Oxford International Primary

6

Computing

Student Book

OXFORD

Oxford International Primary

6

Computing

Student Book

Alison Page
Howard Lincoln
Karl Held

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.oxfordprimary.com/computing.

Structure of the book

This book is divided into six chapters, for Year 6 (ages 10–11).

1. **The nature of technology:** Introduction to robots and what they can do
2. **Digital literacy:** Creating a web page
3. **Computational thinking:** Making and fixing algorithms
4. **Programming:** Controlling movement
5. **Multimedia:** Collecting and presenting data
6. **Numbers and data:** Structuring, sorting and filtering data

What you will find in each unit

- Introduction: An offline 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 learning activity for the students.


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

Questions check students' understanding of the lesson.

Additional features

You will also find these features throughout the book:

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 Scratch 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, Scratch programming files and downloadable images.

To access the files, click 'Download resources' at:
www.oxfordprimary.com/computing.

Teacher's Guides

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

1 The nature of technology: Robots

You will learn

- what a robot is and how it works
- what control systems can and cannot do
- what robots can do now and how they are developing.

People have been fascinated by robots for hundreds of years. Fictional robots appear in science fiction films such as *Star Wars*. Robots appear in books and in video games. Today, robots are with us in real life. Robots are used to make electronics equipment such as TVs and computers. They are used to build cars. Robots drive cars. We live in an age of robots.

Talk about...

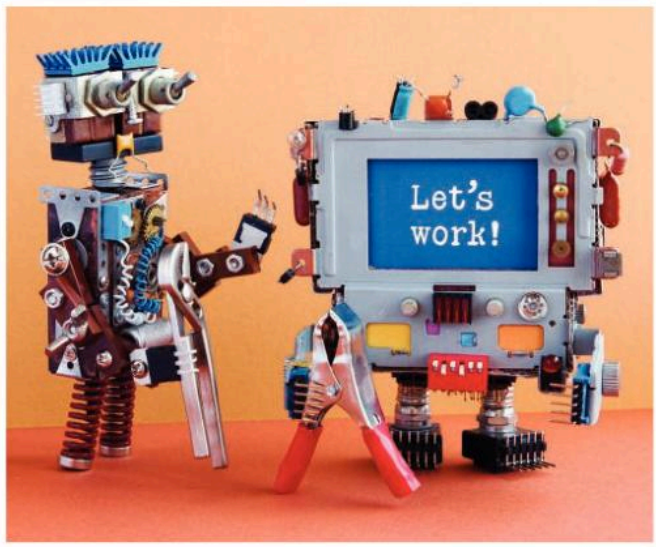
Some robots are designed to look like people. Do you think that makes them more likeable? Or do you prefer robots that don't look like people? Think of examples of robots in films and TV shows. Which ones do you like?

Learning outcomes: Describe the technology of robotics and control systems; Discuss the potential and the limitations of robotics and control systems

Class activity

Design your perfect robot assistant. What will your robot look like? What special tools and powers will it have? What jobs will your robot assistant do that will make your life easier?

robot robot arm drone
actuator sensor
controller control system
artificial intelligence
robot car nanobot



Did you know?

A British scientist called Kevin Warwick had a computer chip implanted into the nerves of his arm. In an experiment that took place in 2002, he controlled a robot hand over the internet. As he moved his hand in the USA, a robot arm in the UK made the same movements. In the future, people who have lost limbs might have mechanical replacements that work just as well as the real thing.

1.1 What is a robot?

In this lesson

You will learn:

- what a robot is and what it looks like.

Spiral back

In Student Book 4, you learned that microprocessors are built into machines and devices that we use every day. Cars, refrigerators and TVs have built-in microprocessors. A microprocessor adds more power to a device and makes it easier to use.

One machine is made more powerful and useful by microprocessors than any other. That machine is the robot. In this unit, you will learn how robots work and what they do.

What is a robot?

A **robot** is a machine. Robots are built by people to do a job. A robot must be programmed by a human to do its job. Once a robot is programmed, it does its job without help from humans.

Unless it breaks down, a robot does its job automatically. It does not need a human to control its actions. A robot does not get bored with its job. It does not make mistakes. The only time a human is needed is if the robot develops a fault.

What does a robot look like?

When people think of a robot, they often imagine them to look like people. Robots in science fiction films are often shown with arms, legs, a body and a head. In the real world, it is different.

