _____ characters.

What is the missing word?

A. Computer
B. Printed
C. Punctuation
D. Cartoon

Activity

Add a new slide. Select a 'Title and Content' slide.

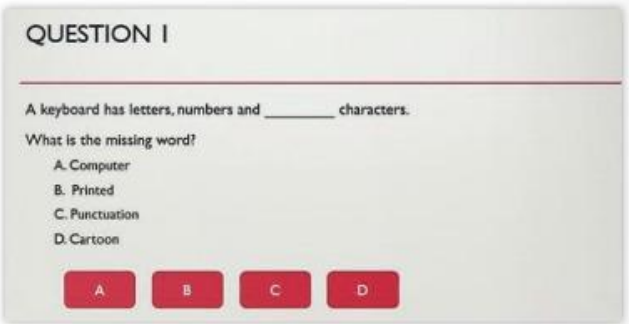
Type 'Question 1'. This is the title for your slide.

Type 'A keyboard has letters, numbers and _____ characters' followed by 'What is the missing word?'

Add four answers, like in the example: A. Computer, B. Printed, C. Punctuation and D. Cartoon.

Adding answer buttons

Once you have written your question, you must provide a way for students to give their answer. You will add buttons for the student to click.



Activity

Go to the 'Insert' tab and click on 'Shapes'.

Select a rectangle from the 'Shapes' menu. Draw a rectangle below your question.

Copy the first rectangle and paste it three times. That way you will get four buttons that are the same size. Insert the letters A, B, C and D on the boxes.

Making the buttons work

You now have a question slide with buttons to allow a student to give their answer.

Now you will add actions to each button to make them work.

The slides contain emoticons (icons showing facial expressions). If you want to add emoticons to your slides, click 'Icons' in the 'Insert' tab. The emoticons are under 'Faces'.

Activity

First, add three slides to your presentation. Add them in this order:

1. A slide with the heading 'That's right!' and a button that has the label 'Next question'.
2. A slide with the heading 'Wrong answer' and two buttons labelled 'Try again' and 'Next question'.
3. A slide with the heading 'Question 2'. This page doesn't need any other content.

Adding links

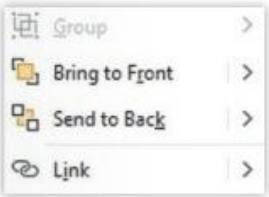
Now you are ready to add links that will tell your students whether they have the right answer.

Go back to the Question 1 slide you made earlier.

Activity

Right-click on the button labelled 'A'. A box should appear around the button. A menu will appear on-screen.

Click 'Link'. You will find this about half-way down the menu.

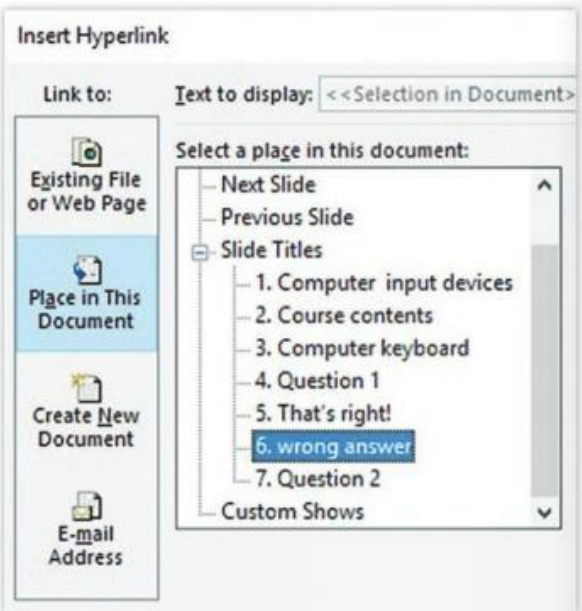


A box will open on your screen. There is a button on the left that says 'Place in This Document'. Click that button. A list of the slides you have made appears to the right of the button.

Choose the slide that you want button A to link to. Answer A is the wrong answer, so you should pick the Wrong answer slide.

Clicking button A on the Question 1 slide should now take you to the Wrong answer slide. Test it works by running your presentation. (Click the 'From Beginning' link in the 'Slide Show' tab.)

Once the link works, add links to the other buttons in your question slide. Here are the links you need to make.



Slide title	Button	Link to slide
Question 1	C and D	Wrong answer
Question 1	B	That's right!
That's right!	'Next question'	Question 2
Wrong answer	'Next question'	Question 2
Wrong answer	'Try again'	Question 1

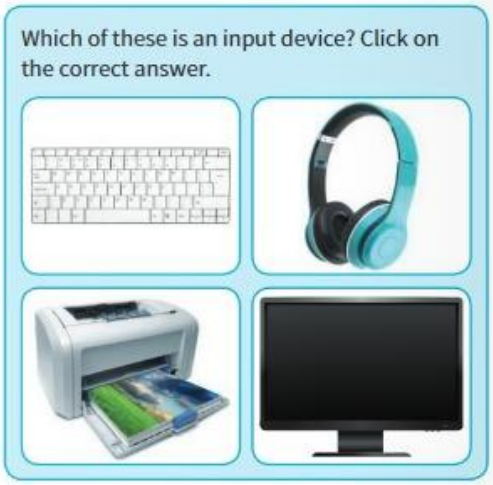
Other question types

You have created a multiple-choice question and answers in this lesson.

You could use other types of questions in your test. Here are a couple of ideas.

- True/False. You could give a statement such as: 'A printer is an input device.' Then ask the student to say whether the statement is True or False. This is a type of multiple-choice question but with only two options.
- Use pictures instead of words as possible answers. The picture on the right shows an example.

A picture question doesn't need buttons. You can add links to the pictures. Use the same method that you used for the A, B, C and D buttons.



Activity

This lesson showed you how to make a test question based on the computer keyboard example. Working as a group, make one test question that matches your e-learning lesson.

Extra challenge

Add more questions to the end of the presentation. You can do this working as a group, or each person can make their own slides and then copy them all into the group file.

Test

1. How do you run a slide show? What happens on your screen?
2. How many correct answers can you have in a multiple-choice test?
3. Explain how a link works in a presentation slide. Give an example of how you use a link.
4. Outline three types of test you could use. Why would you use different types of test?

Check what you know

You have learned

- how computers are used to help people learn
- how computers are used to make discoveries
- how to use online research to help complete a project.

Try the test and activities. They will help you to see how much you understand.

Test

1 Outline two ways that you have used computers to learn and find things out in subjects like science or geography.

2 Identify two types of natural disaster that can be researched using computers.

3 Describe how computers have been used in space exploration.

4 Give two advantages of using simulations for learning.

5 Describe how assistive technology helps people with disabilities to learn.

6 Describe how technology has been used to discover how natural events like hurricanes work. How have these discoveries benefited people?

Activities

In Lessons 2.4 to 2.6 you worked with a group to create a short e-learning course. Think back to that task and complete the following activities.

1. Give at least two examples of content or facts you found online when doing research for your e-learning lesson.
2. List the content items you used in your lesson. Where did you find the information? Give reasons why you chose that content. List one item of content you found that you decided not to use in your lesson. Why did you reject it?
3. Describe some of the presentation features you used to make the e-learning effective.

Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1.
- I answered test questions 1–4.
- I completed activities 1 and 2.
- I answered all the test questions.
- I completed all the activities.

Re-read any parts of the unit you feel unsure about. Try the test and activities again – can you do more this time?

3 Computational thinking: Team Manager

You will learn

- how to store a series of items as a list
- how to add, delete and edit items in the list
- how to look at every value in (traverse) a list
- how to make a menu interface to help the user
- how to block bad input to stop the program from crashing.

In this unit you will make a program in Python. The program can be used by anyone who manages a team or group of people. For example, the manager might be a team coach, an orchestra leader or a teacher. The program will help the manager make a team list. They will be able to add names, change them, or delete names from the list. They will be able to print out the list.

You will make a menu of choices to help the user work with the program. You will add input checks so the user cannot crash the program with bad inputs.

Unplugged

Imagine you are the manager of a team or group. Pick something you enjoy. It does not have to be a sports team – for example, you could be managing a group of actors, a horse-riding club or a choir.

On paper, design the interface for a phone application (app) that you could use to help you do your work as manager. Your interface can have:

- pictures
- a menu
- maps.
- text
- buttons

Include anything that you think could be useful.

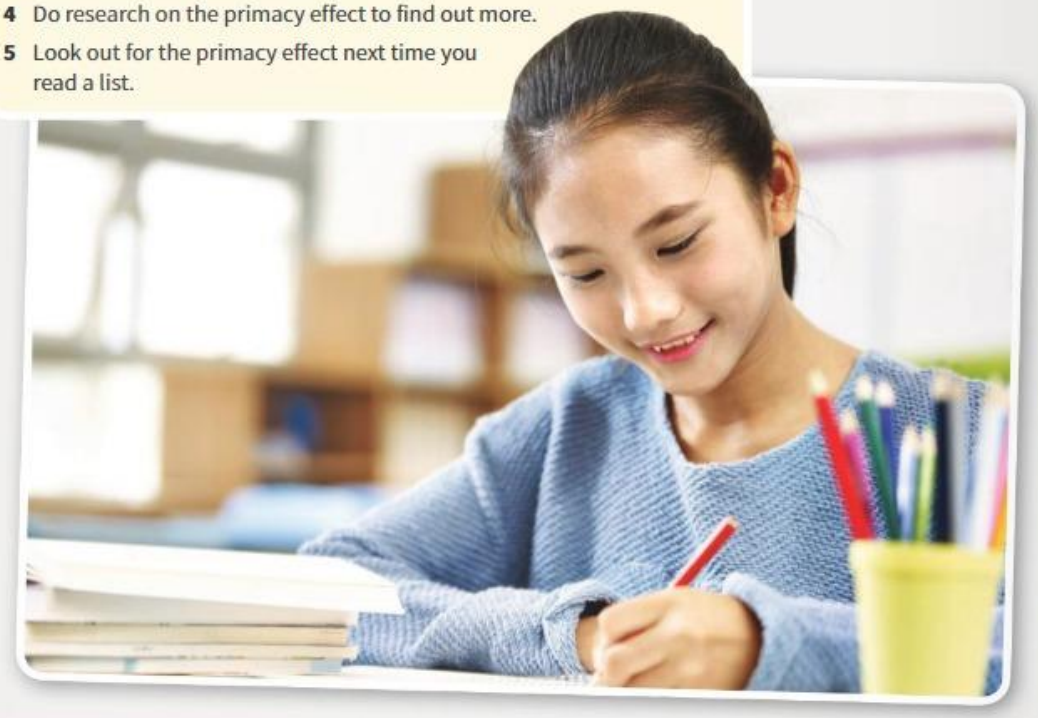
Learning outcomes: Write a program that processes a data structure (for example, a list)

Did you know?

Putting facts into lists helps people to read and remember facts.

The order of the list also matters!

1. People tend to think that the first fact in a list is the most important.
2. People remember the first fact better than the other facts on the list.
3. This is called the 'primacy effect'.
4. Do research on the primacy effect to find out more.
5. Look out for the primacy effect next time you read a list.



Talk about...

In this unit you will make a simple list of names. In real life, a manager would store many other pieces of information as well as names. What information does your school store about you? What other organisations have information about you?

list
element append index number
out of bounds error robust program
validation data structure
traversing stop value interface
nesting

3.1 Make a list

In this lesson

You will learn:

- how to store a series of values as a list variable.

Spiral back

Last year you learned to write programs in Python that are user-friendly and readable. You learned to use variables and to make programs with input and output. You learned to use `for` loops and `while` loops. You will use these Python skills as you make the Team Manager program. All of these skills and commands are important for this lesson. If you do not remember these Python commands, go back and review the work you did in Student Book 7.

Variables and lists

You have learned that a variable is a named area of storage. A variable stores one value or piece of data. Here are two Python commands which assign values to variables:

```
number = 5
colour = "red"
```

A **list** is a special kind of variable. A list can store several different items of data. A list is shown inside square brackets.

A computer company sold devices in a range of colours. This Python command will make a list of those colours. The list is called `colourlist`.

```
colourlist = ["red","yellow","blue","green"]
```

Each item in a list is called an **element**. This list has four elements. The elements are separated by commas.

If a list is long, it can go over more than one line of the program.

```
colourlist = ["red", "blue", "green", "orange", "purple",
              "pink", "brown", "teal", "scarlet", "grey",
              "crimson", "mauve", "magenta", "amber", "black"]
print(colourlist)
```

Append an element

Append means 'add to the end'. You can append items to a list. That means a new item will be added to the list. This command will append `"orange"` to `colourlist`.

```
colourlist.append("orange")
```

Append user input

You can get input from the user and append that value to the list.

```
colour = input("Enter a colour ")
colourlist.append(colour)
```

Remember to make input commands user-friendly. You learned about this in Student Book 7. An input command should include a clear prompt. In this example, 'Enter a colour'. It is good practice to include a space at the end of the input prompt, inside the quote marks. That means there will be a space between the text of the prompt and the input typed by the user.

Print a list

You can print a list just like any other variable.

```
print(colourlist)
```

This command will print out the whole list, including the brackets and commas.

The completed program

Here is the completed program. It combines all the commands shown so far.

```
colourlist = ["red","yellow","blue","green"]
colourlist.append("orange")
colour = input("Enter a colour ")
colourlist.append(colour)
print(colourlist)
```

Team Manager program

A football manager wants to use Python to record his team selection. Here is a command that makes the list. `teamlist` is empty. It has no elements.

```
teamlist = []
```

This command will add 'Jamie' to `teamlist`.

```
teamlist.append("Jamie")
```

Or you can get input from the user and add that input value to the list.

```
name = input("Enter a name ")
teamlist.append(name)
```

Activities

1 Make and run the `colourlist` program shown in this lesson.

2 Make and run a program which:

- **a** creates an empty list called `teamlist`
- **b** gets a name from the user
- **c** appends the name to the list
- **d** prints the list.

Append elements using a loop

The programs you have made add one element to the list. They:

- use an input command to get a value from the user
- append that value to the list.

Often you want to add more than one element. A loop is a program structure that repeats commands. By using a loop structure, you can add multiple elements to the list.

There are two types of loop in Python:

- `for` loop (counter loop)
- `while` loop (conditional loop).

Using a 'for' loop

A `for` loop is controlled by a counter. It is a counter-controlled loop. You learned to use counter-controlled loops in Student Book 7. Most programmers name the counter `i`. The program will still work if you use another name for the counter. But using `i` makes your programs more readable.

When you write the program, you set the number of times the loop will repeat. You can use a `for` loop if you know exactly how many elements you want to add to a list.

Here is an example.

```
colourlist = []
for i in range(7):
    colour = input("Enter a colour ")
    colourlist.append(colour)
print(colourlist)
```

This program adds exactly seven values to `colourlist`.

Activities

1 Make and run the `colourlist` program using a `for` loop.

2 Make and run a program which:
- creates an empty list called `teamlist`
- uses a `for` loop to append 11 names to the list
- prints the list.

Using a 'while' loop

A `while` loop is controlled by a logical test. It is a condition-controlled loop. When you write the program, you set a logical test. When the test is False the loop will stop. You can use a `while` loop if you don't know exactly how many elements you want to add to a list.

Here is an example. This program adds values to `colourlist`. The loop is controlled by a question:

Do you want to add another? (Y/N)

If the user types 'Y', the loop will repeat. If they type anything else, the loop will stop.

```
colourlist = []
repeat = "Y"
while repeat == "Y":
  colour = input("Enter a colour ")
  colourlist.append(colour)
  repeat = input("Do you want to add another? (Y/N) ")
print(colourlist)
```

Activities

1 Make and run the `colourlist` program using a `while` loop.

2 Make and run a program which:

- creates an empty list called `teamlist`
- uses a `while` loop to append names to the list
- prints the list.

Test

This command creates a list:

```
trees = ["oak","beech","willow"]
```

1 How many elements are in this list?

2 Write the command to append the value `"sycamore"` to the list.

3 Write the commands to get user input and add it to the list.

4 You want the user to add elements to the list, but you are not sure how many elements. What type of loop would you use?

Extra challenge

In this lesson you have made two programs that use loops. One program used a `for` loop. A `for` loop repeats a set number of times. You set the number to 11 so the program adds exactly 11 names to the list.

Now make a program that:

- asks the user how many names they want to add
- loops exactly that number of times.

3.2 Work with list elements

In this lesson

You will learn:

- how to identify list elements using an index number
- how to print, edit and delete list elements.

List elements

You have learned that a list is a series of elements.

A student made a list of goals. It had three elements.

```
goals = ["pass exams","help in shop", "buy trainers"]
```

You can choose any goals that are right for you.

The elements are separated by commas. Each element stores one value or piece of data. Every element in a list has its own name.

Index numbers

The name of an element is made of two parts:

- the name of the list
- the position of the element in the list.

List numbering starts at 0. The first element in the list is called:

```
goals[0]
```

The next element is called:

```
goals[1]
```

And so on.

The number in square brackets is called the **index number**. The index number tells you the position of an element in the list.

Print an element

Each element of a list is a variable in its own right. You can print the whole list. Or you can print a single element.

```
goals = ["pass exams","help in shop","buy trainers"]
print(goals)
print(goals[0])
```

You can try out these commands in the Python Shell, or make them into a program and run the program. You can include any goals that you like.

Choose which element to print

The program above prints out the element with index number 0. You can change the program so that the user decides which element to print.

```
goals = ["pass exams","help in shop","buy trainers"]
i = input("which goal do you want to print? ")
i = int(i)
print(goals[i])
```

Here is an example of the output of this program.

```
which goal do you want to print? 2
buy trainers
```

Edit a list element

Remember that **edit** means make changes to something. You can change a single element of a list. You can give it a new value. This program changes the final element of the list to the value `"buy a bicycle"`.

```
goals = ["pass exams","help in shop","buy trainers"]
print(goals)
goals[2] = "buy a bicycle"
print(goals)
```

This program changes the final element of the list to a value input by the user.

```
goals = ["pass exams","help in shop","buy trainers"]
print(goals)
goals[2] = input("enter a new goal ")
print(goals)
```

Try these commands in the shell or as a program.

Choose which element to edit

The program above edits the element with index number 2. You can change the program so that the user decides which element to edit.

```
goals = ["pass exams","help in shop","buy trainers"]
print(goals)
i = input("which goal do you want to change? ")
i = int(i)
goals[i] = input("enter a new goal ")
print(goals)
```

Here is an example of this program, run in the Python Shell.

```
['pass exams', 'help in shop', 'buy trainers']
which goal do you want to change? 0
enter a new goal learn to drive
['learn to drive', 'help in shop', 'buy trainers']
```

The Python Shell always uses single quote marks at the start and end of a string.

Delete an element

You can delete elements from a list. The command to delete is `del`. This program deletes element 0.

```
goals = ["pass exams","help in shop","buy trainers"]
print(goals)
del goals[0]
print(goals)
```

Try these commands in the shell or as a program.

Choose which element to delete

The program deletes the element with index number 0. You can change the program so that the user chooses which element to delete.

```
goals = ["pass exams","help in shop","buy trainers"]
print(goals)
i = input("which goal do you want to delete? ")
i = int(i)
del goals[i]
print(goals)
```

Example program

Here is a complete Python program. It lets the user make and edit a list of goals. It combines commands you have learned in Lessons 3.1 and 3.2.

```
goals = []

#append values
for i in range(5):
    new = input("add a new goal to the list ")
    goals.append(new)

#edit an item
print(goals)
i = input("which goal do you want to change? ")
i = int(i)
goals[i] = input("enter a new goal")
print(goals)

#delete an item
i = input("which goal do you want to delete? ")
i = int(i)
del goals[i]
print(goals)
```

The program includes comments. Comments begin with the # symbol. The computer ignores comments. They are there to help human readers understand the program.

Activities

1 Make and run a program to create and edit a list of goals.

2 Make and run a program to:

- create a team list of 11 names (using any method you know)
- allow the user to edit and delete names from the team list.

Extra challenge

Amend the program that you made to make and edit a team list. Use a `while` loop so the edit and delete commands are repeated.

Test

This command creates a list:

```
trees = ["oak","beech","willow"]
```

1 What is the value of element 0?

2 Give the command to delete element 2.

3 Give the command to change element 1 to `"eucalyptus"`.

4 The user entered this command:

```
trees[6] = "pine"
```

This caused an error. Explain why.

Digital citizen of the future

In this lesson you wrote a program using a list of names. You may have used made-up names, or the names of people you know. Many computer programs store details of real people. If you store information about a real person you must look after it carefully. You must be careful that the information is true. You must make sure the information is kept private. In many countries there are laws to protect personal data stored on computers. But even in countries without these laws, it is the responsibility of a programmer to treat personal data with care.

3.3 Block bad input

In this lesson

You will learn:

- how to recognise, avoid and fix out of bounds errors
- how to block bad input to the program.

Spiral back

Last year you learned about input validation. Input validation lets you block incorrect inputs. In this lesson you will write code to block incorrect input to the program.

Using index numbers

You can print, edit or delete a single element of a list. To do this, you must give the index number of the element.

The index number can be included in the code. Here is an example.

```
colourlist = ["red","yellow","blue","green"]
print(colourlist[2])
```

Or the index number can be input by the user. Here is an example.

```
colourlist = ["red","yellow","blue","green"]
i = input("which colour do you choose? ")
i = int(i)
print(colourlist[i])
```

But only some numbers will work. For example, if the list has four items, they are numbered from 0 to 3. If you give an index number that is bigger than 3, then the program will crash. For example, this user typed number 7. There isn't an element number 7. So they saw an error message.

```
which colour do you choose? 7
Traceback (most recent call last):
  File "C:/Users/Alison/Documents/Python/temp.py", line 4, in <module>
    print(colourlist[i])
IndexError: list index out of range
```

This type of error is called an **out of bounds error**. In this lesson you will see how to avoid and fix out of bounds errors.

Length of a list

The Python function `len()` will tell you the length of a string or a list.