Humanoid robots

Humanoid means like a human. Some robots are made to look like people. The robots take part in competitions where they carry out day-to-day tasks just like humans would. They are set challenges such as climbing a flight of stairs and carrying drinks without spilling them.

These robots are used by scientists to develop methods that might be used in the future.

Robot arms

Most robots are built to do jobs that humans do. Humans use their hands and arms to do most jobs. To do the same jobs, a robot has to be able to move like a human. This is the reason that many robots look like a human arm.

A **robot arm** has joints that bend and twist. A robot arm has a hand to grip things with.

Non-humanoid robots

Many robots do a job that humans do without looking human. An example is a robot vacuum cleaner. Another example is a **robot car**. A device does not have to look like a human to be called a robot.

Activity

Invent your own robot which would help with cleaning. Think of a name for your robot. List the main functions of your robot and the advantages of using it to do the cleaning.

Extra challenge

Design a poster to advertise the new robot cleaner.

Be creative

Create an image of the robot cleaner and include it in your poster.

Think again

Is your robot cleaner humanoid or not? Why did you make that choice?

1.2 What robots are used for

In this lesson

You will learn:

- what kinds of jobs robots can do
- how robots are used to build cars.

What jobs can robots do?

Repetitive tasks

Robots do jobs that are repetitive. This means jobs where an action needs to be done in the same way many times. An example is bolting wheels onto a car. Eight thousand wheels are put onto cars every day in a typical car factory. Humans find this kind of work boring. They can get tired and make mistakes or injure themselves. Robots never get bored or make mistakes.

Detailed work

Some jobs involve tasks to be carried out with great accuracy. A single mistake can mean a whole product is spoiled and must be thrown away. Making microprocessors involves a lot of detailed and accurate work. Robots are used to make microprocessors and other electronic equipment such as TVs.

Dangerous work

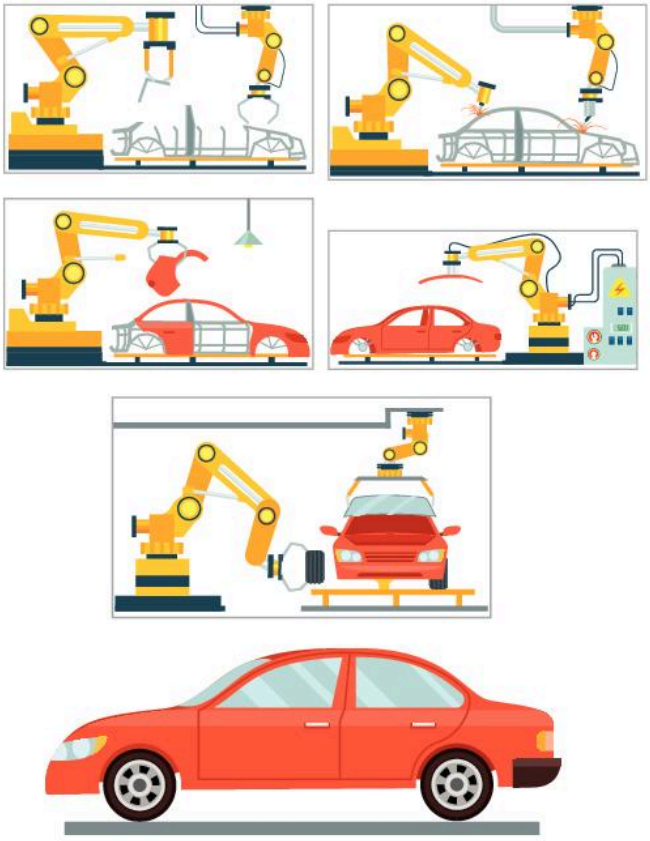
Robots are used in places where it is dangerous for humans to work. For example, robots are used on space missions. Robots called Sojourner and Pathfinder were sent to explore the surface of Mars. The picture gives you an idea of what Sojourner looked like.

Robots are used under the sea. They explore the sea bed or inspect and repair oil rigs. Robots also work in environments where chemicals or radioactivity make it unsafe for humans to work.

Below are the main differences between how robots and humans do jobs.

Robots	Humans
• More accurate than humans	• Intelligent – they can find better ways of doing jobs
• Do not get bored or tired	• Creative – they can invent new things, such as robots
• Work faster than humans	• Solve problems
• Do not make mistakes	• Have emotions and feelings

Robots in the car industry



A car is not built by a single robot. The car moves along an assembly line. The car stops at points along the line. At each point, robots work together to build part of the car. For example, one team of robots will build the engine. Another team will put on the doors.