Type this in the Python Shell:

```
len("hello")
```

You will see the integer 5, because the string `"hello"` has five characters.

Now try these commands in the Python Shell.

```
colourlist = ["red","yellow","blue","green"]
listlength = len(colourlist)
print(listlength)
```

You should see the integer 4, because `colourlist` has four elements.

Think about how you can use this information to stop the user from making an out of bounds error.

What numbers are allowed?

If we know the length of a list, we know what numbers are allowed. The numbers go from 0 to one less than the number of items in the list.

Here are some examples.

- If a list has 10 items, the index numbers go from 0 to 9.
- If a list has 100 items, the index numbers go from 0 to 99.

We can make a program to print this information.

```
colourlist = ["red","yellow","blue","green"]
final_value = len(colourlist) - 1
print("index numbers go from 0 to", final_value)
```

Try these commands in the Python shell or as a program.

Helpful message

One way to reduce errors is to give the user helpful information. Then they know what they have to type.

This message is not helpful. It is easy for the user to make a mistake.

```
you can print one colour from the list
which colour do you choose?
```

Here is the output of a program that does have a helpful message. The message tells the user what numbers they can input.

```
you can print one colour
index numbers go from 0 to 3
which number do you choose?
```

The user of this program is much less likely to make a mistake.

Design the message

Remember to be user friendly. In this case, the message should tell users what range of values they are allowed to input.

Activities

1. Make and run a program that prints a single item from `colourlist`. Include a helpful prompt for the user.
2. In the last lesson you made one or more programs to create and edit a list. Open any program you made last lesson. Add a helpful prompt to make sure user input is correct.

Validation

What if the user still makes a mistake? The program will crash.

A program which does not crash is called a **robust program**. The user might input bad data. But a robust program will not crash.

To make sure your program is robust, you can add a command that blocks bad input. A check which blocks bad input is called a **validation** check.

'If... else'

One way to do validation is to use an `if... else` structure.

- An `if` structure begins with a logical test. Test if the number input by the user is smaller than the length of the list.
- If the test is True, carry out the command.
- If the test is False, show an error message.

Here is an example.

```
colourlist = ["red","yellow","blue","green"]
i = input("which colour do you choose? ")
i = int(i)
if i < len(colourlist):
    print(colourlist[i])
else:
    print("out of bounds error")
```

'While' loop

There is an alternative to using `if... else`. You can use a `while` loop instead. A `while` loop will ask for input until the value matches the requirement of the program. Here is an example.

```
colourlist = ["red","yellow","blue","green"]
i = input("which colour do you choose? ")
i = int(i)
while i >= len(colourlist):
    i = input("enter a number ")
    i = int(i)
print(colourlist[i])
```

This program uses the relational operator `>=`. This operator means 'greater than or equal to'.

You will get an out of bounds error if you enter a number:

- that is greater than the length of the list
- that is equal to the length of the list.

Remember that list numbering starts at zero. For example, think of a list with four elements. The elements will be numbered 0, 1, 2, 3. In a list with four elements, the numbering stops at 3. In a list with 100 elements, the numbering stops at 99.

To summarise – an index number that is equal to the length of the list will cause an out of bounds error. The `while` loop in the program will loop until you enter a value that will not cause this error.

Activity

Make and run the example program shown in this lesson that uses `if... else` to block bad inputs to the program.

Change the program you have made so that it uses a `while` loop.

Extra challenge

In the last lesson you made two programs to create and edit a list. Open any program. Use `if... else` to block bad input to this program.

Adapt the program to use a `while` loop instead of `if... else`.

Test

This command creates a list:

```
trees = ["oak","beech","willow"]
```

1 What would be the output of these commands?

```
number = len(trees)
print(number)
```

2 Write a print command that will tell the user how to enter a valid index number.

3 Write the commands to get an index number from the user. Include a helpful message, for example, the one you thought of for Q2.

4 A programmer wanted to check whether the user input was out of bounds. Write a logical test which will be True if the index number is valid.

Explore more

Investigate user interfaces of real-life software apps. That might include apps you use on your phone. It might also include software you use at home or at school. Here are some questions to ask yourself.

- Are there clear messages?
- Do you know what input you have to provide?
- If you enter the wrong input, might you crash the program?

Ask family and friends about the software they have used, for example, at work. Studies have shown that a good user interface has a big impact on whether people like to use software.

Reflecting on your experiences with software will help to improve your programming skills.

3.4 Traverse a list

In this lesson

You will learn:

- what traversing a list is
- how to traverse a list.

Data structures

A list is an example of a **data structure**. A data structure is a type of variable that can hold many values. In a list data structure, the different values are called elements. Lists and other data structures are used a lot in programming.

That is because in real life we often want to store many data values. Here are some examples.

- A social media app such as Facebook or Twitter wants to store information about all the different accounts.
- A business wants to store information about its products, employees and suppliers.
- A team manager wants to store information about the members of the team.

And there are many other examples. Data structures are useful when we want to store a lot of data values. Some data structures can store millions of values.

You have written programs to work with lists. You know how to append, delete and edit the elements of a list. If you are not sure how to do these things, look back at previous lessons.

How to traverse a list

Now you will learn how to traverse a list.

Traversing a data structure means looking at every value in the data structure. Think of the reasons people might want to do that.

- A social media app might want to check that every user is still active.
- A business might want to print out the number of items in stock.
- A team manager might want to write an email to every team member.

In this lesson you will traverse a team list. You will print out each name in the list. This is a good way to show that you have traversed the list. Remember, real-life programs usually do something more complicated than just printing out.

'For' loop

A `for` loop is a counter-controlled loop. The loop repeats a set number of times. You can use a counter-controlled loop when you know exactly how many times you want a loop to repeat.

A `for` loop has a counter. The counter starts at zero. It stops when it reaches the value set at the top of the loop. The number that stops the loop is called the **stop value**.

In Lesson 3.1 you used a `for` loop to append values to a list. Here is an example. This program adds exactly four values to `colourlist`. The stop value is 4.

```
colourlist = []
for i in range(4):
    colour = input("Enter a colour ")
    colourlist.append(colour)
print(colourlist)
```

Print using a 'for' loop

Each time the loop repeats, the value of `i` goes up by one. A `for` loop with a stop value of 4 counts through these values:

```
i = 0
i = 1
i = 2
i = 3
```

These are the same values as the index numbers of a list. This makes it easy to print out every element in a list.

Example: colourlist

This command creates a list with four elements.

```
colourlist = ["red","yellow","blue","green"]
```

Because the list has four elements, we can use a `for` loop that counts to 4.

```
for i in range(4):
```

Each time round the loop, we print a single element of `colourlist`.

```
print(colourlist[i])
```

The value of `i` goes up by one each time. The program prints the next element in the list. When the value of `i` reaches 4, the loop stops.

You can also adapt the command so that it prints out the index number as well as the colour.

```
print(i, colourlist[i])
```

Here is the completed program.

```
colourlist = ["red","yellow","blue","green"]

for i in range(4):
    print(i, colourlist[i])
```

Activities

1 Make and run the example `colourlist` program shown in this lesson, using a `for` loop to traverse the list.

2 Make and run a program which:

- creates an empty list called `teamlist`
- uses a `for` loop to append 11 names to the list
- traverses the list, printing out each name in turn.

Wrong stop value?

The programs you have made work fine. When we write the program, we know how many elements there are in the list. We can use this number as the stop value.

But in real life we don't always know the size of a list. We could use a `while` loop to append values to a list, which means we don't know how many values there will be.

A list can also change in size.

- The user may append values to the list. This will make the list bigger.
- The user may delete values from the list. The list will get smaller.

This could make the program go wrong. The stop value won't match the size of the list. What effect would this have?

- If the list was bigger than the stop value, then the loop would stop too soon. The loop would miss out some of the values in the list.
- If the list was smaller than the stop value, then the loop would go on too long. There would be an out of bounds error (look back at Lesson 3.3).

Find list length

The solution is to find the length of the list. You already know the command to do this. Store the list length as a variable. In this example we have called the variable `stop`.

```
stop = len(colourlist)
```

Now we can use the variable as the stop value of the loop.

```
colourlist = ["red","yellow","blue","green"]
stop = len(colourlist)
for i in range(stop):
    print(i, colourlist[i])
```

Activity

Make a program that creates a list of colours.

- Use the `len()` function to find the stop value.
- Use a `for` loop to traverse the list, printing out each value.

Extra challenge

Make and run a program which:

- creates an empty list called `teamlist`
- uses a `while` loop to append names to the list
- uses a `for` loop to traverse the list and print each element in the list.

Test

This command creates a list:

```
trees = ["oak","beech","willow","ash"]
```

1. We could use a `for` loop to traverse this list. What is the stop value?
2. Write a command that assigns the length of the list to a variable called `stop`.
3. A user wrote the following program to traverse this list. What is the error in this program? What is the output of this program?

```
trees = ["oak","beech","willow","ash"]
for i in range(3):
  print(trees[i])
```

4. Give one reason that a list can change in size while a program is running.

3.5 Menu of choices

In this lesson

You will learn:

- what an interface is
- how to make a menu interface.

What is a program interface?

A program **interface** is the part of a program that handles input and output.

- **Input:** The interface gets user inputs. The user can control the program.
- **Output:** The interface displays program outputs. The user can see results and content.

In Student Books 1–7 you used the Scratch programming language. Scratch provides a colourful interface with images and sounds. Python has a text-only interface.

Requirements

A program interface should have all of the following features. It should:

- tell the user what the program is and what it can do
- let the user make choices or select options
- let the user enter information – you should include a helpful message and validation (remember Lesson 3.3)
- display results or answers
- let the user close the program.

In this lesson you will make a menu interface for the Team Manager program.

Make the Team Manager interface

Start a new Python program file. Start with the command to make a new empty team list.

```
teamlist = []
```

Now you will make an interface to let the team manager work with the team list.

Introduce the program

First, make the part of the interface that meets this requirement:

Tell the user what the program is and what it can do.

This program is for the manager of a team, for example a football team or a music group. It will let them manage the membership list.

Here are the commands to do this. You can change the words if you like.

```
teamlist = []
print("T E A M  M A N A G E R")
print("======================")
print("This program will help you manage your team.")
print("\n")
```

The final command prints `"\n"`. That stands for 'new line' and it makes a blank line in your program.

Menu of choices

Now you will add commands for the next requirement:

Let the user make choices or select options.

The first version of the menu has only three options (A, B and X). Here is the menu design.

Add print commands to your program to print out options A, B and X, with the words you see here. Your aim is to make this menu appear when you run the program. If you want to change the wording or layout you can.

```
T E A M  M A N A G E R
======================
This program will help you manage your team.


A: Append a value
B: Print the team list
X: Exit the program
```

Get user input

After the menu has appeared, the user will make a choice. This command will get input from the user and store it as a variable called `choice`.

```
choice = input("Enter your choice: ")
```

Run the program and you will see the menu. Remember, the menu choices don't work yet.

```
teamlist = []
print("T E A M  M A N A G E R")
print("======================")
print("This program will help you manage your team.")
print("\n")
print("A: Append a value")
print("B: Print the team list")
print("X: Exit the program")
choice = input("Enter your choice: ")
```

Activity

Start a new Python program.

Use a series of print commands to make and display the Team Manager interface.

Get user choice with an input command.

Loop to repeat

The team manager may want to do many different actions. For this reason, you will use a loop to repeat the menu. But remember this requirement:

Let the user close the program.

You must have some way to stop the program when the team manager has finished using the menu.

You do not know exactly how many times the manager will want to look at the menu. That means you must use a conditional loop (a `while` loop). The menu will repeat until the user enters the choice 'X'.

What can go wrong?

A class made the Team Manager program. When they made it, some things went wrong. In this section we'll look at some of the mistakes they made.

Barbara's mistake

Barbara entered this code at the start of the `while` loop. The program did not work properly.

```
teamlist = []
choice == "X"
while choice == "X":
    print("T E A M  M A N A G E R")
    print("=====================")
    print("This program will help you manage your team.")
    print("\n")
    print("A: Append a value")
    print("B: Print the team list")
    print("X: Exit the program")
    choice = input("Enter your choice: ")
```

Barbara's program did not work because she used this logical test:

```
choice == "X"
```

The operator == means 'is equal to'. The loop will repeat when the user enters the value 'X'. But we want the opposite. We want the loop to repeat if the user enters a value which is not 'X'. The symbol for 'not equal to' is:

```
!=
```

Barbara had to change the first line of the loop to this:

```
while choice != "X":
```

If `choice` was not set to `"X"`, the loop would repeat.

Rio's mistake

Rio entered this code.

```
teamlist = []
while choice != "X":
    print("T E A M  M A N A G E R")
    print("=====================")
    print("This program will help you manage your team.")
    print("\n")
    print("A: Append a value")
    print("B: Print the team list")
    print("X: Exit the program")
    choice = input("Enter your choice: ")
```

The program crashed. Rio saw this error message.

```
Traceback (most recent call last):
  File "C:/Python/rio.py", line 2, in <module>
    while choice != "X":
NameError: name 'choice' is not defined
```

Rio's program crashed because the `while` loop uses the variable `choice`. But the variable `choice` has not been given a value yet. Rio had to change his program to give `choice` a value before the `while` loop. The variable can have any value, even a blank.

```
choice = " "
```

Kalif's mistake

Kalif entered this code. The program worked. But there is still a mistake. Can you see what it is?

```
choice = " "
while choice != "X":
    teamlist = []
    print("T E A M  M A N A G E R")
    print("----------------------")
    print("This program will help you manage your team.")
    print("\n")
    print("A: Append a value")
    print("B: Print the team list")
    print("X: Exit the program")
    choice = input("Enter your choice: ")
```

This command makes a new empty team list.

```
teamlist = []
```

Kalif's program was wrong because the command to make an empty list is inside the loop. It will repeat. The team list will go back to being empty each time. Kalif corrected the error by moving the command. He moved it so that it came before the loop.

Activity

Put the Team Manager menu inside a loop with 'X' to exit.

Avoid the errors you saw in the examples.

Extra challenge

Add code so that if the user enters the option 'A', they can append a new name to the team list.

Test

A programmer made a menu interface. It was inside a `while` loop.

This is the first line of the `while` loop:

```
while more == "Y":
```

1. What is the name of the variable used in this command?
2. What value could the user input to make the loop continue?
3. What line of code would you include before the loop to set the value of the variable?
4. Give an example of a line you would include inside the loop to let the user stop the loop.

3.6 Activate the menu choices

In this lesson

You will learn:

- how to give users control over what software does.

User interface

In the last lesson you made a menu. The user entered a letter. It was stored as a variable called `choice`.

Make sure you have completed the activities in the last lesson before you continue with the tasks in this lesson.

Think of different types of software. Each type of software has a different user interface.

When you touch or click on the interface, something happens in the program.

In this lesson, you will make that happen with your Team Manager program. You will add code to make your menu options work. The user will input a choice. The computer will carry out their choice.

'If' structure

You used an `if` structure in Lesson 3.3. An `if` structure starts with a logical test. The logical test compares two values. If the test is True, then the actions inside the `if` structure will be carried out.

Append

If the choice is 'A', then the user can append a name to the team list.

You already know the code to append a name to the list. You did this activity in Lesson 3.1.

Put the code into the `if` structure.

```
if choice == "A":
    name = input("Enter a name ")
    teamlist.append(name)
```

The code lines inside the `if` structure are indented.

Print

If the choice is 'B', then the computer will print out the team list.

You already know the code to print the list. You did this activity in Lesson 3.1. Put the code into the `if` structure.

```
if choice == "B":
    print(teamlist)
```

The code line inside the `if` structure is indented.

Double indentation

Here is the program so far.

```
teamlist = []
choice = " "

while choice != "X":

    print("T E A M  M A N A G E R")
    print("======================")
    print("This program will help you manage your team.")
    print("\n")
    print("A: Append a value")
    print("B: Print the team list")
    print("X: Exit the program)

    choice = input("Enter your choice: ")

    if choice == "A":
        name = input("Enter a name ")
        teamlist.append(name)

    if choice == "B":
        print(teamlist)
```

This program has got two `if` structures. The `if` structures are inside a `while` loop. When a program structure is inside another structure, this is called **nesting**. The `if` structure is 'nested' inside the `while` loop.

- Commands inside a `while` loop are indented.
- Commands inside an `if` structure are indented.

That means the program has double indentation. Can you see the double indentation in this program?

Activity

Continue the Python program you made in the last lesson. Add code so that the menu options work properly.

- Append a name.
- Print the list.

Remember to include double indentation.

In the next part of the lesson you will develop the program further. You will:

- add new menu options to print, delete and edit elements
- block bad input from the user to avoid an out of bounds error
- sort the list using a new Python function.

Make sure you have completed all earlier activities before you go on to the final part of the lesson. If you need more practice, take time to get the previous activities right before you go on to the final part of this unit.

Explore more

In Lesson 3.3 you explored software interfaces and reflected on whether they were user-friendly. Continue this task. But this time look out for software menus. What software do you know that uses menus? How do you choose an option from the menu? Do you find the menu easy to understand and use? Does the menu use words or pictures?

Add more menu options

In Lesson 3.2, you learned some extra commands which would be useful to the team manager.

- Print an element.
- Delete an element.
- Edit an element.

The table shows the commands that carry out these actions.

Action	Commands

Print an element	`i = input("which list item do you want to print? ")`
	`i = int(i)`
	`print(teamlist[i])`

Delete an element	`i = input("which list item do you want to delete? ")`
	`i = int(i)`
	`del teamlist[i]`

Edit an element	`i = input("which list item do you want to change? ")`
	`i = int(i)`
	`teamlist[i] = input("enter a new name ")`

Block bad input

The table shows the commands that will let the user print, delete or edit an element of the array. In each case the user has to enter a number. The number is the index of a list item. In this example the value is stored as a variable called `i`.

The value input by the user must be a valid index number. It can't be bigger than the biggest index number in the list. If it is too big it will cause an out of bounds error. For example, if the list had five items in it, and the user tried to delete item 8, that would cause an error.

In Lesson 3.3 you learned how to block bad input. You could use these skills to prevent the user from entering an index number that is too large.

Sort the list

This command will sort the team list.

```
teamlist.sort()
```

Because the elements of `teamlist` are strings, they will be sorted into alphabetical order. If they were numbers, they would be sorted into numerical order.

Use this knowledge to add a new option to the menu, to sort `teamlist`.

Extra challenge

Add code to implement one of the new menu options given in this lesson. This image shows all of the options – you don't have to do them all!

If you have time, do more than one.

If you have time, add a check to block invalid index numbers.

```
Enter your choice:
T E A M  M A N A G E R
======================
This program will help you manage your team.


A: Append a value
B: Print the team list
C: Print one element
D: Delete one element
E: Edit one element
F: Sort the list
X: Exit the program


Enter your choice:
```

Test

A programmer made a program. Here is an extract from the program.

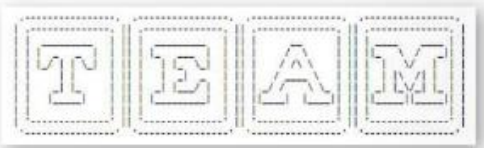
1. What choice should the user enter in order to print the list?
2. What happens if the user types 'A'?
3. If the user types 'A', what will they see next on the screen?
4. What happens if the user types 'C'?

```
colourlist = ["red","yellow","blue","green"]
choice = ""

while choice != "Z":

    print("A: Add a new colour to the list")
    print("B: Print the colour list")

    choice = input("Input your choice: ")

    if choice == "A":
        new = input("Type a new colour ")
        colourlist.append(new)

    if choice == "B":
        print(colourlist)

    if choice == "C":
        colourlist.sort()
```

Be creative

It is hard to make an interface using Python. That is because the program will only print simple output made of text characters.

Usain liked creative challenges. He made this title for the program, using nothing but Python print commands.



Can you design and make an interesting title for your Team Manager program?

Check what you know

You have learned

- how to store a series of items as a list
- how to add, delete and edit items in the list
- how to look at every value in (traverse) a list
- how to make a menu interface to help the user
- how to block bad input to stop the program from crashing.

Try the test and activities. They will help you to see how much you understand.

Test

Jamal was learning about prime numbers. He made a list of prime numbers. He used this command:

```
primes = [1,2,3,5,7,11]
```

1 How many elements are there in this list?

2 What is the command to append the value 13 to the list?

Jamal wanted to print out all the elements in the list, one after the other. He used this command:

```
for i in range (stop):
  print(primes[i])
```

3 If the variable `stop` had the value 13, there would be an error. What type of error?

4 What command could Jamal use to set the value of the `stop` variable?

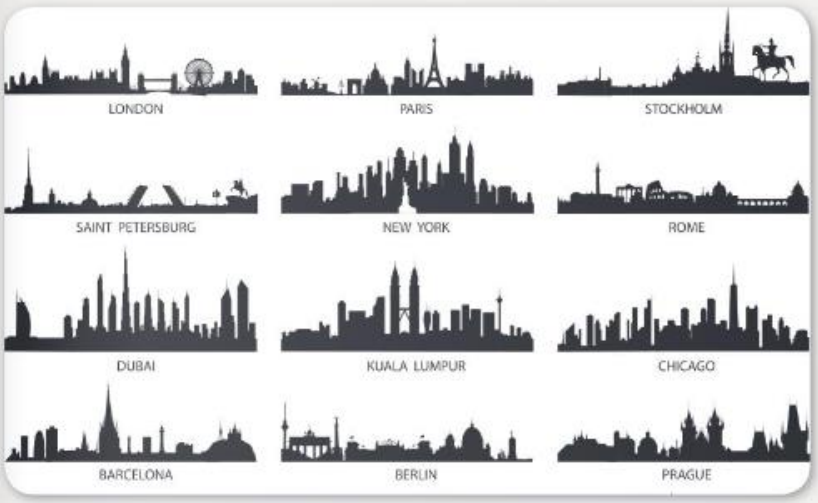
5 Write a set of commands which would prompt the user to enter a value and then append that value to the list.

6 Write a set of commands which would prompt the user to enter an index number and then delete that element from the list.

Activities

Una wanted to store the names of cities. She created a program that made a list.

1. Write a program to make a list of five cities. You can use some of the names shown here or think of new ones.
2. Add commands to traverse the city list, printing out every name one after the other.
3. Add commands to your program so the user can append a name to the city list.
4. Add commands to your program so the user can delete a name from the city list.
5. Add commands to print the list of cities to show these changes.



Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1.
- I answered test questions 1–4.
- I completed activities 1 and 2.
- I answered all the test questions.
- I completed all the activities.

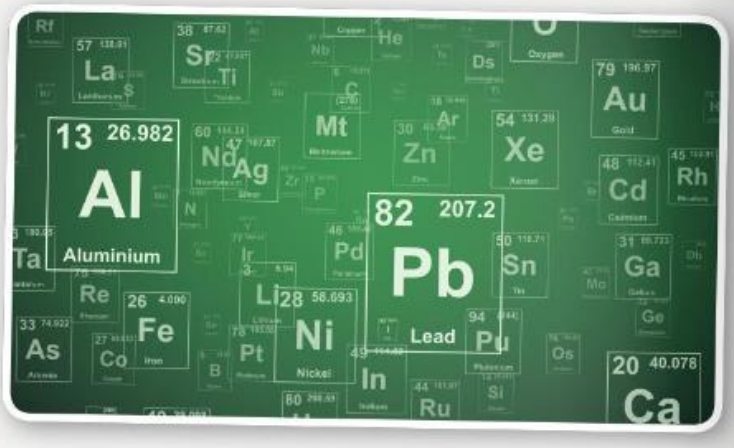
Re-read any parts of the unit you feel unsure about. Try the test and activities again – can you do more this time?

4 Programming: Atom Finder

You will learn

- how to store blocks of code as procedures and use them in programs
- how to create a procedure to find an element in a list
- how to compare algorithms used for searching a list.

In the last unit you worked with a list that stored the members of a team. In this unit you will work with a list of atoms. You will write a Python program to add and remove the names of atoms from the list. You will write a Python program to find an atom in the list.



Unplugged

The computer will look through the list. The computer can only look at one item at a time. This activity will help you to understand how hard it can be to find an item.

The students at City Park School played an 'atom search' game. They agreed on this list of atoms.

Hydrogen	Helium	Carbon	Nitrogen	Oxygen
Fluorine	Neon	Sodium	Magnesium	Aluminium
Silicon	Phosphorus	Sulfur	Chlorine	Argon
Potassium	Calcium	Iron	Cobalt	Nickel
Copper	Zinc	Silver	Tin	Antimony
Iodine	Platinum	Gold	Mercury	Lead

They wrote the names of the atoms on cards. They wrote each name on a different card. The students put the cards face down so the names were hidden. The atoms were not in sorted order.

Learning outcomes: Write a modular program that uses procedures or functions; Compare alternative algorithms to solve a problem (for example, searching)

The rule of the game is that you can only look at one name at a time. Then you have to turn the card face down again.

Tasha challenged Jasmin to find the search term 'Gold'. Jasmin turned over the first card – it did not show 'Gold'. Then she turned over the next card – it did not show 'Gold'. Jasmin had to keep turning cards over until she found the search term.

Try this game in your class. Keep the cards – you can use them again in Lesson 4.5.

Did you know?

Scientists use computers to study very small things such as atoms and sub-atomic particles. They also use computers to study the largest structures in the universe such as galaxies. The interstellar mystery of why stars form was solved by scientists using a massive computer simulation of a galaxy. The simulation was tested on a new and extremely powerful computer called the Stampede supercomputer.

Talk about...

In this unit you will use computers to hold and process science data. Computers can help us learn, not only in science but in any school subject. Would you like to use computers more at school?

header body
module function procedure
modular programming search term
linear search binary search
parameter

4.1 Amend a program

In this lesson

You will learn:

- how to adapt a program
- that program structures have a header and a body.

Spiral back

Last year, you learned to write programs in Python. You learned to use loops and conditional (`if... else`) structures. You will develop your skills in this unit. You will extend and adapt a Python program.

Atom List program

In Unit 3 you made a Team Manager program. It had a menu. The menu let you add and delete items from the team list. In this unit you will work with a similar program. It is called Atom List. This program manages a list of the atomic elements from hydrogen to radium.

The Atom List program has been prepared for you. Your teacher will give you a copy of this program.

Look at the program code

Open the Atom List program. Read through the code.

The program is in three sections. The sections of the program are marked using comments. Remember that comments are ignored by the computer. They are there for you to read.

The three sections of the program are:

- the list of atoms
- procedure definitions
- the main program.

The first part of the program makes a list. The list is called `atoms`. There are 88 elements in the list.

```
### the list of atoms
atoms = ["Hydrogen", "Helium", "Lithium", "Beryllium",
         "Boron", "Carbon", "Nitrogen", "Oxygen", "Fluorine",
         "Neon", "Sodium", "Magnesium", "Aluminium", "Silicon",
         "Phosphorus", "Sulfur", "Chlorine", "Argon", "Potassium",
         "Calcium", "Scandium", "Titanium", "Vanadium", "Chromium",
         "Manganese", "Iron", "Cobalt", "Nickel", "Copper", "Zinc",
         "Gallium", "Germanium", "Arsenic", "Selenium", "Bromine",
         "Krypton", "Rubidium", "Strontium", "Yttrium", "Zirconium",
         "Niobium", "Molybdenum", "Technetium", "Ruthenium", "Rhodium",
         "Palladium", "Silver", "Cadmium", "Indium", "Tin", "Antimony",
         "Tellurium", "Iodine", "Xenon", "Cesium", "Barium", "Lanthanum",
         "Cerium", "Praseodymium", "Neodymium", "Promethium", "Samarium",
         "Europium", "Gadolinium", "Terbium", "Dysprosium", "Holmium",
         "Erbium", "Thulium", "Ytterbium", "Lutetium", "Hafnium", "Tantalum",
         "Tungsten", "Rhenium", "Osmium", "Iridium", "Platinum", "Gold",
         "Mercury", "Thallium", "Lead", "Bismuth", "Polonium",
         "Astatine", "Radon", "Francium", "Radium"]
```

The next part of the program is called 'Procedure definitions'. This part of the program is empty. In the next lesson you will make procedure definitions in this part of the program.

```
### Procedure definitions
```

The final part of the program is called 'Main program'. This is the main part of the Python program. It has the code that displays the menu.

It also has the code that makes the menu options work.

```
### Main program

choice =""
while choice != "X":
    print("====================")
    print("A T O M  F I N D E R")
    print("====================")
    print("\n")
    print("A: Append an atom to the list")
    print("B: Remove an atom from the list")
    print("C: Print the list")
    print("X: Exit the program")
    print("\n")
    choice = input("Choose an option: ")
```

The code below appends a new atom name to the list of atoms. Find this code in the program.

```
if choice == "A":
    name = input("enter the name of an atom to add: ")
    atoms.append(name)
    print(name, "has been added to the list")
```

The code below removes an element from the list. Find this code in the program.

```
if choice == "B":
    name = input("enter the name of an atom to remove: ")
    atoms.remove(name)
    print(name, "has been removed from the list")
```

This block of code uses a new command: `remove`. This command deletes an element from the list. You give the name of the item. The computer looks through the list. If it finds an item that matches the name you gave, it removes it from the list.

Run the program code

Run the program. You will see the menu interface displayed in the shell.

By using the menu, you can work with the list of atoms.

- **If you type option 'A'** the program will ask you to enter the name of an atom. It will add this name to the list.
- **If you type option 'B'** the program will ask you to enter the name of an atom. It will remove this name from the list.
- **If you type option 'C'** the program will print the whole list.
- **If you type option 'X'** the program will close.

```
====================
A T O M  F I N D E R
====================


A: Append an atom to the list
B: Remove an atom from the list
C: Print the list
X: Exit the program


Choose an option:
```

Activity

Copy the Atom List program onto your computer. Open the program in the usual way.

Read through the code and look at the three sections of the program.

Run the program. Using the menu interface:

- add 'Uranium' to the list
- delete 'Iron' from the list
- print out the list. You will see the changes you made.

The changes to the list will not be permanent. Next time you load the program, the list will return to its original content.

Header and body

The Atom List program includes `if` structures. You have used `if` structures before in other programs.

A Python `if` structure is made out of a **header** and a **body**. The header has the commands that control the `if` structure. The body has the commands that are controlled by the header. Every time you make an `if` structure, you must make the header and the body.

The header

The header starts the `if` structure. It has:

- the word `if`
- a logical test
- a colon (two dots).

```
if choice == "A":
    name = input("enter the name of an atom to add: ")
    atoms.append(name)
    print(name, "has been added to the list")
```

Here is an example of a header from the Atom List program.

```
if choice == "A":
```

The body

The body of the `if` structure has all the other commands that belong to the structure. The commands in the body are indented to show that they belong inside the `if` structure. These commands will be carried out if the test in the header is True.

Here is an example from this program.

```
name = input("enter the name of an atom to add: ")
atoms.append(name)
print(name, "has been added to the list")
```

When will the computer carry out these commands?

Loop structure

The Atom List program includes a loop. You have used loops before in other programs.

A loop also has the header/body structure.

- **The header** of the loop has the command which controls the loop. It sets the exit condition of the loop. It might be a `for` loop or a `while` loop.
- **The body** of the loop has the commands which belong inside the loop. They are indented. These commands will be repeated. The header controls how many times the commands are repeated.

Double indentation

In this program there is a `while` loop. Inside the `while` loop there are `if` structures.

In Unit 3 you learned that:

- if one structure is inside another, this is called a nested structure
- a nested structure is shown using double indentation.

Look out for double indentation in this program.

Activity

Kassim extended the menu to include option 'D' to sort the list.

Add a new `print` command to your program to display this new menu option.

Add a new `if` structure to the program to make the new menu option work.

```
if choice == "D":
```

Here is the command to sort the list. Put this command inside the new `if` structure.

```
atoms.sort()
```

```
Choose an option:
==================
A T O M  F I N D E R
==================


A: Append an atom to the list
B: Remove an atom from the list
C: Print the list
D: Sort the list
X: Exit the program
```

Extra challenge

In Unit 3 you learned the command to print a single element of a list. Add an option to the menu which allows the user to do this. Add an `if` structure to make this extra menu option work.

Test

1. Python structures have a header and a body. The header always ends with what symbol?
2. How does Python show that commands belong inside a Python structure?
3. How does Python show that one structure (for example, an `if` structure) is nested inside another (for example, a loop)?
4. In Unit 3 you learned about comments. Give an example of a comment in the Atom List program.

4.2 Make and use procedures

In this lesson

You will learn:

- how to define a procedure
- how to call a procedure in a program.

What is a procedure?

A **module** is a ready-made block of code, with a name, that you can use in your program. If you put the name of the module into your program, all the commands in the module will be carried out. That makes your programs shorter and easier to write.

A **procedure** is a type of module that you can make yourself. In this lesson, you will make and use procedures. You will write a new program to practise using procedures. Then you will add procedures to the Atom List program.

How to use procedures

You can use procedures in any program. You have to do two things.

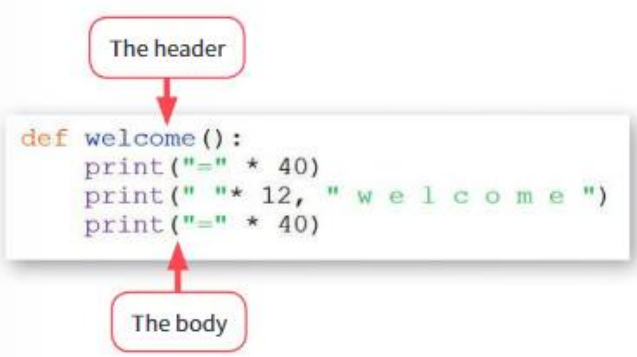
1. **Define the procedure.** Give the procedure a name. Store commands inside the procedure.
2. **Call the procedure.** Put the name of the procedure into your program. All the stored commands will be carried out.

Now you will practise defining and calling a procedure.

What a procedure looks like

In the last lesson, you learned that structures in Python have a header and a body. Python procedures have the same structure. Every procedure has a header and a body.

Here is an example.



The header

The header of a procedure always has this structure:

- the word `def`, which stands for 'define the procedure'
- a name for the procedure. This example procedure is called `welcome`
- open and close brackets
- a colon.

The body

The body of a procedure has all the commands that are stored in the procedure. The procedure has three commands. They are all `print` commands. The commands print a welcome message on the screen.

Procedure name

When the programmer writes a procedure, they have to choose a procedure name.

- It must be a single word – no spaces.
- It must begin with a letter.
- It can only contain letters, numbers and the underscore character.

The name should remind you of what the procedure does. This procedure prints out a welcome message. So the programmer decided to call it 'welcome'.

Call the procedure

After you have defined the procedure, you can call the procedure. Calling the procedure means you put the name of the procedure into your program.

Activity

Start a new program and define the `welcome` procedure shown on this page. If you run the program, nothing will happen.

Below the `welcome` procedure, add code to call the procedure.

```
welcome()
```

Now when you run the program, the commands in the procedure will be carried out.

You can call the procedure more than once. Extend the program like this:

```
welcome()
welcome()
welcome()
```

Now when you run the program, the commands in the procedure will be carried out three times.

Use a procedure in the Atom List program

You practised defining and calling a procedure called `welcome`. Now you will work on the Atom List program. You will define and call a procedure called `add_atom`. It will store all the commands to add an atom to the list.

Procedure definitions

Programmers usually put procedure definitions near the start of a program. The Atom List program has a section at the top called:

```
### Procedure definitions
```

This is where you will define the `add_atom` procedure. Later, you will write lots more procedures in this section.

Define the procedure

You will define the `add_atom` procedure. It will store all the commands to add an atom to the list. Here is the procedure definition. You should recognise these commands and understand what they do.

```
def add_atom():
    name = input("enter the name of an atom to add: ")
    atoms.append(name)
    print(name, "has been added to the list")
```

Call the procedure

Now, when you want to add an atom to the list, you can call the `add_atom` procedure.

Find the section of the program that looks like this.

```
if choice == "A":
    name = input("enter the name of an atom to add: ")
    atoms.append(name)
    print(name, "has been added to the list")
```

You can delete all the code. Put the name of the `add_atom` procedure instead.

```
if choice == "A":
    add_atom()
```

Activity

Open the Atom List program that you worked on in the last lesson. Define the `add_atom` procedure shown on this page. Put the procedure definition at the start of the program.

Change the program as shown on this page. If the user types 'A', the program will call the `add_atom` procedure.

Run the program and check it works.

Advantages of using procedures

Most programmers use procedures, or similar structures, in their programs. Using procedures and other modules in your programming is called **modular programming**.

Modular programming has many advantages.

- Storing code in procedures makes your main programs shorter and easier to read.
- Code stored in a procedure can be reused, saving time and effort.
- The work of writing a program can be shared among a team. Each person in the team writes a different procedure.
- When students first begin to make and use procedures it can be hard to see these advantages. But in the long run, knowing how to use procedures makes you a better programmer.

Built-in functions

You have defined and used modules called procedures.

Python has some modules that have been written for you. They are called **built-in functions**. You don't have to define the built-in functions. You can just use them in your code.

In fact, you have been using these built-in functions all along. Here are some examples of built-in functions that are included in Python:

```
input()
int()
print()
```

Do you recognise these functions? The names of these functions are shown in purple when you type them in your programs.

Extra challenge

This is the code in your program which removes an atom name from the list.

```
if choice == "B":
    name = input("enter the name of an atom to remove: ")
    atoms.remove(name)
    print(name,"has been removed from the list")
```

Use the skills you have learned. Make a procedure called `remove_atom`. Call the procedure in your main program.

Test

These commands create a new procedure.

```
def say_hello():
  name = input("what is your name? ")
  print("hello", name)
```

1. Give the command to call this procedure in a program.
2. Describe what happens when this procedure is called.
3. Two lines of the procedure are indented. Explain why.
4. Describe the advantages of using procedures in programs (modular programming).

4.3 Linear search

In this lesson

You will learn:

- how to write a procedure to find a value in a list
- what a linear search is.

Spiral back

In previous books you have used relational operators. Using a relational operator you can make a logical test with the value True or False. In this lesson you will learn a new relational operator: the word `in`.

Search a list

The students at City Park School decided to make a program to meet this requirement:

The user enters the name of an atom. The program searches for the name in the list. If the name is in the list, the program will display a message.

The new program is called Atom Finder. Later in this lesson you will download the new program and make changes to it.

A new relational operator

When you do a search, the item you look for is called the **search term**. Python offers a ready-made way to find a search term in a list. It is a relational operator:

```
in
```

Remember that relational operators compare values. Comparing values makes a logical test.

The relational operator `in` compares a single value and a list. If the value is 'in' the list, then the test is True. If the value is not in the list, then the test is False.

Here is a simple example program that uses `in`.

```
mylist = ["A","B","C"]
letter = input("Enter a letter: ")
if letter in mylist:
  print("The letter is in the list")
```

Now you will use the `in` operator to find an atom in the list.

Start a new program

Open the Atom Finder program.

Or copy and paste the list of atoms from the last program into a new program file.

Tasha added code to the Atom Finder program. She used the `in` operator. The program will tell the user if the name they input is in the list.

```
name = input("enter a name ")
if name in atoms:
    print(name,"is in the list of atoms ")
```

Punam made a longer version. This version uses `if... else`. This extended program will show one message if the name is in the list, and a different message if the name is not in the list.

```
name = input("enter a name ")
if name in atoms:
    print(name,"is in the list of atoms ")
else:
    print(name, "is not in the list of atoms")
```

Activity

Open the Atom Finder program. Add commands so that the program gets a name from the user. It tells the user if an atom is in the list or not in the list. Use the `in` operator and the `if... else` structure.

Other ways to search

In this unit you will learn more about how searches work. You will learn how to do a search without using the `in` operator. This will help you to understand the different ways a computer can search a list for a value.

The two methods you will learn are called:

- the **linear search**
- the **binary search**.

In this lesson you will learn about the linear search.

Traverse the list

In Unit 3 you learned how to traverse a list using a `for` loop. Traversing a list means visiting every item in the list. You may print each item to show you have visited it. This table shows the actions for traversing a list and the Python code that matches each action.

Action	Python code
Find the stop value for the `for` loop. (The stop value is the same as the length of the list.)	`stop = len(atoms)`
Use a `for` loop to count through the list one item at a time.	`for i in range(stop):`
Print the list item.	`  print(atoms[i])`

If you put all the commands together, the computer will traverse the list.

```
stop = len(atoms)
for i in range(stop):
    print(atoms[i])
```

When the program is run, it prints out the name of every atom. This picture only shows part of the output.

```
Hydrogen
Helium
Lithium
Beryllium
Boron
Carbon
Nitrogen
Oxygen
Fluorine
Neon
Sodium
Magnesium
Aluminum
Silicon
Phosphorus
```

Activity

Work with the Atom Finder program.

- Delete the code that used the `in` operator.
- Add new code to traverse the list.
- Run the program. Your program should print out every value in the list.

Overview of the linear search

A human might look at a whole list with one glance. But a computer has to check each value one at a time. To find a search term in a list, the computer must look at each value one after the other. It will check each value to see if it matches the search term. That is how the linear search works.

Program the linear search

You have written code to traverse the list. You used a `for` loop to visit every item in the list. Now you can adapt that code. Instead of printing out the items, the computer will check each item. It will check if it matches the search term.

This table sets out all the actions of the linear search. The table is incomplete.

Action	Python code
Get the search term from the user. Store the input as a variable called 'name'.	`name = input("Enter a name: ")`
Find the `stop` value (length of list).	
Use a `for` loop to count through the list one item at a time.	
An `if` statement. The logical test checks if the list item matches the search term.	`if atoms[i] == name:`
If the test is True, print a message saying you have found the search term.	

Activity

Copy this table.

Complete the table to show the Python command that matches each action. Look back at the 'traverse' example to help you.

Summary

A linear search is a way of finding a value in a list.

- It uses a `for` loop to count through the list.
- It compares each value in the list with the search term.
- If there is an exact match, it displays a message saying the item is found.

Here is the complete code to carry out a linear search.

```
### the linear search
name = input("Enter a search term: ")
stop = len(atoms)
for i in range(stop):
    if atoms[i] == name:
        print(name, "found in the list")
```

Activity

Edit the Atom Finder program.

- Delete the code to traverse the list.
- Add code to do a linear search.
- Run the program to make sure it works.

Extra challenge

Here is an extended menu for the Atom Finder program.

- Add code to the program to make this menu appear on screen.
- Add code so all the menu options work.
- Put the menu into a `while` loop.

```
====================
A T O M  F I N D E R
====================


A: Add an atom to the list
B: Remove an atom from the list
C: Print the list
D: Search the list
X: Exit the program


Choose an option:
```

Test

1. Explain how the `in` operator is used to make logical tests.
2. In this lesson you used a loop to traverse a list. Explain what 'traverse' means.
3. What is a search term and how is it used in a search program?
4. The following code says that an atom is in the list. Change the code so it will print out the name of the atom, as well as saying that it is in the list.

```
print("The atom is in the list")
```

4.4 Linear search procedure

In this lesson

You will learn:

- how to change the linear search code into a linear search procedure.

Linear search procedure

In Lesson 4.2 you learned how make a procedure. In the last lesson, you learned how to make the linear search. In this lesson you will make a procedure to do the linear search. Make sure you understand the work from earlier lessons before you begin this task.

You should have the Atom Finder program open on the screen. Go to the section of the program where you will put the procedure definitions.

Header of the procedure

Programmers usually define procedures at the top of a program. Every procedure starts with a header. Look back at Lesson 4.2 to remind yourself how to define a procedure.