Robots must work together to make sure every task is performed in the right order. They must avoid colliding with each other. These teams are called **collaborative robots**. Collaborate means to work together.

Painting

Robots are used to paint cars. Robots always give cars a high-quality finish. The paints used on cars are hazardous to humans. Humans do not have to risk their health to do the job.

Activity

Make a poster with the title 'Robots versus humans'. List the strengths of robots and humans in your poster. Illustrate your poster.

Extra challenge

Describe a real-life example where robots are used to do dangerous jobs. Research the web to find your example. Some ideas for your research are fire-fighting, clearing landmines and inspecting pipelines. Are there images or videos of the robots in action?

Think again

Research how robots are used in the car industry. Find examples of jobs that robots are good at. Find examples of jobs that are still done by humans. Write up your findings in a report.

Digital citizen of the future

Some researchers think robots will replace 800 million jobs worldwide by 2030. Workers will need to train to do new jobs. Many of these jobs will be in the robotics industry. Having up-to-date computer skills will be more important than ever. Will you be ready for the new world of work?

1.3 How robots work

In this lesson

You will learn:

- how a robot uses sensors, a controller and actuators to do its work.

What makes a robot work?

You must balance a tennis ball on a book and walk across the room without dropping the ball. How do you do this?

As you cross the room, you use your senses to warn you when the ball is starting to roll. Your brain works out how to stop the ball from falling. Your brain tells your muscles how to move your arm to stop the ball rolling.

A robot works in the same way.

- A robot has **sensors** that act like your eyes and ears. They tell the robot what is happening in the world around it.
- A **controller** is the brain of a robot.
- **Actuators** are the muscles of a robot. They make the robot move.

Sensors

A robot does not have eyes and ears like humans. It uses components called sensors.

- Video cameras give the robot sight.
- **Proximity sensors** tell a robot how close an object is.
- **Bumpers** tell the robot it has collided with something.
- **Pressure sensors** let a robot know that it is touching something.
- Sensors that detect chemicals give a robot its sense of smell and taste.

A robot uses the sensors it needs to do its job.

Controller

A controller is the brain of a robot. The controller is programmed to do a job. A robot cannot think like a human. It carries out whatever instructions it has been programmed to do.

Actuators

A robot does not have muscles and bones like humans. It uses mechanical parts to lift and move objects. The mechanical parts that make a robot move are called actuators. The actuators in a robot do the same job as the muscles in your arm.

How the robot vacuum cleaner works

A robot vacuum cleaner helps to clean a house. A motor is the only extra mechanical part that a robot vacuum cleaner has compared to a normal cleaner. The motor moves the cleaner around the home on wheels.

Proximity sensors measure a room so that the cleaner can calculate the best route to take. Proximity sensors use a light beam to calculate distance. When the light beam hits something solid, it bounces back to the sensor. The time it takes to bounce back tells the sensor how far the light travelled.

A sensor called a bumper detects if the cleaner hits a wall or piece of furniture. If the cleaner detects a collision, it follows an algorithm. The cleaner makes a small turn and tries to move again. Once it finds a clear path it continues cleaning.

The cleaner has a **vibration sensor** on the base. If a lot of dirt hits the sensor, the vibration it causes tells the cleaner that area needs more cleaning.

A sensor warns the cleaner if the battery power is low. The cleaner finds an electrical charging unit using proximity sensors. The cleaner recharges its batteries before continuing cleaning.

Activity

Give two examples of the sensors a robot vacuum cleaner uses. How are the sensors used?

Think again

What is an actuator? Why does a robot need actuators?

Extra challenge

Have you seen a 'barista' in a coffee shop? A barista is the person who makes the cups of coffee. Could a robot be made that works as a barista? Describe how sensors and actuators might be used by a robot barista.

1.4 What is a control system?

In this lesson

You will learn:

- what control systems are used for
- what a control loop does.