Think of a good name for the linear search procedure. In this example we have called it 'linsearch'. The header of the procedure looks like this:

```
def linsearch():
```

Body of the procedure

The body of the procedure stores all the commands to carry out the linear search. You already wrote these commands. Find these commands in the program. Copy and paste the commands into the procedure.

Remember to make sure that the commands in the body of the procedure are indented.

Here is the completed procedure definition.

```
def linsearch():
    name = input("Enter a search term: ")
    stop = len(atoms)
    for i in range(stop):
        if atoms[i] == name:
            print(name, "is in the list")
```

Call the procedure

To call the procedure, you must include the name of the procedure in your code.

Go to the part of your program where the linear search code was. If you have not deleted it already, delete all the code that carries out the linear search. Replace all the deleted code with the name of the procedure.

```
linsearch()
```

Your program will now carry out the linear search.

Activity

Create the linear search procedure and use it in your main program. Run the program and answer these questions.

a What is the output if you type the name of an atom that is in the list?

b What is the output if you type the name of an atom that is not in the list?

If the search term is not found

The students made this program and tested it. Amal ran the program. He searched to see if 'Silver' was in the list. This was the output.

```
Enter a search term: Silver
Silver is in the list
```

But if the search term is not in the list, you do not see a message. The program just stops. The students at City Park School decided to make changes to their program. They wanted to meet this requirement:

The user enters the name of an atom. The program searches for the name in the list. If the name is in the list, the program will display a message. ***If the name is not in the list, the program will display a different message.***

The students worked on the program. They made some mistakes at first.

Amal's mistake

The linear search procedure uses `if`. If it finds the search term, it displays a message. Amal decided to use `else` to display a different message if the program did not find the term. Here is the linear search procedure that Amal made.

```
def linsearch():
    name = input("Enter a search term: ")
    stop = len(atoms)
    for i in range(stop):
        if atoms[i] == name:
            print(name, "is in the list")
        else:
            print(name, "is not in the list")
```

Amal ran his program. This was the output. This is just part of it.

```
Enter a search term: Silver
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
Silver is not in the list
```

The program displays the message 'Silver is not in the list' every time it finds an item that does not match the search term. That is not helpful. The user will see the message even if the search term is in the list. The user will see lots of messages.

Ria's mistake

Ria decided to put the 'not found' message at the end of the loop, so it would only show once. Here is the linear search procedure that Ria made.

```
def linsearch():
  name = input("Enter a search term: ")
  stop = len(atoms)
  for i in range(stop):
    if atoms[i] == name:
      print(name, "is in the list")
  print(name, "is not in the list")
```

Ria ran her program. This was the output.

```
Enter a search term: Silver
Silver is in the list
Silver is not in the list
```

The program displays the message 'Silver is in the list'. It also displays the message 'Silver is not in the list'. That is not helpful. The user doesn't know which message is true.

A hint from Mr Shakir

The computer science teacher at City Park School is called Mr Shakir. Mr Shakir gave the students some advice. He told them about a new program command:

```
return
```

The `return` command is a Python key word. It is shown in gold. Remember that key words are used in Python to control the structure of the program.

A programmer can put the `return` command into a procedure. When the computer sees the `return` command it will stop the procedure and 'return' to the main program.

Using return

By using the `return` command, you can make a linear search procedure that works. Here is the completed procedure.

```
def linsearch():
    name = input("Enter a search term: ")
    stop = len(atoms)
    for i in range(stop):
        if atoms[i] == name:
            print(name, "is in the list")
            return
    print(name, "is not in the list")
```

Mr Shakir put the `return` command into the procedure.

- If the computer finds the search term, it will display the message 'the atom is in the list'. Then it will return to the main program. It will never show the final message.
- If the computer does not find the search term in the list, it will finish the loop and go to the final message. It will display the message 'the atom is not in the list'.

Here are two examples of the output of the program. In the first example, the search term 'Oxygen' is in the list.

```
Enter a search term: Oxygen
Oxygen is in the list
```

In the second example, the search term 'Water' is not in the list.

```
Enter a search term: Water
Water is not in the list
```

The program will always show the right message. The program now meets the program requirement that the students planned.

Activity

Amend the linear search procedure so that it displays a message if the atom is in the list, and a different message if the atom is not in the list.

Test the program with different inputs to make sure it works.

Extra challenge

It is important for programs to be readable. To make programs readable:

- use helpful names for variables and procedures
- include comments to explain the parts of the program.

Make sure the Atom Finder program is readable.

Test

Simran made a procedure called `my_procedure`.

1. Write the header of this procedure.
2. Write the line in the main program that calls this procedure.
3. The procedure includes the command `return`. What happens when the computer sees this command?
4. Mr Shakir said that Simran has not chosen a very good name for his procedure. What is the problem with the name Simran chose?

4.5 Two types of search

In this lesson

You will learn:

- about the different types of search
- how the binary search works.

Search algorithms

An **algorithm** sets out the steps to solve a problem or complete a task. An algorithm can be used to plan a program. Sometimes there is more than one algorithm that will solve the same problem.

An example is searching a list for an item. There are two ways to do this:

- linear search
- binary search.

Linear search algorithm

You have made a program that uses the linear search algorithm. It has these steps:

1. Count through the list from the first item.
2. Compare each item to the search term.
3. If there is a match, the item is in the list.

The computer has to look at every item in the list. If the list has a lot of items that can take a long time.

You used the linear search algorithm to find a search term in a list of atoms. The list of atoms is not very long. But in real life, lists can be very long. Think of the list of all the customers of a bank. Or the list of all the users of a social network site such as Facebook. In real life, lists can have millions or even billions of items. If you need to find an item in a big list like this, the linear search is too slow.

Binary search algorithm

A faster way to search a list is called the binary search. Binary means 'made of two parts' or 'divided into two'. The binary search works by splitting the list in two.

Here is how the binary search works.

1. Sort the list into order.
2. Find the midpoint of the list.
3. Compare the search term to the midpoint. Is it higher or lower?
4. Split the list in half and keep the higher or lower half.
5. Keep splitting the list in half until it has only one item.

The binary search is much faster than the linear search. But it has one drawback. You can only use the binary search on a sorted list.

Act out the binary search

You can act out the binary search in class.

Write out the names of atoms on cards:

- Carbon
- Gold
- Helium
- Hydrogen
- Mercury
- Nitrogen
- Oxygen
- Platinum

(You can add more atoms to the list if you like.)

Stand in a line. The line represents the list of atoms. Each of you should hold a card with one atom name. Make sure the names are in alphabetical order.

Your teacher will give you a search term. It might be a name in the list or not.

Find the midpoint

Count the number of students in the line (the list). Find the middle student in the list. The middle position in a list is called the **midpoint**.

The value stored in this position is the **midpoint value**. Look at the card held by the student at the midpoint. This card shows the midpoint value.

Split the list

Now you can split the list into two halves.

- One half has all the values smaller than the midpoint value.
- The other half has the midpoint and all bigger values.

Keep one half

Compare the search term with the midpoint value.

- **If the search term is earlier in the alphabet**, keep the first half of the list.
- **Else**, keep the second half of the list.

The other students can sit down.

Repeat

Repeat all the steps with the new smaller list.

- Find the midpoint.
- Compare the search term to the midpoint.
- Only keep half of the list.

Repeat again. The list will get smaller and smaller.

Stop when there is only one student left

After a few repeats, you will be left with one student. Compare their atom with the search term.

- **If it matches the search term**, you have found it.
- **Else**, the search term is not in the list.

Act out the binary search. Either do it as a whole class or in a group.

Using the searches

A computer tried to find an item in a list of 128. The item was not in the list. How long did it take the computer to find that out?

- **Using the linear search** the computer had to look at every item in the list. It took 128 operations.
- **Using the binary search** the computer had to split the list in half 7 times. That took 7 operations.

The binary search uses fewer operations. It is a much quicker way to search for an item. If the list has a million items, then the binary search is much quicker than the linear search. Linear search takes a million operations. Binary search takes twenty operations.

The big disadvantage of the binary search is that the list has to be sorted. Sorting a list can take a long time. If new items are added, the list might need to be sorted again. In cases like this we might decide to use the linear search instead.

Comparing the two searches

This table compares the two types of search. Some of the cells are empty.

	How it works	Advantages	Disadvantages
Linear search			Can be slow if the list has a lot of items
Binary search	Split the list in half over and over again until it only has one item		Can only be used if the list is sorted

Activity

Copy the table. Fill in the empty cells using information from this lesson.

You can do it by hand or use software, such as word-processing software.

Extra challenge

How many times do you have to split a list to get it to one item? That depends on the size of the list. You have seen that a list of 128 items takes 7 splits to get to one item.

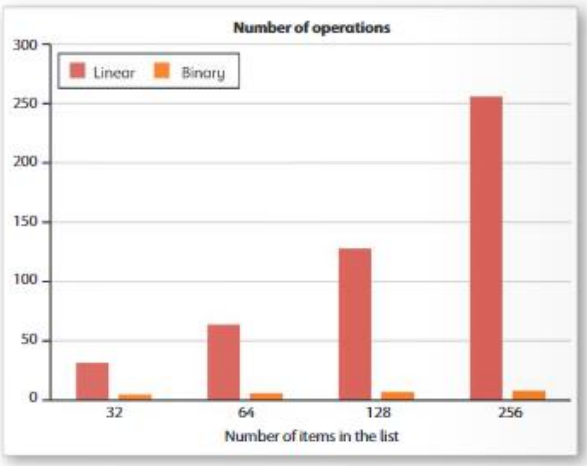
How many operations would it take to split each of these lists down to one item, splitting in half each time?

a 32 items

b 64 items

c 256 items.

Make or draw a graph to show the number of operations needed for each size of list, using each type of search.



Test

1 What are the names of the two search algorithms?

2 Which of the two search algorithms finds the search term more quickly?

3 Give one reason why you might use the slower search algorithm.

4 Describe the steps of a binary search in your own words.

Be creative

Make a poster about one of the searches. Your poster should describe how the search works and the advantages of this type of search. Use software to make the poster.

4.6 Binary search

In this lesson

You will learn:

- some of the commands used in the binary search
- how to make the binary search procedure.

Commands used in the binary search

These commands are used in the binary search:

- Sort a list.
- Find the midpoint of a list.
- Split the list in half at the midpoint.

You will practise these commands in the Python Shell.

Practise in the shell

Open the Python Shell. Enter this command to make a list of atoms. You can add different values to the list if you want.

```
atoms = ["Hydrogen", "Helium", "Carbon", "Nitrogen",
"Oxygen", "Platinum", "Gold", "Mercury"]
```

Sort the list

Before you start splitting the list, you must sort it into alphabetical order. This is the command to sort the list of atoms:

```
atoms.sort()
```

Type this command. Print the list and you will see it is in alphabetical order.

```
>>> atoms = ["Hydrogen", "Helium", "Carbon", "Nitrogen", "Oxygen",
"Platinum", "Gold", "Mercury"]
>>> atoms.sort()
>>> print(atoms)
['Carbon', 'Gold', 'Helium', 'Hydrogen', 'Mercury', 'Nitrogen', 'O
xygen', 'Platinum']
```

Find the midpoint

Each item in a list is identified by a number. This number is called the index. Index numbers are integers. The index number is shown in square brackets.

To carry out the binary search you need to find the index number of the midpoint. The midpoint is found by:

- the length of the list
- divided by two.

The result of this calculation gives you the index number at the midpoint of the list. Remember that an index number must be an integer.

Integer division

Luckily, Python lets you do **integer division**. That means normal division, but the result is an integer. The answer is rounded down to a whole number.

To do integer division you use a double division sign:

```
midpoint = len(atoms)//2
```

In this case the list has 8 items, so the midpoint is 4.

```
>>> print(atoms[0])
Carbon
>>> print(atoms[3])
Hydrogen
>>> print(atoms[7])
Platinum
```

Output the value at the midpoint

To output a single item from the list, you have to give its index number. Here is an example.

To output the value at the midpoint of the list, use this command:

```
print(atoms[midpoint])
```

The midpoint of the list of atoms is position 4. This command will output the value at position 4 in the list.

```
>>> midpoint = len(atoms)//2
>>> print(midpoint)
4
>>> print(atoms[midpoint])
Mercury
```

Split the list in half

To see a single element of a list, you put the index number inside the square brackets. For example:

```
print(atoms[2])
```

A part of a list is called a 'slice'. When you split the list in half, the two halves are called slices.

To see a slice of a list, you give the numbers at the start and end of the slice, with a colon in between. For example:

```
print(atoms[0:2])
print(atoms[2:5])
```

If you leave out a number before or after the colon, the computer will use the start or end of the list. Here are some examples in the Python Shell.

```
>>> print(atoms[2])
Helium
>>> print(atoms[0:2])
['Carbon', 'Gold']
>>> print(atoms[2:5])
['Helium', 'Hydrogen', 'Mercury']
```

This command shows all the values up to the midpoint:

```
print(atoms[:midpoint])
```

This command shows all the values from the midpoint to the end of the list:

```
print(atoms[midpoint:])
```

```
>>> print(atoms[:midpoint])
['Carbon', 'Gold', 'Helium', 'Hydrogen']
>>> print(atoms[midpoint:])
['Mercury', 'Nitrogen', 'Oxygen', 'Platinum']
```

Activity

In the Python Shell, make a list of atoms. Enter commands to:

- sort and output the list
- find and output the midpoint value
- print the list in two slices – one up to the midpoint, and one from the midpoint to the end.

BInary search procedure

You have learned some of the commands that are used in the binary search procedure. You tried them out in the Python Shell. Here are some of the commands you have learned:

- sort the list
- find the midpoint
- split the list at the midpoint.

Now you will use these commands to make the binary search procedure. Only complete this task if you have done everything else in this unit.

The procedure is called `binsearch`. That is short for binary search.

'While' loop

The `binsearch` procedure has a `while` loop in it. It loops while the length of the list is greater than one.

```
while len(atoms) > 1:
```

When the list is sliced down to just one atom, the loop stops.

Passing a parameter

The procedure slices the list until it only has one element left. That sounds a bit worrying. Doesn't that spoil the list and lose all the atoms?

Don't worry – this procedure will use a copy of the list. Only the copy gets sliced up. The original list is not affected. You can send a copy of a value to a procedure. This copy is called a **parameter**. Sending the copy is called **passing a parameter**.

In this example:

- the procedure is called `binsearch`
- the parameter is called `atoms`.

The header of the procedure looks like this:

```
def binsearch(atoms):
```

The name of the parameter is included in the brackets in the header of the procedure.

Define the procedure

Because the procedure is quite complicated, here is the whole procedure definition in full. You can use this procedure in your program.

```
def binsearch(atoms):

    atoms.sort()
    name = input("Enter a search term: ")

    while len(atoms) > 1:

        midpoint = len(atoms)//2

        if name < atoms[midpoint]:
            atoms = atoms[:midpoint]
        else:
            atoms = atoms[midpoint:]

    if atoms[0] == name:
        print(name,"found in the list")
    else:
        print(name,"not found in the list")
```

Call the procedure

The Atom Finder program calls the `linsearch` procedure.

```
if choice == "D":
    linsearch()
```

Change this code so it calls the binsearch procedure instead. Remember to include the parameter.

```
if choice == "D":
    binsearch(atoms)
```

Extra challenge

Open the Atom Finder program. Define the `binsearch` procedure as shown on this page.

Call the `binsearch` procedure.

Run the program and check that it works.

Test

Here is the `binsearch` procedure. Different parts of the procedure have been labelled with letters.

```
def binsearch(atoms):

    atoms.sort()
    name = input("Enter a search term: ")

    while len(atoms) > 1:

        midpoint = len(atoms)//2

        if name < atoms[midpoint]:
            atoms = atoms[:midpoint]
        else:
            atoms = atoms[midpoint:]

    if atoms[0] == name:
        print(name,"found in the list")
    else:
        print(name,"not found in the list")
```

A

B

C

D

Give the letter that matches each of these parts:

1. The user enters a search term.
2. The program tells the user whether the search term is in the list.
3. The program finds the midpoint of the list.
4. The program splits the list at the midpoint.

Check what you know

You have learned

- how to store blocks of code as procedures and use them in programs
- how to create a procedure to find an element in a list
- how to compare algorithms used for searching a list.

Try the test and activities. They will help you to see how much you understand.

Test

1. Name two different ways to search a list.
2. Which of the two methods is usually faster?
3. Which type of search algorithm uses a `for` loop? What commands go inside the `for` loop?
4. Choose one type of search algorithm. Explain its advantages and disadvantages.
5. Describe the reasons why a programmer might choose to use procedures in their programs.

Activities

A recruitment company is working on a program to search a list of jobs that are available.

The program is called Job Search. The program is incomplete. Your task is to finish the program.

```
jobs = ["printer","tailor","soldier","sailor","programmer",
        "teacher","doctor","nurse"]

def traverse():
    stop = len(jobs)

def linear_search():
    name = input("enter the name of a job")
    stop = len(jobs)
```

Download the Job Search program. Or copy the code shown here.

Complete these activities to finish the program.

1. Two procedures have been defined. They are called `traverse` and `linear_search`. Add commands at the bottom of the program to call these two procedures.

2 The procedure `traverse` is incomplete. Complete this procedure so it prints out every item in the jobs list.

3 The procedure `linear_search` is incomplete. Complete this procedure so it tells the user if a job is in the list or not.

Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1.
- I answered test questions 1–4.
- I completed activities 1 and 2.
- I answered all the test questions.
- I completed all the activities.

Re-read any parts of the unit you feel unsure about. Try the test and activities again – can you do more this time?

Digital citizen of the future

You have learned that writing modular programs is good for teamwork. Almost all software development is done in teams. Knowing how to work in a team is a key skill for employment. Think about the personal attributes that help with teamwork – respect for all, willingness to help and share, tolerance and open communication. People who can work in this way are at an advantage in the modern world.

Explore more

Create a set of cards to demonstrate the binary search to a friend or member of your family. The cards can show numbers, letters of the alphabet or words.

Place the cards face down in sorted order. Ask your friend or family member for a 'search term' (you might have to explain what that means).

▶ Turn over the midpoint card and compare it to the search term.

▶ You can now discard half the cards without looking at them.

Repeat these two steps until there is only one card left on the table. This will match the search term (or the search term is not among the cards at all).

5 Multimedia: Creating and sharing digital media assets

You will learn

- how to plan a media project using a brief and storyboard
- how to choose the right hardware and software for your project
- how to record video clips and edit them using video editing software.

In this unit, you will plan and deliver a media project. You will create and share digital assets, and use them in the project. A **digital asset** is digitally stored content that has some value to you and other people.

The types of digital asset in this project are:

- still images, text and graphics
- video clips.

Digital assets can be used in different ways. In this unit, you will use digital assets to create a video featuring your classmates. Your video will be unique to you. It will use assets you have created with your class to help you make a record of your time at school.

You will work in collaboration with other students. This means you will work together to choose the hardware, software and services for your project. You will share the work of creating the digital assets.

The assets could be reused by your school. For example, they could tell new students and their parents about the school.

Learning outcomes: Select and use suitable technology for a given purpose; Use technology creatively

Working together on media projects

Have you ever watched the end credits of a film, TV programme or computer game? Media projects often involve hundreds of people working together to create the final product. The people working together on a film are often called the 'crew'. The average number of crew members in popular films is over 500. Modern video games are often developed by teams of more than 1000 people.

To work together on complex projects, crews or teams need to have a good understanding of their responsibilities. They need to understand how their work contributes to the final product.

In this unit, you will learn some ways to work together with your classmates as a media production crew.

Choosing technology

One of the most important decisions a media production team makes is about technology. What software, hardware and digital services should the team use? Technology has a strong influence on projects, so you need to make choices carefully. In this unit, you will learn how to analyse choices and make technology decisions.

Unplugged

- List some of the different crew roles in films, TV and video game production. For example: a director, an editor, a sound recordist. What work do they do in the project? Who do you think they collaborate with most in the project? How do they work together and make decisions?
- Think about the different technologies that film-makers, TV crews and games development teams use. How do you think media teams make decisions about technology?

Share ideas.

Talk about...

What technology choices have you made at home? Think about devices like computers, phones and tablets, TVs and household appliances. How did you and your family decide between the different options?

Did you know?

The largest number of crew members in a Hollywood film is for Iron Man 3.

3310 people worked on the movie.

project brief
digital asset
prototype
storyboard
requirement
options analysis
metrics
aspect ratio
resolution

5.1 Plan a media project

In this lesson

You will learn:

- how to plan a project from a brief
- how to create a prototype to help you plan.

Spiral back

In Student Book 6 you presented the findings of a collaborative project. Last year, you created content to meet the needs of an audience. In this unit, you will build on these skills to collaborate in a media project. You will make technology choices based on the needs of your project.

The importance of planning

You are going to create a video using shared digital assets. That means assets created by you and your classmates. The digital assets are:

- short video interviews with classmates
- still photographs
- graphics, such as logos and charts
- text, such as titles and captions.

You will only have a short time to create these assets and put them together. So you need to plan your work carefully with your classmates. You need to choose the right hardware and software to create your content. A project brief will help you do this.

Project brief

A **project brief** is a document that tells you important facts about your project. It should answer these questions:

- What does the customer want?
- When does the customer want it?
- Who is the product for?
- Why does the customer want it?

In this project, the customer is your school. Your class needs to deliver the project. Here is your project brief.

About the project

Our school is creating a library of digital assets that will present a positive message about life at school. Every student will be able to choose some of the digital assets to create a video.

Project goals

The videos will show students' talent and IT skills. They will encourage other children and parents to choose the school, by showing:

- videos with a positive message about the school and learning
- images of the school
- information about the school.

About the audience

The audience are:

- current students, who will have a record of their time at school
- parents and carers who would like to see how their children are doing at school
- other children, who might join the school.

Scope and production guidelines

Each student must take part in a video interview and produce at least two still images. The images can be of the school buildings or classrooms, or group pictures of the class. Still images may also include charts and drawings. They must have a white background.

The team must follow these guidelines for the video interview.

Content	Each student should answer these three questions: ▶ What has been your favourite activity in school this year? ▶ Who inspires you? Why? ▶ What are you looking forward to learning about next year?
Style	Each interview should be filmed in one of the following styles: ▶ piece-to-camera ▶ off-camera conversation. The interview may take place indoors or outdoors. It must be in or around the school buildings.
Technical	The interviews must be recorded and saved in high-definition video, in colour. They must be recorded in a widescreen format (16:9 aspect ratio). The file names must include the name of the student.

All assets (videos and still images) will be stored on a file-sharing service. Students must save, load and share files using this service.

Each student can use any of the digital assets to produce their final video. Videos should:

- be no more than five minutes long
- have at least three interview segments with classmates
- feature still images.

Timeline

The video must be ready for viewing and sharing by the end of this unit.

Planning your product

The project brief can help you plan your product. In this project, the product is the video. In IT and media projects, people often use **prototypes** to plan a product in more detail after they have read the project brief. A prototype is a model of the product that you make during the design process. It helps you to imagine what the finished product will look like.

There are different kinds of prototype.

- A low-fidelity prototype is usually the earliest kind of prototype. It is often a hand-drawn sketch of a product or service, like a web page.
- A high-fidelity prototype is often quite close to the finished product or service. If it was a web page, it might have some working links and features.

In media projects, designers often use a low-fidelity prototype called a **storyboard** to start planning. A storyboard is usually a set of simple drawings and notes that shows how the parts of a video or pages of a website should be ordered. It also shows what should be shown in each part or on each page. Drawing a storyboard helps you think about every part of your design.

Here is an example of the kind of storyboard you might put together for this project.

PROJECT NAME _____

Title screen. Outside school. Show for 5 seconds.

Interview with Majid.

Interview with Mohammed.

Activity

Download and open the storyboard template.

Review the project brief. Think about what you want to include in your video.

Complete your storyboard. You should add at least three interview parts to your storyboard – each one in its own frame. Write in the name of the students being interviewed. You will need this information later.

Add a storyboard frame for each still image you want to include.

Add notes to the storyboard frames to help you plan your work. For example, notes about the style of the images, or any ideas for captions or other text you want to show.

Extra challenge

Storyboards are used by many creative people in the media industry.

Search online for services that can help you storyboard your projects.
For example, try www.storyboardthat.com/storyboard-creator.

Test

1. What does a project brief tell you?
2. What is a prototype?
3. Name two things that a storyboard can show you about a media product.
4. What is the difference between a low-fidelity prototype and a high-fidelity prototype?

5.2 Understand requirements

In this lesson

You will learn:

- how to make the right technology choices by understanding requirements
- how to prioritise requirements to make choices easier.

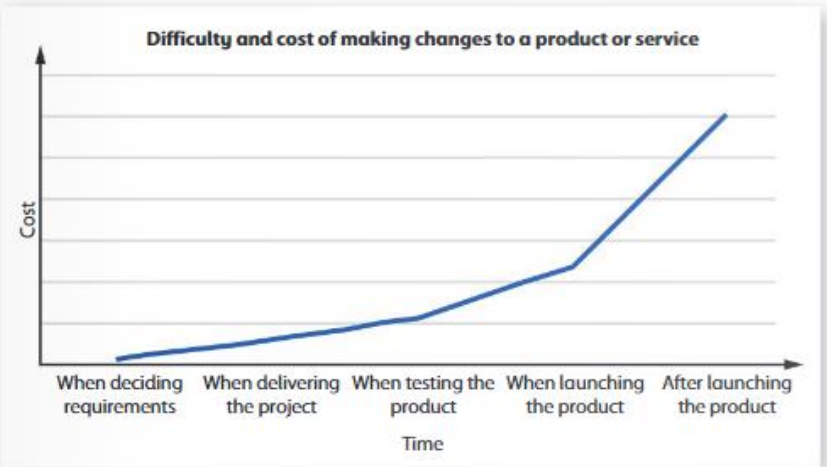
Why is it important to understand requirements?

Requirements are the things that technology needs to be able to do, in order to help you achieve your project goals.

In any project, it is important to understand as much as possible about your requirements before you start work. A project is strongly influenced by the early decisions you make, for example, about what software or hardware to use.

The graph shows how difficult and expensive it can be if you do not plan and think very carefully about your needs early on in a project. It shows that as the project goes on, making changes to the product can become more and more expensive.

If you have not thought carefully about your project needs, you might find there are requirements that your chosen technologies can't meet. It may be difficult and expensive to introduce new technologies. You might even have to redo some of the work.



How can you make the best choices?

There are lots of different types of hardware and software to choose from. So it's important to have an organised way of making decisions.

Here is a set of steps to help you.

1 Understand the things you are trying to achieve – your **requirements**.

2 Decide on what is most important. This is called **prioritising**.

3 Research your choices and how they match your requirements and priorities. This is called an **options analysis**.

4 Make your decisions based on your analysis.

Creating a list of requirements

There are many ways of writing down the requirements for a technology project.

These are the most important features of requirements:

- Use simple statements that are easy to understand, even by people who are not technology experts.
- Include the role of the person or team.
- Say what the person or team needs to achieve.

You can start collecting requirements by listing the roles that you and your team need to perform in the project. When you have identified the roles, write down the requirements that each of these roles has. You can write your requirements in a table.

In this project, the team roles are video editor, artwork designer and camera operator. You will create your list of requirements in the activity at the end of the lesson.

Functional requirements

This table shows some roles and requirements for a video project like the one in this unit. The table also shows the priority of each requirement. You will learn about prioritising requirements later in this lesson.

Role	Requirement	Priority
Camera operator	*I want to be able to record high-definition video.*	*Must have*
Camera operator	*I want to be able to check my recordings without downloading them from the camera.*	*Should have*
Video editor	*I want to be able to import video clips from a file-sharing service.*	*Should have*
Video editor	*I want to be able to edit clips together to make a film.*	*Must have*
Video editor	*I want to be able to add music to my video.*	*Should have*
Artwork designer	*I want to be able to create charts.*	*Should have*
Artwork designer	*I want to be able to combine text and images.*	*Must have*

The requirements in this table are called functional requirements because they describe what technology must do to help the project.

Non-functional requirements

Most projects also have non-functional requirements. Non-functional requirements describe how a piece of software and hardware must work. You can write non-functional requirements as a simple statement.

Here are some examples.

- The software must process videos quickly, so that the editor can work with them.

  You could set a performance target. For example: The app must start up in less than 10 seconds.

- A file-sharing service must be password-protected so that the files are not lost or stolen.
- Hardware or software must cost less than $50.
- Software or hardware must work with other products and services being used in the project.

Non-functional requirements often have values that can be measured, for example, a time limit or an amount of money. These are known as **metrics**. Metrics can be helpful when comparing technology options.

Prioritising requirements

Projects often have many requirements. Often you can't meet all the requirements. This can make it difficult to choose a technology or service to help you deliver your project. Prioritising requirements can make it easier to decide.

When you have put the requirements in order of priority, you can focus on meeting the most important ones first.

Prioritise requirements by using three categories.

1. **M**ust haves: things your technology or service must be able to do. If you can't do these things, your project will fail.
2. **S**hould haves: things you should have but don't absolutely need, usually because there are other ways of meeting the requirement.
3. **C**ould haves: things that would be good to have, but you can do without.

Activity

Download the requirements template.

Review the project brief from the last lesson.

Use the template to choose six functional requirements for the software, three needed by the video editor and three needed by the artwork designer.

Remember to think about all aspects of your project. Review the project brief again to make sure.

Next, choose three functional requirements for the hardware, needed by the camera operator.

Now prioritise your requirements by giving each one a '**M**ust have' or '**S**hould have' priority. Write an M or S next to each of the requirements.

Work on your own or as a team. Keep your work safe. You will need it in the next lesson.

Extra challenge

If you have time, write down some non-functional requirements for the software and hardware in your project.

Think about:

- the cost and availability of software and hardware
- how easy it should be to use
- the compatibility of hardware and software.

Ask yourself:

- Can I find the device or software app in school or at home?
- How much time do I have to learn how to use new equipment and software?
- What apps and equipment need to work together in this project?

Test

1. What does the word 'requirement' mean in a technology project?
2. Why is it important to prioritise requirements in a project?
3. Put these priorities in the right order: should have, could have, must have. Write a short paragraph describing the difference between these priorities.
4. Describe the difference between a functional and non-functional requirement.

5.3 Make technology choices

In this lesson

You will learn:

- how to research technology options for your project
- how to analyse options and make a final decision.

In the last lesson you learned that making the right choices about hardware and software is an important part of successful IT and media projects.

You have worked out your requirements and prioritised them. In this lesson, you will use the list of requirements to make choices about the hardware and software to use in your project.

You will research your choices and assess how they match your requirements and priorities. This is called an options analysis. You will make decisions based on your analysis.

Research

Now you have your prioritised list of requirements, you can begin to explore what software and hardware will meet them.

You can research options using:

- your own knowledge of software and hardware
- recommendations from teachers and friends
- websites and other online resources.

You can also try out software and hardware as part of your research. Your computer may already have software installed that can meet some of your requirements. You may have suitable hardware in your classroom or at home.

Software companies often allow customers to try applications (apps) before they buy them, using a 'demo' or trial version. Hardware suppliers often allow customers to test a product first. Explore all the options available to you.

Analyse

To analyse your options, you can use a table called a requirements matrix. It will help you check whether software or hardware meets your needs. The matrix is a table that allows you to compare each option against all of the requirements.

You can quickly see what the best option is by checking which one meets all of your 'must haves', and most of your 'should haves' and 'could haves'.

Here is an example of a requirements matrix for video editing software. You can see that three options were analysed against the functional and non-functional requirements. A tick (✓) shows that an option meets the requirement. A cross (✗) shows where an option does not meet a requirement.

Priority	Requirement	Type	Solution option		
			Instagram app	Windows Photos app	Adobe Premiere
Must have	Can edit clips	Functional	✓	✓	✓
	Can add captions	Functional	✗	✓	✓
	Can add still images/graphics	Functional	✓	✓	✓
	Costs less than $50.00	Non-functional	✓	✓	✗
Should have	Can add music/sound	Functional	✗	✓	✓
	Can create up to 5 minutes of video	Functional	✗	✓	✓
	Can export video for sharing in different formats	Functional	✗	✓	✓
	Easy to use; quick to learn	Non-functional	✓	✓	✗
Could have	Built-in transition effects	Functional	✗	✗	✓

This app is the best option. It meets all of the 'must have' requirements.

Making difficult decisions

Sometimes it is difficult to make a technology choice. Your analysis of the functional requirements might leave you with two or more options that are suitable. So how do you decide?

In this situation, look again at your non-functional requirements. Check that you have included all of them in your options analysis. Have you missed any out?

Here are some non-functional requirements to consider.

- **Usability:** How easy are the different options to use? Do any of them need more training and practice to learn?
- **Cost:** Think carefully about the prices of the different options. It can be difficult to compare software costs when they are based on purchase (buying a copy of the software) and subscription (paying a monthly fee to use the software, usually through a cloud service).
- **Support:** Try to find out how much support and guidance there is for the different options. Do they all have manuals and how-to guides? Check on the web to see if users have posted helpful guides and videos.
- **Future use:** If you're choosing software or hardware to use for a long time, try to assess how well it will be supported in the future. Is it an established technology? Is there something more modern that will replace it soon? Future use is important to consider when you choose between a purchase and a subscription service. Subscription services usually give you regular updates to the latest version.

Activity

Download the requirements matrix template.

Work in groups. Research the options listed in the table below for your project requirements.

Try out software and hardware that you already have access to. You can also test online services and other products. And watch reviews and read how-to guides online.

Use the requirements matrix template to analyse your options for each requirement. This should help you decide on the options to use in your project.

Make sure all your chosen options are available to you for the next lesson. You will be creating the content for your project then.

Technology	Suggested options to investigate and analyse
Hardware options for filming interviews	A video camera Smartphone or tablet Desktop or laptop computer with built-in camera
Software options for editing video	Microsoft Photos app (for Windows) iMovie (for Apple iOS) Filmmaker Pro (Android and Apple iOS) Video editor available on your own device (smartphone or tablet) Other options suggested by your teacher
Software options for creating and editing graphics and images	Microsoft Paint Microsoft PowerPoint GIMP Other options suggested by your teacher
Service options for storing and sharing files	Microsoft OneDrive (for Windows) Apple iCloud (for iOS) Dropbox Google Drive Shared drive at school (if available)

Extra challenge

In business, a project team needs to get agreement on important decisions from a project manager or management board. Use your matrix to help you write a short options analysis report for your project board. Your report should explain:

- what options you considered
- what your final decision is
- how and why you made that decision.

Test

1. Name two ways you can find out about technology options for a project.
2. What types of requirements should you include in your options analysis?
3. Write down two non-functional requirements you could consider when doing an options analysis.
4. Explain one of the advantages of choosing subscription software.

5.4 Create and share content

In this lesson

You will learn:

- why it is important to agree an approach to creating media assets
- how to make video recordings
- how to save media files for sharing in a team project.

So far you have planned your project and decided on the technologies you will use to deliver it. In this lesson you will create the content for your project.

Agree an approach

The project brief asked for a video of a short interview with each member of the class.

To save time, you will share the task of filming interviews with classmates. This is how professional media producers and film-makers work. They often have separate teams who make different parts of the film.

When you work like this, it is important to agree:

- what should be filmed – the content
- how the content should look – the style
- how the content should be recorded – the technical details.

Content

To make sure that teams create content that meets the brief, they decide on a common approach.

For example, they might agree to use:

- the same questions for every interview
- three questions chosen at random from a list of questions
- a certain style for graphics and images.

Style

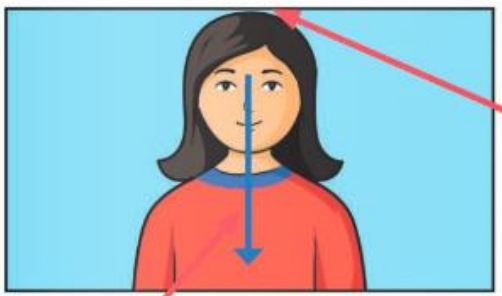
The project brief asks for a consistent look for the different interviews. To make sure you meet this requirement, follow the instructions about:

- location and background. Will you film indoors or outdoors? Will the interviewee sit or stand? What should be in the background?
- framing. How will you place the interviewee in the frame?

A 'talking heads' video shoot

You are going to look at two different ways of framing an interview shot. You may have seen these styles in videos and TV programmes.

1 This style is called 'the piece to camera'. The person being interviewed is in the centre of the frame. They look straight at the camera. This style is often used when the person is giving information to the audience. It is often used by TV news reporters and video bloggers.

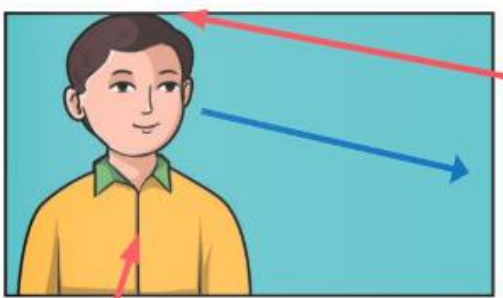


The person is framed close to the top of the frame.

The person is in the centre of the frame.

The person is looking straight at the camera.

2 This style is called 'the off-camera conversation'. The person being interviewed looks to the side of the camera, at the interviewer. The interviewer is not in the shot. They stand to the side of the camera. This style makes the audience feel like they are part of the conversation.



The person is framed close to the top of the frame.

The person is looking across the frame, past the camera.

The person is placed to one side of the frame.

Technical standards

Teams need to agree the technical standards they will use, so that they can work with the assets in the next stages of production. Here are some important details that the project brief gives instructions about: screen format, file type/quality and file-sharing.

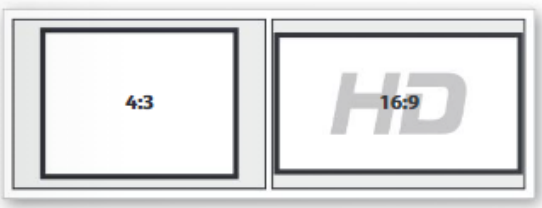
Screen format

The **aspect ratio** describes the relationship between the width and the height of the screen. The screen format, or aspect ratio, you shoot your video in should be the same for all the interviews.

Most video cameras and editing software can work with different formats. But the result will not look consistent if there are changes in the aspect ratio between different shots in your video.

The most common aspect ratios are:

- 4:3 – this is the format used in televisions before plasma and LCD-/LED-type screens were common. 4:3 is a common format in still photos.
- 16:9 – this is the most common high-definition 'widescreen' format. Most modern television screens are in 16:9 format. Screens that have a 16:9 format can also display the other aspect ratios.



Don't forget to agree a screen orientation. The usual way to film video shots is in landscape format. But many smartphone users hold their phone in portrait format when filming.

File type/quality

Most video cameras and smartphones can record video at different **resolutions**. The resolution is often described with letters and numbers. The higher the numbers, the sharper the picture will be, but the more storage space the video files need.

- 4K video (4096 × 2160) is very high resolution. Many computer displays and most TVs cannot show 4K.
- Full HD (1920 × 1080) is the modern 'high resolution' video standard. It will display perfectly on 16:9 format screens.
- Other ratios, like HD (1280 × 720), VGA, PAL and MMS, are lower resolutions. They are designed for older computer monitors and TVs, or for low-quality image sharing via messaging services.

File-sharing

The project brief requires the project team to use a file-sharing service to save the digital assets. A file-sharing service is an online service. It can save files so you can access them from anywhere with an internet connection, using any device. Sometimes, file-sharing services are called 'cloud storage'.

File-sharing services have a number of advantages over technologies like USB drives.

- They are more secure. You can lose a USB drive. Cloud storage means there's no device to lose.
- They are cheap. Many services offer free storage.
- The content can be shared. You can share files and folders with classmates by giving them access to your file storage area.

Teams need to agree how to use a file-sharing service so that everyone can easily find and access the shared files. The project brief gives instructions about how each file should be named.

Activity

Divide up the work of filming the student interviews and creating the still images. The digital assets should be ready to use in the next lesson.

Film the interviews and make the still images. Use the project brief guidelines to help you agree the content, style and technical details.

Remember to regularly save the digital assets you create in the file-sharing area.

Extra challenge

If you have time, track the progress of the project. Review the digital assets as they are saved in the file-sharing area.

If any of your classmates need support, offer them help.

Test

1. Name two things that a team must agree on before starting a video project.
2. Add the missing word in this sentence: The higher the resolution, the larger the _____ size.
3. What does an aspect ratio describe?
4. Why is it important to agree naming conventions before sharing files?

5.5 Make a rough cut of your video

In this lesson

You will learn:

- how to use a video editing app to arrange video clips
- how to add still images to your video.

You have worked with your classmates to create the video clips for your project. Now is your chance to put everything together and produce new and original digital content of your own.

You will start by using a video editing app to put together a 'rough cut' of your video.

A rough cut is an early version of a film that the director and editor create. It helps them to decide which clips to use and what order to put them in. It helps them to see what their film will look like.

Usually, a rough cut has some mistakes – it is 'rough'. There are no special effects, titles or music. You can edit and improve your chosen clips at the next stage. This is called the 'final cut'.

How to use a video editing app

There are two common types of video editing apps.

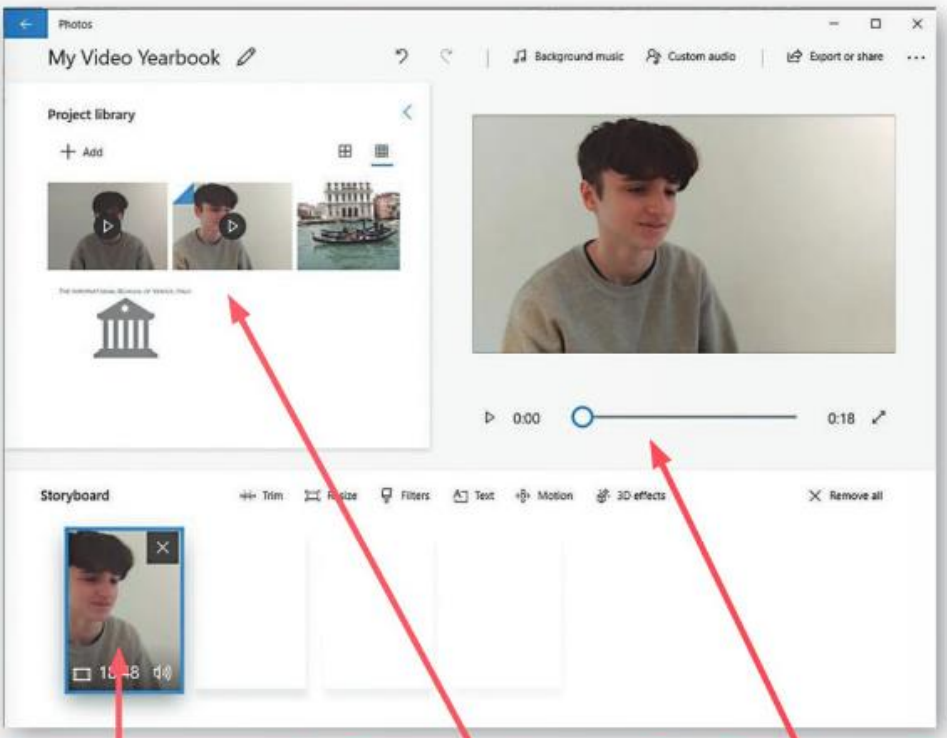
1. **Storyboard editors** let you put together your clips in a sequence. You can move the clips around in the sequence until you have a rough cut. You can then edit each clip separately to make your final cut.
2. **Timeline editors** show a timeline. They work just like audio editors. You can place clips along the timeline. You can edit the clips directly on the timeline, for example by trimming them and splitting them.