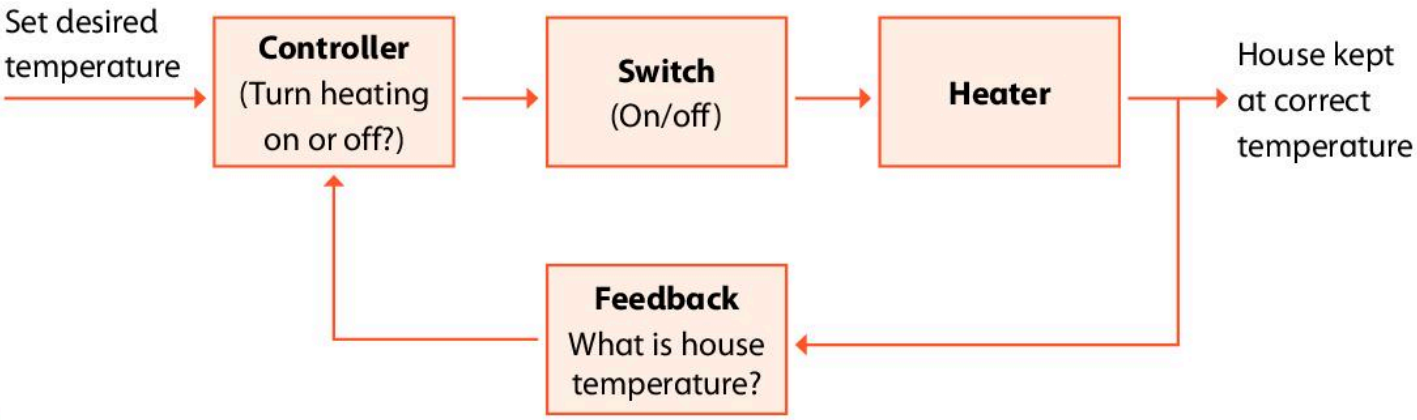
What is a control loop?

If you have a heating system in your home, you may have seen a controller like this:

A heating controller lets you set an ideal temperature for your home. Suppose your ideal temperature is 20 degrees Celsius.

- When the heating is switched on, the controller sends a message to turn on the radiators.
- The message passes through a **temperature sensor**.
- The sensor sends a message back to the controller telling it the actual temperature in the house.
- When the temperature reaches 20 degrees, the controller turns off the radiators.
- The temperature sensor keeps sending messages back to the controller. When the temperature falls below 20 degrees, the radiators are turned back on.

This is a **control loop**. It is sometimes called a **feedback loop**. The sensor feeds back information to the controller.



Control systems in the home

Control systems are installed in many homes. They control temperature, lighting and security.

Modern homes may have many sensors installed:

- Heating systems use temperature sensors.
- Alarm systems use proximity sensors and pressure pads to detect intruders.
- Security systems use video cameras.
- Lighting systems use proximity sensors to detect movement.

Lights can be turned on automatically when a sensor detects someone entering a room. They are turned off again when the sensor detects the room is unused. Heating can be kept at a lower level when the house is empty and warmer when people are in it.

The controller is connected to the internet, so the house owner can control systems from anywhere. Heating can be turned on so the house is warm when people arrive home.

Devices such as ovens can be turned on so food is cooked when people arrive home from work. Messages are sent to the home owner if an intruder is detected. Security cameras can be viewed on a smartphone. The home owner can even view the contents of their fridge to find out what they need to buy from the supermarket.

A house with control systems is called a **smart home**. Control systems make life easier, more secure and reduce energy use.

Activity

You are travelling home in your robot car with your parents. What messages can you send to your smart home so that it is welcoming when you arrive?

Think again

A feedback loop can be used to control air conditioning to keep a home at a low temperature. Draw the feedback loop (or control loop) for this process.

Extra challenge

Use your program design skills to write an algorithm. The algorithm should describe how a heating controller keeps the temperature in the house at 18 degrees Celsius.

1.5 Living with robots

In this lesson

You will learn:

- how robots are changing the world you live in.

You have learned how robots help to make things in manufacturing industries. Robots and control systems make life easier in homes. Robots are used in many other industries.

Robots on farms

Harvesters, ploughs and other farm machines are driven by robots. The vehicles find their way around fields using the same **satellite navigation** (satnav) that car drivers use. The farmer does not have to drive farm vehicles so has time to do other tasks on the farm.

A **drone** is a helicopter or plane that does not need a pilot. Drones are robot-controlled. Drones are used to spray pesticides or fertilisers where they are needed.

Robot arms are used to harvest and pack produce. Robots use sensors to decide what is ripe. Robot arms with pressure sensors can pick delicate fruits and vegetables without causing damage.

Shopping with robots

More people buy products online. Online shops store the products in large warehouses. Robots find and pack items when they are purchased.