The example in this lesson uses Microsoft Photos as a video editing app. It has a storyboard interface. In Microsoft Photos, you add videos and still images to an area called the project library. You can select files that are saved on your computer. Some apps will also let you select files that are saved in shared storage, such as file-sharing services.

Add clips

When you have added your clips to the project library, you can add them to your storyboard. In Microsoft Photos, you do this by dragging and dropping the file **thumbnails** anywhere on the storyboard. You can also change the order of the clips on the storyboard by dragging and dropping them.

You can select a clip on the storyboard by clicking on its thumbnail. You can play it using the controls under the playback window. If you have put another clip after it on the storyboard, the playback window will show the next clip until all clips on the storyboard have been played.



The storyboard shows the order that the assets will be played in.

The project library shows all the assets you have imported for your project.

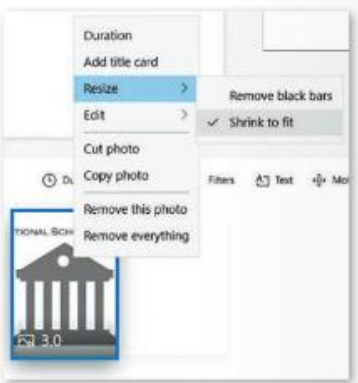
The playback window lets you play the video.

Add still images

You can add a still image, such as a photograph or a drawing, in the same way as a clip.

Still images sometimes have a different aspect ratio to video images. Your video editing app may shrink the still image to fit it into the video aspect ratio. Or it may add black bars to the top and bottom or sides to fill the screen.

You can change the way your app displays still images. In Microsoft Photos, right-click on a still image in the storyboard and change the setting.



Assemble your rough cut

A storyboard editor lets you easily change the order of your clips and images. This means you can try different ways of putting your video together. You can quickly see which way works best.

When you start to assemble your rough cut, review the storyboard you created in Lesson 5.1. Use it as a guide, but you can make changes to your film at this stage.

Try these ideas.

- Change the order of the interviews to see if they work better.
- Put still images in different places, for example, between the interviews, to 'break up' the film and add interest.
- Experiment! For example, start the film with an interview section before the title card. See if it helps for the audience to go straight into the action.

Try different ways of ordering your clips and images. Keep looking back at the project brief in Lesson 5.1. Make sure that your film still meets the requirements of the project brief.

Activity

Assemble a rough cut of your personal video.

Choose the clips and still images you need from the file-sharing area. Choose three interview clips: your own and two others.

Add the assets to the project library in your video editing app.

Put the assets into your preferred order, using the storyboard or timeline in your app.

Play the film and review your rough cut. Make any changes you want to the order of the clips.

Save your work.

Extra challenge

If you have time, explore editing some of the still image assets using the features of your chosen app.

Think about using cropping, filters and frames to improve the look of the images. If you make a change to an asset in the file-sharing area, save it with a new name. For example, 'School Building Image1_edited'.

Test

1 What does 'rough cut' mean?

2 What are playback controls?

3 Describe the difference between a storyboard editor and timeline editor in a video editing app.

4 Explain how you can use the project brief to help you when you are assembling a rough cut of a video.

Digital citizen of the future

Video content is becoming more popular on the web and on social media. Audiences often prefer watching videos rather than looking at text and images. For example, on the social media platform Twitter, a tweet (short message) containing a video is much more likely to be opened than one containing a picture. Most companies that use video content for advertising and marketing say that videos help to attract new customers.

Your video skills will help you in the world of work. Video content is likely to be become even more popular in the future.

Explore more

Ask your family and friends about the videos they watch on their devices.

Why do they watch videos? To entertain themselves? To learn new things? Find out how important video content is to the people you know.

5.6 Make and share the final cut

In this lesson

You will learn:

- how to edit clips and still images in your video
- how to add titles and captions
- how to export and share your final video.

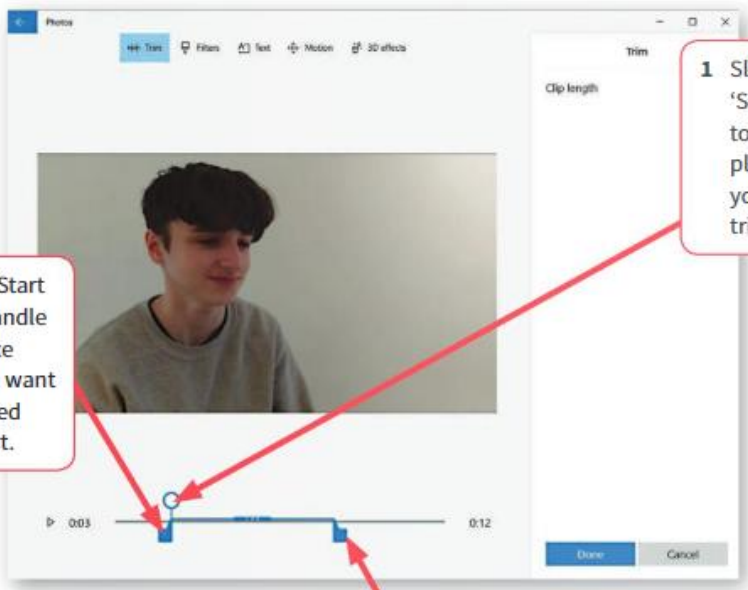
In the last lesson you created a rough cut of your personal video by arranging the assets in order. In this lesson you will edit the video clips and still images to create your final cut.

Trim video clips

You can trim the video clips. To trim a clip is to take out the unwanted parts.

You might want to remove parts of a clip from the start or end. In Microsoft Photos, you can do this by clicking on a clip in your storyboard and choosing Trim.

You can use the slider controls to trim parts of the clip.



1 Slowly drag the 'Seek' control to find the place where you want to trim the clip.

2 Drag the ' Start of Trim' handle to the place where you want the trimmed clip to start.

3 Drag the 'End of Trim' handle to where you want the clip to end. Click 'Done' to trim the clip.

If you want to take out the middle part of a clip, follow these steps.

1 Add two copies of the clip to your storyboard, one after the other.

2 Select the first clip. Slide the 'End of Trim' handle to the start of the section you want to cut out.

3 Select the second clip. Slide the 'Start of Trim' handle to the end of the section you want to cut out.

4 Play back the two clips. The section should now be removed.

In other video editor apps, you might be able to use a 'Split' tool. This will split your clip at the start and end of a section you want to cut. You can then select and delete the part that you don't want.

Change the times for still images

You can change the length of time a still image shows in your video.

In Microsoft Photos, click on a still image in your storyboard and choose the 'Duration' option. Duration means length of time. You can choose from one of the options, or enter your own duration in seconds.

Experiment with different duration settings to see what works best. It might help to show still images with text for five seconds or longer, so that your audience has time to read the text.

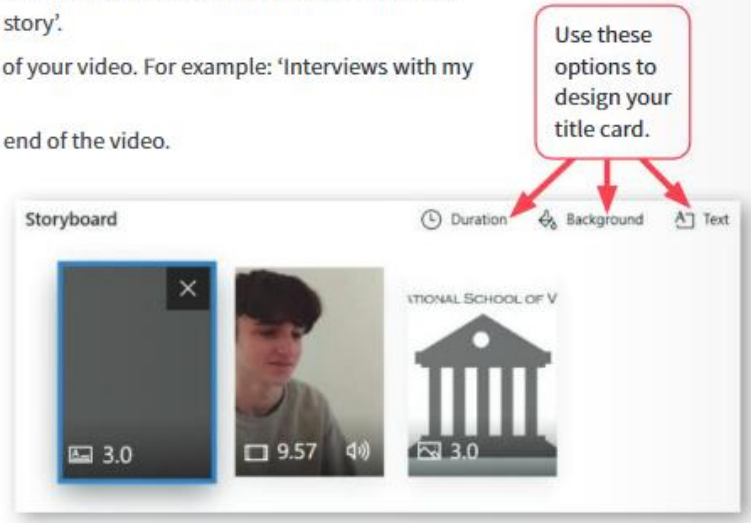
Change the setting then use the playback controls to review your video. Imagine you are watching the video for the first time. Is there enough time to read and understand everything?

Add titles

You can add a title before any clip or still image.

- Use a title at the start of your video to make it clear what the video is about. For example: 'My Year 8 video story'.
- Use subtitles for each section of your video. For example: 'Interviews with my classmates'.
- Show a list of credits near the end of the video.

If you are using Microsoft Photos, add a title by right-clicking on a clip or still image and selecting 'Add title card'. The app will add a title card to your storyboard. Select the title card and add your text. You can choose a background colour and set the length of time that the title card will be shown.

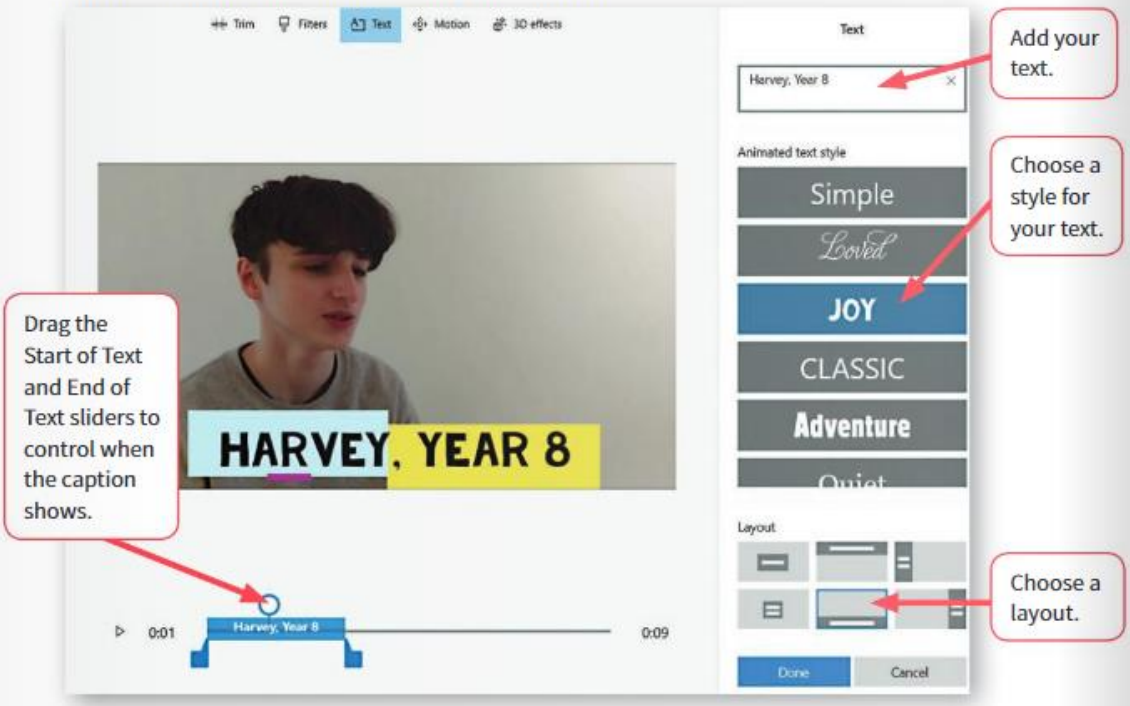


Add captions

To add a caption, select the clip or still image and choose 'Text' from the menu. A caption is text that will be shown while the clip or still image is being played. Captions are useful for things like:

- naming the person being interviewed
- explaining what a still image is about.

Use the slider to control when your caption will be displayed. Use the Playback controls to experiment with the best settings for your video.



Export and share your video

When you have finished editing your video, you can export it. Exporting means saving the video as a file that other people can watch on their devices.

Your video editing software may have many options for file formats and resolutions. Decide on the best format for how you want to use your video.

If you want to show it on large screens, you should save it at the highest resolution possible. However, this will make your file very large and difficult to share. If you want to share your video easily with friends and family, choose a lower resolution.

In Microsoft Photos there is a simple choice of small, medium and large. Choose the one that best fits your needs and save your exported file.

Activity

Open the rough cut you saved at the end of the last lesson.

Trim the video clips to show only the content you want to include.

Adjust the length of any still images, if you need to.

Add a title at the start of the video.

Export your completed video.

Save a copy of your final project to the file-sharing area. Save your work in a different folder to the one you used for the digital assets.

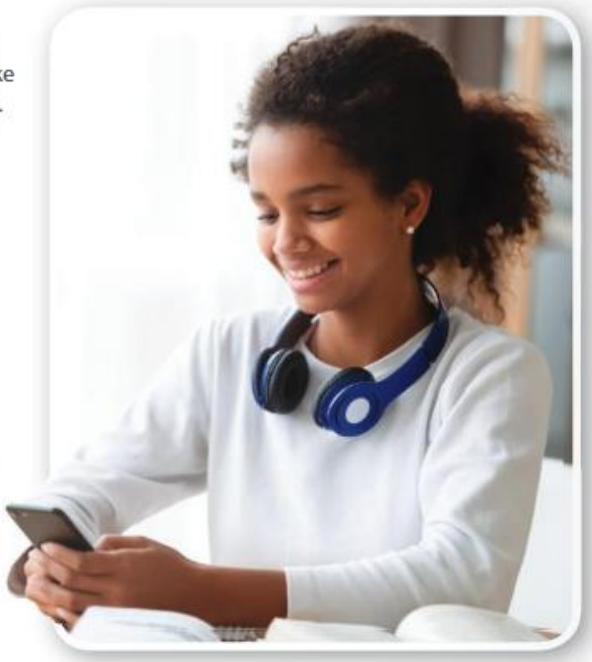
Extra challenge

If you have time, add a caption to one or more of your clips or still images.

Be creative

Use your video editing app to explore filters, transitions and motion effects on clips and still images. If you use these carefully, they can make your video look more exciting and professional. But take care not to add too much. Remember to stick to the brief!

A media project is a creative activity. Watch your classmates' videos. Discuss how you and your classmates met the project brief in Lesson 5.1.



Test

1. What does trimming a video clip mean?
2. Describe the difference between a title and a caption.
3. What do you need to think about when choosing a resolution to export your video?
4. Describe the difference between saving a video project file and exporting the video.

Check what you know

You have learned

- how to plan a media project using a brief and storyboard
- how to choose the right hardware and software for your project
- how to record video clips and edit them using video editing software.

Try the test and activities. They will help you to see how much you understand.

Test

This test is about the project work you did in this unit.

1 Name an item of hardware you used to create digital content.
2 How did you use this hardware to create digital content?
3 Describe two different types of software you used in this project.
4 Describe how you used each type of software to meet the project requirements.
5 Explain a choice you made when you were developing the project. For example, a change you made to the digital content. How did this choice improve the final product?

Activities

Download and complete the worksheet.

Read the project brief and requirements section.

1 The shop manager needs image-editing software. He has completed the software options analysis table. Read the table and decide which software you should buy. Why have you made this choice?
2 The hardware options analysis table for an underwater camera is incomplete. Complete the 'Hardware requirement' column with the missing information for the 'must have' features. Read the descriptions of the three cameras and complete the rest of the table to show whether they meet each requirement. Which camera do you recommend and why?
3 Write an email to the tour guide outlining how they can use the underwater camera and image-editing software to produce a digital asset that meets the project requirements. You do not have to give detailed instructions.

Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1.
- I answered test questions 1–4.
- I completed activities 1 and 2.
- I answered all the test questions.
- I completed all the activities.

Re-read any parts of the unit that you feel unsure about. Try the test and activities again – can you do more this time?

6 Numbers and data: Mobile Medical Services

You will learn

- how to analyse data stored in a data table
- how to use computer data to help with decision making.

In this unit you will use a data table to store information. The work is based on a case study. You will be the manager of a supply depot. A depot is a place where supplies are stored. Your depot will provide vital medical supplies to mobile health clinics. It is your job to ensure that the depot never runs out of supplies. You will use computer data to help with this task. You will highlight emergency shortages. You will make estimates for the year ahead so the service can plan what supplies it will need.

Did you know?

The World Health Organization (WHO) was established by the United Nations in 1948. Its job is to promote good health around the world. The WHO uses technology to collect and analyse data about health needs and how to stay healthy. World Health Day is 7 April every year. The WHO uses this day to communicate key facts about good health.

Learning outcomes: Use technology to analyse data

Unplugged

In this unit you will record the stock in a supply depot. The depot provides supplies to 12 mobile health clinics. Here is an extract from the data table. Rows 8–12 have been left out.

MedCode	Category	Type	Packs in stock	Packs per clinic	Packs needed
MED001	Bandages	plain	307	20	
MED002	Bandages	elastic	133	6	
MED003	Bandages	triangular	200	12	
MED004	Bandages	adhesive	21	1	
MED005	Cotton wool	roll	12	2	
MED006	Tape	adhesive roll	10	3	
MED007	Tape	hypo-allergenic	65	3	
MED013	Scissors	straight	50	3	
MED014	Scissors	curved	15	1	

The table shows the number of packs needed by each clinic. There are 12 clinics.

1. Copy this table onto paper.
2. Work out the total number of packs needed by multiplying the packs per clinic by the number of clinics (12).
3. Compare the number of packs needed with the number in stock. Is there a shortage of any item? Make a note in the table if you are short of any of the items.

You may find that this task takes a while! It is OK if you run out of time before you finish it. Don't worry – in this unit you will use a spreadsheet to make the work much easier and quicker.

Talk about...

What charities and voluntary organisations have you heard of? If you had the chance to work for a charity, which one would you choose? What are your reasons for making this choice? What benefits might you get from doing voluntary work for a charity?

AutoSum
cell reference **conditional format**
fields **highlighting** **IF formula**
key field **recalculate** **records**
reorder level **shortfall** **summary data**
surplus **worksheet**

6.1 What medical supplies do you have?

In this lesson

You will learn:

- how to organise data in a structured format to make it more useful
- how to use calculations to make new information.

Spiral back

Last year, you created a data table to store information about products sold by a business. In this lesson, you will apply these skills to a new case study. You will organise information about medical supplies, to help the doctors of a mobile hospital.

Your task

From: Director of Mobile Medical Services

I am appointing you manager of Supply Depot Four. Some basic data has been prepared for you. Get organised as quickly as you can.

Some parts of the world are affected by natural disasters and extreme events such as floods or forest fires. Travel can be difficult in these areas. It can be hard to get sick or injured people to hospital. A mobile medical service can travel to these areas and help people. It can save lives and help people to recover.

Many people work in the medical service. For example: doctors, nurses, and pilots and drivers who transport medical staff to danger zones. But there are other vital jobs. For example, it is important to make sure that doctors have all the supplies they need. Without this in place, the service would fail.

In this unit you will set up a supply data table for a mobile medical service. It will keep track of supplies and make sure they never run out. A good computer system can save lives.

What supplies are needed?

The World Health Organization makes recommendations about the stock needed by emergency clinics.

Examples include:

- bandages
- cotton wool
- tape
- scissors
- bowls
- surgical gloves
- stretcher
- trolley
- kerosene lamp
- stethoscope
- steam steriliser.

How spreadsheet features can help

In this unit you will use a spreadsheet to organise data. You will make a data table. You will use software features to answer questions and help make decisions. Each lesson will introduce new features.

In this lesson you will put the data into a table and use a formula to calculate information.

Organise the data

A spreadsheet has been made for you. It contains information about medical supplies. The spreadsheet is called 'Mobile Medical Services'.

Open the spreadsheet and look at the contents. The spreadsheet has many rows.

The top of the file looks like this:

	A	B	C	D	E	F
1	**Mobile Medical Services - Supply Depot Four**					
2	Number of clinics supported:		12			
3						
4	MedCode	Category	Type	Packs in stock	Packs per clinic	Packs needed
5	MED001	Bandages	plain	307	20	
6	MED002	Bandages	elastic	133	6	
7	MED003	Bandages	triangular	200	12	
8	MED004	Bandages	adhesive	21	1	
9	MED005	Cotton wool	roll	12	2	
10	MED006	Tape	adhesive roll	10	3	
11	MED007	Tape	hypo-allergenic	65	3	

This spreadsheet stores the stock information for Supply Depot Four. The depot has a target to provide supplies for 12 emergency clinics.

Advantages

There are many advantages to organising data as a table. Putting data in a table makes it easier to:

- sort and search the data
- use formatting to pick out important facts
- use formulas to calculate new information.

You will do all of these activities in this unit.

Rows and columns

It is your job to organise the data in the spreadsheet. The first step is to make the data into a table of rows and columns. In Student Book 7 (and in earlier books in this series) you learned about creating and using data tables.

- The columns of a table are called **fields**. Each field stores one piece of data.
- The rows of a table are called **records**. Each record stores all the data about one thing.

One of the fields is a **key field**. The key field stores data that is unique for each item.

What belongs in the table?

Look at the spreadsheet and see what data belongs in the table.

- Rows 1 and 2 have general facts about the depot. Row 3 is empty. These rows do not belong in the table.
- Rows 4 to 28 belong in the table.
- Columns A to E have data in them. They belong in the table.
- Column F has a heading but no data. You should include this column in the table. You will add data later.

With these facts in mind, select all the cells that belong in the table. You will select all the cells from A4 to F28. Select cells by dragging the mouse across the cells.

Format as a table

When the cells are selected, click on 'Format as Table' at the top of the window. This option is in the 'Styles' section of the 'Home' tab. Choose a colour for the table.

The finished table will look something like this:

	A	B	C	D	E	F
1	**Mobile Medical Services - Supply Depot Four**					
2	Number of clinics supported:		12			
3						
4	MedCode	Category	Type	Packs in stock	Packs per clinic	Packs needed
5	MED001	Bandages	plain	307	20	
6	MED002	Bandages	elastic	133	6	
7	MED003	Bandages	triangular	200	12	
8	MED004	Bandages	adhesive	21	1	
9	MED005	Cotton wool	roll	12	2	
10	MED006	Tape	adhesive roll	10	3	
11	MED007	Tape	hypo-allergenic	65	3	
12	MED008	Safety pins	38mm	20	1	
13	MED009	Safety pins	45mm	16	1	
14	MED010	Safety pins	87mm	20	1	

Activity

Open the Mobile Medical Services spreadsheet.

Convert the data into a table.

Calculate packs needed

Your target is to supply 12 clinics. Column F has the heading 'Packs needed'. You will enter a formula into column F to work out how many packs you need.

First, think about how to work out this value. You did this task by hand in the Unplugged activity in the introduction to this unit.

- Column E shows the number of packs needed by one clinic.
- Your target is 12 clinics.

So you must multiply the number of packs by 12 to get the total number you will need.

Enter the formula

You have put the data into a table. That means you only need to enter the formula once, at the top of a column. The computer will copy it down to all the rows below.

Select cell F5. That is the first cell in the 'Packs needed' column.

- Type = to start the formula.
- Click on the cell to the left (cell E5) which shows the number of packs needed per clinic.
- Type the multiply operator *
- Type the number 12.

The formula looks like this:

Packs per clinic	Packs needed
20	=[@[Packs per clinic]]*12

Press 'Enter' and the computer will fill in the answer for every row of the table.

Packs in stock	Packs per clinic	Packs needed
307	20	240
133	6	72
200	12	144

Activity

Enter a formula in cell F5 to work out the total number of packs needed.

Test

1. How many fields are there in the table you made?
2. What are the names of the fields in the table?
3. Which field is the key field?
4. Give examples to explain why the other fields are not suitable as key fields.

Extra challenge

Through research or your own knowledge, think of three additional items that would be needed in an emergency clinic. Create three new records for the data table. Make sure you put suitable data in columns A to E. You will have to make up these numbers.

If you fill in these values, then the computer will work out the total number of packs needed. It will 'copy' the formula down into these rows.

6.2 Surplus or shortfall?

In this lesson

You will learn:

- how to format data to highlight the most important facts.

Your task

From: Director of Mobile Medical Services

Do you have enough stock to supply your target of 12 clinics? Tell me by the end of today if you have a shortfall. I will send emergency supplies by parachute drop.

In the last lesson, you organised the data for this depot. You made it into a table. Now you will use the data table to answer this urgent request. If you have very low stock of any items, you need to arrange a parachute drop for more supplies.

It is expensive and dangerous to supply items by parachute. You must only request items if they are needed urgently.

How spreadsheet features can help

In this lesson you will use **highlighting** to pick out the answers to the question. Highlighting means using colour or other features to pick out key data in a table. In this example you will pick out items in red.

Highlighting is an example of a **conditional format**. It is a format, such as cell colour, which is based on a logical test. The computer will add highlighting if the test is True. In this example, you will pick out items where stock is too low.

Calculate surplus and shortfall

You need to know if you have enough stock in the depot. You have the information you need to answer this question.

- Column D has the number of packs in stock.
- Column F has the number of packs you need.

To work out if you have enough stock, you must calculate the amount in stock **minus** the amount you need.

If the result is 0 or more, you have as many packs as you need. If the number is negative, you have fewer packs than you need. In that case you must ask for more supplies.

Start a new field

You will add a new field to the data table to store this new data. You will use a calculation.

- If the number is positive, it represents a **surplus**. That means you have more packs than you need.
- If the number is negative, it represents a **shortfall**. That means you need more than you have. You will need more packs.

You can call the new column 'Surplus/Shortfall'. Make a new column heading.

Packs needed	Surplus/Shortfall
240	
72	
144	

The table automatically expands to include the new column. You might need to make the column wider to fit the new text.

Enter a formula

Remember, the formula is the number of packs in stock minus the number of packs needed. You will enter this formula in the first row of the table. The computer will fill in the answer in every other row.

The formula looks like this:

Packs in stock	Packs per clinic	Packs needed	Surplus/Shortfall
307	20	240	=[@[Packs in stock]]-[@[Packs needed]]
133	6	72	
200	12	144	

Use the skills you have learned to make this formula.

1. Type the equals sign.
2. Click on the first value in the 'Packs in stock' column.
3. Type the minus sign.
4. Click on the first value in the 'Packs needed' column.

When you press 'Enter', the computer will calculate the right answer for every item in stock.

Surplus/Shortfall
67
61
56
9
-12
-26
29
8
4
8

Activity

Create a new column to record surplus and shortfall. Enter a formula to calculate the surplus or shortfall.

Highlight the shortfall

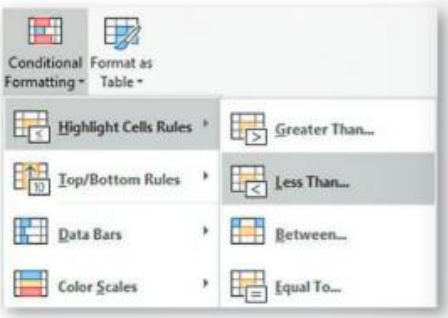
The director has asked you to report if there is a shortfall of any item. Remember, a shortfall means the number you need is bigger than the number in stock. You will have to ask for an emergency drop of these supplies.

If there is a shortfall, you will see a minus number in the new column. Because the table is quite small you can probably look down the list and spot the minus numbers. However, in a large table it is helpful to highlight important facts. Spreadsheets give you tools to do this.

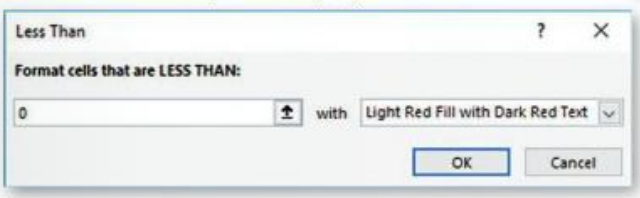
Conditional formatting

First, select the cells in column G. You can click on the G at the top of the spreadsheet to select the whole column.

There is a button in the 'Styles' section that says 'Conditional Formatting'. Click to open a menu. Select 'Highlight Cells Rules'. You will highlight cells with a value 'Less Than…'



A window will appear. You have to enter a number. Enter 0. This will tell the computer to highlight values less than 0.



Click on OK. You will see that negative numbers in column G are highlighted in red. This makes it easy to spot them.

Activity

Use conditional formatting to highlight the items where there is a shortfall.

Extra challenge

Three students have come to work with you as volunteer helpers. You have given them the job of checking all supplies in the depot. They find that some of the items have been damaged by floodwater. Here is the note you receive from the student volunteers.

REPORT TO THE MANAGER OF DEPOT FOUR

Our stock check has discovered:

- There are only 2 undamaged packs of cotton wool.
- There are 3 kerosene lamps – the rest are broken.
- There are no rolls of adhesive tape left in the depot.

Adjust stock levels to match this information.

Prepare a short email to the director. Tell him the names of all items where there is a shortfall, and the size of the shortfall. This will help him to arrange emergency supplies.

Test

Mobile Medical Services also keeps records of the numbers of doctors and nurses working in the different clinics. Here is part of that spreadsheet.

	A	B	C	D
1	Clinic name	Doctors	Nurses	Column D
2	Mobile clinic 4a	2	15	
3	Mobile clinic 4b	3	6	
4	Mobile clinic 4c	0	7	
5	Mobile clinic 4d	4	5	
6	Mobile clinic 4e	3	1	
7	Mobile clinic 4f	0	3	
8	Mobile clinic 4g	5	9	

1. You are asked to calculate the total number of staff in each clinic. Which column would hold this data?
2. What formula would you use?
3. No clinic should have fewer than five members of staff. There is a risk to staff welfare. Explain how you could use cell highlighting to identify clinics with too few staff.
4. Explain how you would identify clinics with no doctors.

6.3 Can you do more?

In this lesson

You will learn:

- how to use a data table to check or test different values.

Your task

From: Director of Mobile Medical Services

Emergency supplies will arrive tonight.

- Cotton wool – 50 packs
- Adhesive tape – 50 packs
- Kerosene lamps – 50

The health crisis is getting worse. We want to set up more clinics. How many extra clinics can you supply from your current stocks?

Your depot has a target to supply 12 mobile clinics. In this message the director has asked if you can manage to support more than 12 clinics. Can you increase your target? In this lesson you will see what is possible. You will test out alternatives.

How spreadsheet features can help

Spreadsheet formulas are used to calculate a result. In Lesson 6.1, you calculated the number of packs you needed. In Lesson 6.2, you calculated any shortfall.

Cell reference

Spreadsheet formulas use **cell references**. That means they take values from other cells in the spreadsheet. For example, when you calculated the number of packs needed, you used the value in the 'Packs per clinic' column. You multiplied it by 12.

Copy down

You made the formula in the top row of the table. The computer automatically copied the formula down to all the rows of the table. It adjusted the formula in each row to use the value from that row of the table.

Recalculate

After you have used a cell reference, you can change the value in the cell. Any formulas that use the value will **recalculate**. Recalculation means the spreadsheet will work out new answers, using the new data. Recalculation lets you try out different values. You can see the effect of each change.

Change values

Emergency supplies have arrived. Your depot now has more items in stock. The message from the director tells you the number of new items.

- Cotton wool: increases to 62 packs.
- Adhesive tape: increases to 60 packs.
- Kerosene lamps: increases to 60 lamps.

Make sure you update your spreadsheet now. Here is the spreadsheet showing the new values.

You can see the effect of recalculation. Values throughout the spreadsheet have changed. The shortfall has disappeared. There are no highlighted cells.

MedCode	Category	Type	Packs in stock	Packs per clinic	Packs needed	Surplus/Shortfall
MED001	Bandages	plain	307	20	240	67
MED002	Bandages	elastic	133	6	72	61
MED003	Bandages	triangular	200	12	144	56
MED004	Bandages	adhesive	21	1	12	9
MED005	Cotton wool	roll	62	2	24	38
MED006	Tape	adhesive roll	60	3	36	24
MED007	Tape	hypo-allergenic	65	3	36	29
MED008	Safety pins	38mm	20	1	12	8
MED009	Safety pins	45mm	16	1	12	4
MED010	Safety pins	87mm	20	1	12	8
MED011	Kidney dish		47	3	36	11
MED012	Instrument tray		34	2	24	10
MED013	Scissors	straight	50	3	36	14
MED014	Scissors	curved	15	1	12	3
MED015	Bowls	0.5 litre	36	2	24	12
MED016	Bowls	2 litre	30	2	24	6
MED017	Trolley		16	1	12	4
MED018	Stretcher		20	1	12	8
MED019	Surgical gloves	size 6.5	20	1	12	8
MED020	Surgical gloves	size 7.5	45	3	36	9
MED021	Surgical gloves	size 8.5	20	1	12	8
MED022	Kerosene lamp		60	2	24	36
MED023	Stethoscope		60	3	36	24
MED024	Steam steriliser	15 litre	30	2	24	6

Use a new formula

The number of clinics is set at 12. This value is stored in cell C2. Now you will change the formula that calculates how many packs are needed. You will include a cell reference to cell C2.

Then you can try new values in C2. The computer will recalculate based on the new values.

Use a cell reference

You will change the formula in the 'packs needed' column. At the moment the formula looks like this.

=[@[Packs per clinic]]*12

This takes the number of packs per clinic from column E. It multiplies this value by 12. Now you will change the formula. Instead of multiplying by 12 you will multiply by the value in cell C2.

Click on the formula at the top of the 'Packs needed' column. Delete the number 12 and click on cell C2. Press Enter to complete the formula.

Mobile Medical Services - Supply Depot Four

Number of clinics supported: 12

MedCode	Category	Type	Packs in stock	Packs per clinic	Packs needed	Surplus/Shc
MED001	Bandages	plain	307	20	=[@[Packs per clinic]]*C2	
MED002	Bandages	elastic	133	6	72	

The formula now looks like this

=[@[Packs per clinic]]*C2

Problem!

You will see a problem. All the results in the column have gone wrong!

Packs needed	Surplus/Shortfall
240	67
0	133
#VALUE!	#VALUE!
#VALUE!	#VALUE!
#VALUE!	#VALUE!
#VALUE!	#VALUE!
#VALUE!	#VALUE!
#VALUE!	#VALUE!
#VALUE!	#VALUE!
#VALUE!	#VALUE!

This is because the computer has tried to copy the formula down the table. In every row the cell reference has changed. Click on any cell in the 'Packs needed' column and you will see this error. It changes from C2 to C3, C4, C5, etc. That gives the wrong result.

Fix the error

Sometimes you don't want the computer to change the formula. You want it to stay the same in every row of the table. That's what you need this time.

Luckily there is a way to fix this. You put the dollar symbol $ next to any value you don't want to change. The name for this is an absolute cell reference. An **absolute cell reference** will stay the same in every row of the table.

Go to the top of the 'Packs needed' column. Put a dollar sign in front of the 2.

```
=[@[Packs per clinic]]*C$2
```

You now see the correct results in all rows of the table.

Packs needed	Surplus/Shortfall
240	67
72	61
144	56
12	9
24	38
36	24
36	29
12	8
12	4
12	8
36	11
24	10

Activity

Make changes to the spreadsheet table as shown in this lesson.

- Change the number of items in stock.
- Change the formula for packs needed.

Test and check

Now you can change the value in cell C2. Changing the number will test how many clinics you can supply without a shortfall.

For example, if you increase the number of clinics from 12 to 13 there is no shortfall.

Mobile Medical Services - Supply Depot Four

Number of clinics supported: 13

MedCode	Category	Type	Packs in stock	Packs per clinic	Packs needed	Surplus/Shortfall
MED001	Bandages	plain	307	20	260	47
MED002	Bandages	elastic	133	6	78	55
MED003	Bandages	triangular	200	12	156	44
MED004	Bandages	adhesive	21	1	13	8
MED005	Cotton wool	roll	62	2	26	36
MED006	Tape	adhesive roll	60	3	39	21
MED007	Tape	hypo-allergenic	65	3	39	26
MED008	Safety pins	38mm	20	1	13	7

But if you increase the number of clinics to 20 there is a lot of shortfall.

The answer must lie somewhere between the two.

Packs needed	Surplus/Shortfall
400	-93
120	13
240	-40
20	1
40	22
60	0
60	5
20	0
20	-4
20	0
60	-13
40	-6
60	-10

Activity

Try entering different numbers in cell C2. Find the maximum number of clinics you can supply without any shortfall. When you get the number right, no value in the 'Surplus/Shortfall' column will be smaller than 0.

Extra challenge

Your depot could support 20 clinics – but only if you get more supplies to cover the shortfall. Find out more about this using the spreadsheet to help. Write a message to the director that starts like this.

From: Supply Depot Four

We can support 20 clinics. But we will need the following additional supplies:

Finish the message with details of the supplies you need.

Test

Test questions 1–3 relate to the following formula:

=120*A4

1. If cell A4 held the value 20, what would be the result of this formula?
2. What is the name of a cell reference that does not change when it is copied down a table?
3. Show how you would rewrite this formula so that it does not change when it is copied down the table.
4. Explain how you used recalculation in this lesson to answer the director's question.

6.4 What to order?

In this lesson

You will learn:

- how to analyse data to provide guidance for actions.

Spiral back

In this lesson you will use the spreadsheet IF formula. This is similar to 'if… else' structures in programs. The 'if… else' structure is found in most programming languages including Scratch and Python. Your learning in programming units will help you complete this task.

Your task

From: Director of Mobile Medical Services

We have opened the road through to you. That means we can begin regular deliveries by supply truck. Each week, send us a list of the items you need.

Now that the road is open, your depot can receive regular supplies. But which items should you ask for? Which are needed most urgently? In this lesson you will learn how to use spreadsheet features to provide guidance for actions. The spreadsheet will help you to select the items that you need from the supply truck.

How spreadsheet features can help

In this lesson you will use an **IF formula**. This is a spreadsheet formula. It works like the 'if… else' structure in a program.

An IF formula begins with

```
=IF()
```

Three items go inside the brackets (separated by commas):

- a logical test
- the output if the test is True
- the output if the test is False.

Compare to highlight

Using an IF formula is an alternative to using a highlight rule. A highlight rule applies one condition to the whole column. An IF formula lets you set different logical tests for different items in the table.

Reorder level

A stock data table like this often includes a **reorder level**. The reorder level is a minimum level for stock. It is more than zero. If stock gets below the reorder level, that is a sign that you need to order more. In this lesson you will use the IF formula to warn you when stock goes below the reorder level.

Create a new column of the spreadsheet table. Enter the heading 'Reorder level'.

What level?

The reorder level is different for different items. Some items, such as bandages, get used up very quickly. Other items, such as a stretcher, are needed in smaller numbers.

In this spreadsheet you will set the reorder level for each item of stock. To set the reorder level multiply 'Packs per clinic' by 2. If stocks fall below that level you will reorder new stock.

Enter a formula

Enter a formula in the first cell of the reorder column. The formula must multiply 'Packs per clinic' by 2. Use the skills you have learned to complete this formula.

You should see results like this:

Reorder level
40
12
24
2
4
6
6
2
2
2
6
4

Add a new column with the heading 'Reorder level'.

Add a formula to calculate the reorder level for every item in the data table.

Reorder message

Create a final column for the stock table. Give it the heading 'Reorder message'.

Now you will enter an IF formula in this column. If the surplus stock is below the reorder level, the column will show the message 'Reorder this item'. Otherwise, it will show the message 'OK'.

Plan

This plan sets out the parts of the IF formula.

Logical test	Surplus < Reorder level
If the test is True	"Reorder this item"
If the test is False	"OK"

Make the formula

Select the first cell in the 'Reorder message' column.

Start the formula

Begin the formula by typing:

```
=IF(
```

Logical test

The next thing to enter is the logical test.

- Click on the cell that shows the stock surplus/shortfall.
- Type the 'less than' operator: <.
- Click on the cell that shows the reorder level.

Your formula now looks like this:

```
=IF([@[Surplus/Shortfall]]<[@[Reorder level]]
```

If the test is True

Now tell the computer what to show if the test is True.

- Type a comma.
- Enter the message "Reorder this item". Remember to include the quotation marks.

Your formula now looks like this:

```
=IF([@[Surplus/Shortfall]]<[@[Reorder level]], "Reorder this item"
```

If the test is False

Complete the formula by typing a comma and a message to show if the test is False, for example, the message "OK". Don't forget the quotation marks and to close the brackets at the end.

Your formula now looks like this:

=IF([@[Surplus/Shortfall]]<[@[Reorder level]], "Reorder this item", "OK")

Press 'Enter' and the formula will copy down to all the cells in the column.

Result

The result of the formula looks like this. The message tells you every item that needs to be reordered.

Reorder message
Reorder this item
OK
Reorder this item
OK
OK
OK
OK
OK
Reorder this item
OK
Reorder this item
OK
Reorder this item

Activity

Create a new column called 'Reorder message'.

Enter an IF formula to show the reorder message for every item with stock below the reorder level.

Extra challenge

Add conditional formatting to the column that shows the reorder message. If the cell has the contents 'Reorder this item' then highlight the cell. For more of a challenge, use green highlighting.

Reorder message
Reorder this item
OK
Reorder this item
OK
OK
OK
OK
OK
Reorder this item
OK
Reorder this item
OK
Reorder this item
Reorder this item

Test

Here is an example of an IF formula.

=IF([@[Stock]]<0, "URGENT REORDER", "Not urgent")

1. The logical test includes a relational operator. What relational operator?
2. What message is shown if the logical test is True?
3. You should order stock before the level reaches zero. Why is that a good thing?
4. The reorder level for bandages is higher than the reorder level for stretchers. Give one reason for this.

6.5 Summary data

In this lesson

You will learn:

- how to calculate summary data from records.

Spiral back

In Student Book 7 you used the AutoSum button. It adds together a group of values. You will use AutoSum again in this unit. If you can't remember what AutoSum is, look back at Student Book 7, Unit 6.

Your task

From: Director of Mobile Medical Services

Congratulations on running the supply depot for four months. How much stock is left? How much have you used?

You have been manager of the supply depot for 4 months. Your work has supported 16 mobile clinics. You have kept careful records. You have recorded how much stock was used each month.

Now you will produce summary data. **Summary data** means a result calculated from a group of figures. Examples would be totals, averages and other statistics. Summary data gives us the big picture rather than the detail.

In this lesson, you use summary data to find out:

- how much stock has been delivered
- how much stock has been used.

In the next lesson, you will use this data to make estimates for the year ahead.

How spreadsheets can help

A spreadsheet can contain more than one worksheet. One worksheet might have detailed records. The other might have summary data.

In this lesson you will use a spreadsheet with two worksheets:

- **Deliveries:** A worksheet that contains detailed records of deliveries each month.
- **Stock count:** A worksheet that gives summary data, such as the 4-month total.

By using cell references, you can take data from the first worksheet and use it in the second worksheet.

Depot summary

Download the spreadsheet called 'Depot summary'.

This spreadsheet has two worksheets. They are called 'Deliveries' and 'Stock count'. Look at the bottom of the spreadsheet. You will see two tabs. The tabs show the names of the two worksheets.

Total deliveries

Select the worksheet called 'Deliveries'. This worksheet shows the amount of each item delivered to the depot in March, April, May and June. The spreadsheet is already formatted as a data table.

Add a new column to the right of the table. Enter the heading '4-month total'. Select the top cell of this column and click on the AutoSum button.

March	April	May	June	4-month total
40	50	50	0	140
10	10	10	0	30
25	0	25	30	80
2	0	2	2	6
5	2	2	2	11
5	0	0	3	8
5	0	0	1	6
2	2	2	2	8
2	2	2	2	8
2	2	2	2	8

By clicking on the tabs, you can swap between the two worksheets.

The computer adds together the four monthly figures (from March, April, May and June) to give the total deliveries. The computer copies the formula down to all the other rows of the table.

Activity

Load the spreadsheet called 'Depot summary'. Open the worksheet called 'Deliveries'.

Add a column to the worksheet to show the 4-month total. Use AutoSum to calculate this value for every row.

Stock count

The spreadsheet has two worksheets. The second worksheet is called 'Stock count'. Click on the tab to open this worksheet on your screen. It looks like this.

This worksheet shows the amount of stock at the start of the 4-month period ('Starting') and the amount of stock left at the end ('Remaining'). Student volunteers found this data by counting the packs on the shelves.

	A	B	C	D	E
1	Supply Depot Four - Stock count				
2	Number of clinics :		(this year)	16	
3					
4	MedCode	Category	Type	Starting	Remaining
5	MED001	Bandages	plain	7	22
6	MED002	Bandages	elastic	43	20
7	MED003	Bandages	triangular	20	10
8	MED004	Bandages	adhesive	6	6
9	MED005	Cotton wool	roll	32	40
10	MED006	Tape	roll	15	23
11	MED007	Tape	hypo-allergenic	20	12
12	MED008	Safety pins	38mm	5	9

Deliveries

Now you will extend the worksheet. You will add a new field that shows how much stock was delivered in the 4-month period.

First, type a heading for the new column. The heading is 'Deliveries'.

Starting	Remaining	Deliveries
7	22	
43	20	
20	10	

Formula

Now you will use a formula to bring the value across from the other worksheet. If you get stuck, go back to the beginning and start again.

1. Select the first cell in the new column you made.
2. Type an equals sign to start the formula.
3. Open the other worksheet by using the tab at the bottom of the page.
4. Click on the cell that shows the 4-month total.

H5 =Table1[@[4-month total]]

	A	B	C	D	E	F	G	H
1	Supply Depot Four - Monthly delivery data							
2	Number of clinics :		(this year)	16				
3								
4	MedCode	Category	Type	March	April	May	June	4-month
5	MED001	Bandages	plain	40	50	50	0	140

Click on the cell that shows the 4-month total.

Look at the formula at the top of the spreadsheet. It looks like this:

=Table1[@[4-month total]]

This tells you that the formula will use the value labelled '4-month total' from 'Table1' (the first worksheet).

Press 'Enter'. The computer will bring the value from 'Table1'. It will bring the values from every row of the table.

Starting	Remaining	Deliveries
7	22	140
43	20	30
20	10	80
6	6	6
32	40	11
15	23	8

Activity

Open the 'Stock count' worksheet. Add a column headed 'Deliveries'. Enter a formula in this column to bring across the 4-month total from the 'Deliveries' worksheet.

Now make a column with the heading 'Stock used'. Enter a formula in the first cell of this column to do the calculation.

- Type an equals sign to start the formula.
- Click on the cells that store the values you need. Use plus and minus operators to enter the formula.

The image shows what the formula should look like. Press 'Enter' and you will see the calculated results.

Starting	Remaining	Deliveries	Stock used
7	22	140	=[@Starting]+[@Deliveries]-[@Remaining]
43	20	30	
20	10	80	

Extra challenge

Add two new columns to the 'Stock count' worksheet.

- Add a column headed 'Monthly average use'. Calculate this value by dividing the 4-month total by 4.
- Add a column headed 'Stock warning'. Use an IF formula to display a warning if the amount remaining in stock is less than the monthly average use.

Test

A medical volunteer has kept records of the number of patients attending a clinic each day for 2 weeks. The numbers are stored in the cells of a spreadsheet. Answer these questions about the spreadsheet the volunteer made.

1. Summary data means a result calculated from a group of figures. Give an example of an item of summary data you could calculate from the patient numbers.
2. What spreadsheet feature would you use to add up the total number of people attending the clinic in the 2-week period?
3. Explain how you would calculate the average daily attendance at the clinic.
4. The maximum capacity of the clinic is 500 people. But some days more than 500 people attend. Describe how you could use spreadsheet features to highlight this problem.

6.6 Plan for the future

In this lesson

You will learn:

- how to estimate future trends from current data.

Your task

From: Director of Mobile Medical Services

Good news – we have guaranteed funding for next year. Please use your records to estimate your future stock needs.

Your depot supports 16 clinics. In the last lesson you calculated how much of each item was used during a 4-month period. Now you will use that data to estimate the total amount of stock you will need for next year. Mobile Medical Services can use your estimates to plan for the future.

How spreadsheets can help

You can use summary data to estimate future trends. For example, if you know that a tree grew 5 feet last year, you might estimate that it will grow another 5 feet this year. But remember that estimates are not necessarily accurate. Trends that we see in the past might not continue into the future.

To make estimates as accurate as possible:

- base the estimates on reliable data
- take all the important factors into account
- use accurate calculations.

Keeping good records helps you to plan for the future.

Future estimates

Open the spreadsheet file called 'Depot summary'. Select the worksheet called Stock count. The column called 'Stock used' shows how much of each item was used in four months. Four months is a third of a year. So if trends continue then you will use three times this amount of stock in one year.

You have carried out a lot of calculations in this unit. Try to complete this task by working independently. Multiply the stock used by three to give the yearly estimate. Your completed work may look like this. If you have completed the 'Extra challenge' tasks, your spreadsheet may have more columns.

Stock used	Yearly estimate
125	375
53	159
90	270
6	18
3	9
0	0
14	42
4	12
8	24
7	21
11	33
1	3

Activity

Extend the data table to show the yearly estimate for each item.

Projected usage

You have just calculated a yearly estimate. This is based on the results from last year. There were 16 clinics. The number of clinics is shown in cell D2.

	A	B	C	D
1	Supply Depot Four - Stock Count			
2	Number of clinics:		(this year)	16

But what if the number of clinics increases to 20? Then the estimates will change too. In the rest of this lesson you will use the spreadsheet to explore this 'What if' question.

What if the number of clinics increases to 20? How much stock will I need?

To answer this 'What if' question, you will:

- find the amount of stock used by one clinic in a year
- multiply by 20 to give the total stock you will need.

Stock used by one clinic

Add a new column to the worksheet. Give it the heading 'One clinic'.

You can calculate the stock used by one clinic with this formula:

- yearly estimate
- divided by the number of clinics (cell D2).

Remember to put a $ sign into D2, so that it looks like this: D$2. That will fix the cell reference so it does not change when it is copied down.

Here is what the new formula will look like.

Stock used	Yearly estimate	One clinic
125	375	=[@[Yearly estimate]]/D$2
53	159	

Clinics next year

Now you must add a new value at the top of the spreadsheet. You must show the number of clinics for next year. Set the value at 20 for now. Enter this number at the top of the spreadsheet. Here is an example of what it might look like. The new value is in cell F2.

	A	B	C	D	E	F
1	**Supply Depot Four - Stock count**					
2	Number of clinics :		(this year)	16	(next year)	20

Stock for next year

Now create a column to show the amount of stock needed in a year. Enter a suitable column header.

You have worked out the average amount of stock needed to supply one clinic for a year. You have also entered the number of clinics there will be next year.

So, our final best estimate for the amount of stock we need next year is:

- the year average for one clinic
- multiplied by the number of clinics.

The number of clinics is in cell F2. Use a cell reference to F2. Remember to add a $ to make an absolute cell reference.

Yearly estimate	One clinic	Next year
375	23.4375	=[@[One clinic]]*F$2
159	9.9375	

Activity

Find the estimated stock requirements for the depot if the number of clinics increases to 20 in the next year.

Extra challenge

From: Director of Mobile Medical Services

Thank you for the annual estimate. I am planning deliveries for next year. You will get a monthly delivery of stock. Please let me know how many of each item I should send in the monthly delivery.

This is the final request from the director. He wants to know how many items to send in your monthly deliveries next year.

You have an estimated yearly figure. Divide this by 12 to give a rough idea of how much stock will be needed each month. Format the result so it is displayed as a whole number.

Email the director with this information. You can send the email to your teacher to show the work you have done.

Yearly estimate	One clinic	Next year	Per month
375	23.4375	468.75	39
159	9.9375	198.75	17
270	16.875	337.5	28
18	1.125	22.5	2
9	0.5625	11.25	1
0	0	0	0
42	2.625	52.5	4
12	0.75	15	1
24	1.5	30	3
21	1.3125	26.25	2

Test

The depot estimated how many items they would supply to clinics in the year ahead. Estimates are important, but they are never completely reliable.

1. If you have a yearly estimate, how do you calculate a monthly estimate?
2. How can you make sure that your estimates are as good as they can be?
3. Give one reason why the amount of supplies needed by a clinic might be different from the estimate.
4. The manager of the depot used the estimate for the year ahead to plan deliveries. Explain one other reason that the manager of a depot might want to know the amount of items in stock.

Explore more

Carry out independent research into the cost of the spreadsheet items, for example, by looking at online shops. Add the information you have found to the spreadsheet. Multiply the number of items used in a year by the cost of each item. What is the total cost of all the items used in one year?

Digital citizen of the future

Young people do voluntary work for charities and welfare organisations. It is a good way to gain experience and help people. This unit is about using computer skills to help a medical charity. But there are many different types of volunteer work that you can try. You don't always need specialist skills. When you are old enough, look out for adverts and appeals for help.

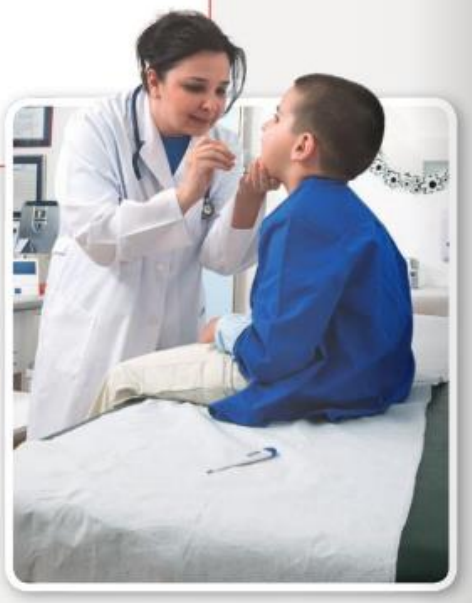
Be creative

Produce a poster for Mobile Medical Services. It should encourage people to make donations to buy supplies. Use information from the spreadsheet in the poster (for example, 'Just one of our clinics uses xxxxx bandages a year').

Check what you know

You have learned

- how to analyse data stored in a data table
- how to use computer data to help with decision making.



Try the activities and test. They will help you to see how much you understand.

Activities

A temporary child health clinic opened for six weeks. This data table shows the number of children treated at the clinic each day during the first week. It also shows the number of doctors attending the clinic each day.

	A	B	C
1	Day	Children treated	Doctors
2	Monday	150	3
3	Tuesday	90	5
4	Wednesday	230	5
5	Thursday	80	4
6	Friday	123	6
7	Saturday	90	3
8	Sunday	219	6

1 Create a data table to store this data. Add a calculation to work out the total number of children treated during the week.

2 Create a new column called 'Patients per doctor'. Enter a formula to calculate the number of patients seen (on average) by each doctor, on each day of the week.

3 The director has said that doctors should not have to treat more than 30 patients a day. Use highlighting to show any days when this value was exceeded.

Test

Here are the results for the next week of the clinic. The table has been extended to show the average number of children treated per day.

	A	B	C	D
1	Day	Children treated	Doctors	Patients per doctor
2	Monday	180	3	60
3	Tuesday	110	5	22
4	Wednesday	210	5	42
5	Thursday	88	4	22
6	Friday	105	6	17.5
7	Saturday	78	3	26
8	Sunday	222	6	37
9	**TOTAL this week**	**993**		
10	**Average per day**	**141.9**		
11				

1 How many fields are there in this data table?

2 What formula would you use to calculate the average number of children treated each day?

3 How would you use this data to estimate the total number of children the clinic could treat in one year?

4 The value in cell B5 was changed. List the other cells where the values will change because of this.

5 This spreadsheet shows the estimated number of patients per day. How can this information help you to plan the number of doctors who should be on duty at the clinic?

6 How could you improve the accuracy of this estimated data?

Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1.
- I answered test questions 1–4.
- I completed activities 1 and 2.
- I answered all the test questions.
- I completed all the activities.

Re-read any parts of the unit you feel unsure about. Try the activities and test again – can you do more this time?

Glossary

absolute cell reference a cell reference that does not change when it is moved or copied to a new location in a spreadsheet table. In Excel, you add a $ symbol to a cell reference to stop it from changing

algorithm a plan that sets out the steps to solve a problem or complete a task. An algorithm can be used to plan a program. Sometimes there is more than one algorithm that will solve the same problem

append 'add to the end'. You can append items to a list

artificial intelligence a technique where computers are programmed to imitate the way humans think and act

aspect ratio the relationship between the width and height of a screen or image. Common aspect ratios in photography and video include 4:3 and 16:9 (also often called 'widescreen')

assistive technology hardware and software that is designed to help people with disabilities to access and use computer systems

AutoSum a shortcut button which adds up all the values in a group with one click

binary search a search that splits the list in two repeatedly until the list has only one item in it

body Python structures are made out of a header and a body. The commands inside the body are indented. These commands are controlled by the header

broadband network a way of sending large amounts of data quickly between computers

built-in functions provided ready-made as part of Python. For example, `input` and `print`. You do not have to make these functions, you can just use them in your code

call a procedure enter the name of a procedure into your program. When the computer sees the name of the procedure it will carry out all the commands stored in the procedure

cell reference every cell in a spreadsheet table is identified by a letter and number, such as A4. If you enter the cell reference into a formula, the computer will take the data stored in that cell and use it in the calculation

cloud used to describe internet services, in particular internet data storage

computer models computer programs that simulate processes in the real world

conditional format a spreadsheet format, such as cell colour, which is based on a logical test. The computer will add the format if the test is True

data centre a dedicated facility connected to the internet where data can be stored, distributed and processed

data structure a variable that can hold many values. A list is an example of a data structure

digital made up digits (number values)

digital asset any digital file such as a document, video or image that you can store and use in projects

digital data data that is converted into number values. Digital data can be stored and processed by a computer

edit make changes to something. You can edit a file, a document, or a single item of data

e-learning using computer systems to deliver learning, usually at a distance

element an item in a list

fibre-optic cable a type of cable made from clear fibres, used to connect network devices. Carries data as pulses of light

fields the columns of a data table. Each field stores one piece of data

function a module or procedure which creates a new value

guidance computer a computer used to guide aircraft, space vehicles and cars

header many Python structures such as loops, `if` statements and procedures have headers. The header controls the structure. The header always ends with a colon. The header is followed by the body of the structure

highlighting using colour or other visual features to pick out key data, for example in a table

hotspot a location in a public place (e.g restaurant or train) where you can make an internet connection

hub a hardware device that directs files and messages around a network. A hub sends data it receives to every device connected to it

hub room a room containing switches and hubs that is used to extend a network to distant parts of a building

IF formula a spreadsheet formula with a conditional structure. It works like the 'if… else' structure in a program

index number tells you the position of an element in the list. In Python, the first position is numbered 0. The index number is shown in square brackets.

integer division a type of division that produces a whole number result. The answer is rounded down to the nearest whole number. In Python the symbol `//` is used

interactive video a video presentation that can be controlled by the viewer. For example, clickable areas on-screen may change the sequence of the video

interface the part of a program that handles user input and output

key field stores data that is unique for each record in a data table. It is often a code number

linear search a search that goes through all the elements in a list, one after the other, to find the search term

list a special kind of variable. A list can store several different items of data. In Python, a list is shown inside square brackets

local area network (LAN) a network installed in a single building, or a group of buildings that are close together. A school network is an example of a LAN

metrics numbers used to measure and compare the quality or performance of software and hardware. For example, a computer processor's speed is measured in gigahertz (GHz). You can use the number to compare it with other computers

midpoint the element at the middle of a list

midpoint value the value stored at the midpoint of a list

modem a hardware device that translates data sent on one type of network so that it can be used on another

modular programming programming that uses modules (such as procedures and functions)

module a ready-made block of code, with a name, that you can use in your program. Procedures and functions are examples of modules. They can also be called 'sub-procedures' or 'routines'

multimedia more than one digital media (e.g text, video) used together to communicate ideas

multi-player game a game played on the internet where multiple players compete and cooperate with each other

nesting when a program structure is inside another structure. For example, an 'if' structure may be 'nested' inside a 'while' loop

network computers that are connected so that they can share files and resources such as printers

network interface card (NIC) a hardware device inside a computer that allows it to connect to a network. NICs are used for both wired and wireless connections

network security key a code that enables a user to log on to a wireless network. A network security key acts as a password

network storage large storage devices on a network that are shared by many users to store data files

options analysis deciding what software, hardware and online services to use for a project by reviewing them against the requirements

out of bounds error if you use an index number that is too big for a list, this causes an out of bounds error

packets small blocks of data that can be sent over a network. Files and messages are split into small packets before being sent over a LAN or the internet

packet switching a method of sending data over a network where packets are sent along the clearest route to their destination

parameter a value sent from the main program to a procedure

pass a parameter sending a value from the main program to a procedure. The parameter is (typically) a copy of the value

prioritisation ranking requirements or activities in the order of importance. You can divide requirements into 'must have', 'should have' or 'could have', depending on how important they are for your project

procedure a module that stores commands. A procedure does not make a new value. A procedure is like a little program that does one task

procedure definition the code that makes a procedure. In Python, a procedure definition is made of a header and a body

project brief a document that explains the purpose of a project

prototype a model of the finished product, such as a drawing or storyboard. Prototypes can be low-fidelity or high-fidelity, depending on when they are created

recalculate when data values are changed, the computer will recalculate results. That means it will work out new answers, using the new data

records the rows of a data table. Each record stores all the data about one thing (for example an object or person)

remove a Python command that deletes a value from a list

reorder level the minimum level for stock. If stock gets below the reorder level, that is a sign that you need to order more

requirement a feature or function of software or hardware that you want for your work

resolution the number of pixels used to create an image. Resolution is often expressed as the number of pixels in the width and height of the image, such as 1920 × 1080 (also known as Full HD)

robust program a program which does not crash, even if the user inputs bad data

router a hardware device that joins two networks together. Used to join a LAN or home network to the internet

search term the item you are looking for in a search

server a computer on a network that provides a service to users, for example, storing, retrieving and sending computer files for users

server room a room at the centre of a network that contains servers, switches and other network equipment

shortfall in stock control, a shortfall means you have fewer items than you need

simulation a computer program that simulates real life events, for example, chemical experiments or geographical events like earthquakes

speech generator a device that turns digital text into simulated speech. It is used by people with disabilities that affect their ability to speak

stop value the number that stops a counter-controlled loop

storyboard drawings and notes that show how a finished video or other media product should look. Use storyboards when you need to decide what order to put things in

streaming a way of accessing audio and video resources over the internet. Streaming allows a resource to be played before it has fully downloaded

summary data a result calculated from a group of figures. Examples would be totals, averages and other statistics

supercomputer a large computer system with very large storage capacity and processing power. Supercomputers are used to process large amounts of complex data in scientific applications such as weather forecasting

surplus in stock control, a surplus means you have more items than you need

switch a hardware device that directs files and messages around a network. A switch sends data it receives only to the device it is intended for

template a design that is applied to presentations, web pages and documents. A template determines how elements of a screen or document fit together

theme a design that is applied to presentations, web pages and documents. A theme determines the colour, fonts and graphical images used in a design

thumbnail a small copy of an image that is used in an application like File Explorer. Clicking on a thumbnail image often opens the larger version of the image

traversing visiting, printing or looking at every value in a data structure

twisted pair cable a type of cable made from copper wire that is used to connect network devices. Carries data as pulses of electricity

validation a check which uses rules to block bad inputs is called a validation check

virtual reality a computer-generated simulation of a real-life or imaginary environment. The viewer can move around the environment and interact with it using a special headset and sensors

web conferencing holding a meeting over the internet

wireless access point (WAP) a hardware device that provides wireless access to a network

worksheet a spreadsheet table. There can be more than one worksheet in a spreadsheet file

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford, OX2 6DP, United Kingdom

Oxford University Press is a department of the University of Oxford. It furthers the University's objective of excellence in research, scholarship, and education by publishing worldwide. Oxford is a registered trade mark of Oxford University Press in the UK and in certain other countries.

First published in 2020

British Library Cataloguing in Publication Data

Data available

ISBN 978-0-19-849786-8

1 3 5 7 9 10 8 6 4 2

Paper used in the production of this book is a natural, recyclable product made from wood grown in sustainable forests. The manufacturing process conforms to the environmental regulations of the country of origin.

Printed in Great Britain by Bell and Bain Ltd. Glasgow

Acknowledgements

The publisher and authors would like to thank the following for permission to use photographs and other copyright material:

All photos © Shutterstock, except: **p36:** PhET ©2019 University of Colorado; **p37(b):** ©2019 Tobii Dynavox. All Rights Reserved.

Cover illustration by Patrick Hruby. All other artwork by QBS Media Services Inc.

Every effort has been made to contact copyright holders of material reproduced in this book. Any omissions will be rectified in subsequent printings if notice is given to the publisher.

Although we have made every effort to trace and contact all copyright holders before publication this has not been possible in all cases. If notified, the publisher will rectify any errors or omissions at the earliest opportunity.

Links to third party websites are provided by Oxford in good faith and for information only. Oxford disclaims any responsibility for the materials contained in any third party website referenced in this work.

Oxford International Lower Secondary

Computing 8

Student Book

This **new edition** of ***Oxford International Lower Secondary Computing*** has been designed to teach students the vital computing skills they need for today's digital world. Across the levels, students grow into digital citizens with the confidence to apply their knowledge and skills in real life situations.

- **More project-based work** better promotes **active learning**, encouraging every student to participate in lessons
- The new **spiral approach to learning** sees students revisiting key concepts to build their digital literacy skills
- New in-built, **three-level differentiation** allows teachers to cater to the needs of every student
- A new focus on **computational thinking** prepares students for the demands of the rapidly evolving digital landscape
- New enhanced Teacher Guides make the course content even more **accessible for all teachers**, whether a computing specialist or not

Also available:

9780198497790 9780198497806 9780198497813 9780198497820 9780198497837 9780198497844 9781382007450 9781382007467

Leading on to:

9780198497851 9780198497875 9781382007474

OXFORD UNIVERSITY PRESS

How to get in touch:
web www.oxfordsecondary.com
email schools.enquiries.uk@oup.com
tel +44 (0)1536 452610
fax +44 (0)1865 313472

ISBN 978-0-19-849786-8