Online stores are experimenting with using robot vehicles and drones to deliver goods. Some high street stores are using robot shop assistants. They move around the shop and answer any questions that customers have.

Robots and jobs

Some people worry that robots will replace jobs and cause unemployment.

Other people say that robots and technology will replace some jobs but will also create new ones.

- Robots are good at repetitive tasks that need to be done quickly and accurately. Many of the jobs that robots replace are low-skilled. People often find the jobs stressful and boring.
- People are needed to invent and build new robots. New factories are needed for robot-controlled machines. **Computer technicians** are needed to install robots. **Software engineers** write programs to control robots. Technicians are needed to maintain and repair robots. The new jobs are often high-skilled and interesting.

Some people say introducing robots will create more new jobs than are lost.

Activity

Have a look at the two car assembly lines. One is from the 1930s when humans did all the work. The other is a modern assembly line with robots. Describe the differences you can see between the two pictures.

Think again

Which of the two assembly lines would you prefer to work in? Explain why.

Extra challenge

Do research on robots working in agriculture or retail. What kind of robot vehicles and machines are used in these industries?

1.6 Robots of the future

In this lesson

You will learn:

- how robots will develop in the future
- how robots are helping doctors
- how robots will change the way we travel

Artificial intelligence

Robots are being developed that can act more like humans. This is called **artificial intelligence** (AI). AI allows robots to:

- recognise pictures and speech
- learn better ways of doing things
- decide the best way to deal with situations.

In the future, a robot shop assistant may recognise how a customer feels from their voice and face. The robot will take a different approach depending on how the customer looks and sounds.

In the classroom, a robot assistant will learn about your progress and how you study. The assistant will set you individual projects and find learning materials that are best for you.

Robot vehicles

Robot cars are also called autonomous cars. Autonomous means the car works automatically with no need for any human help. An autonomous car does not need a driver. It has sensors that can detect pedestrians, other cars and road signs. The controller uses the information from the sensors to operate the accelerator, brakes and steering.

At any moment there will be different ways a robot car can deal with a situation on the road. For example, does it brake or swerve to avoid a collision? AI will help the car make the best decision.

Cars in the future will look different to cars today. An autonomous car does not need a steering wheel or pedals. Seats do not all need to face forward.

Robots in healthcare

Robots are already being used in modern hospitals. Scanners create a 3D picture of the inside of our bodies so that doctors can diagnose problems. Robots help doctors carry out delicate surgery they could not do without robotic instruments. Robots help doctors but will not replace them.

Robot nursing assistants carry out basic health checks on patients. These robots already help look after elderly patients who need regular care. A robot called Robear can lift patients in and out of their bed or wheelchair. It is more comfortable for the patient and prevents nurses from being injured.

Scientists are researching **nanobots**. A nanobot is a microscopic robot. In the future, nanobots may be able to work inside a patient's bloodstream to fight infections or carry out delicate surgery.

Activity

Travelling in cars is one of the most dangerous things we do. Will autonomous cars make car travel safer or more dangerous? Discuss this in a small group, then write down your own opinions.

Explore more

Choose one use for robots you have learned about in this unit. For example, robots in the home or robot cars. For the area you have chosen, find:

- examples of how robots are used
- photographs of robots
- interesting facts and figures.

Go back over the content of this unit and do extra research using the web.

Present your findings as a presentation, word-processed document or website.

Extra challenge

Carry out more research on either robot cars or robots in healthcare. Make notes. Find examples, facts and pictures.

Check what you know

You have learned

- what a robot is and how it works
- what control systems can and cannot do
- what robots can do now and how they are developing.

Activities

You have been asked to design a robot. The robot should throw tennis balls to a student to help them practise catching. The robot needs to throw the balls safely.

1 Draw the robot. Mark at least one part of the robot used for sensing. Mark at least one part used for moving.

2 Describe the sensors your robot needs to decide where and how far to throw the ball. Describe how the robot might throw the ball.

3 It would be useful for the robot to know if the student has caught the ball. What type of sensor would it need? There is more than one right answer to this question.

If you have finished the activities above and want an extra challenge, try these:

4 The robot could throw the ball to different places – sometimes to the left, sometimes to the right, and at different heights. But not too far away so the catch is impossible. How could you change the robot to work in this way?

5 A robot could use AI to develop the catcher's skills. The robot will learn where the catcher finds it easy or difficult to catch a ball. For example, if the robot learns the catcher finds high balls or balls to the left difficult, it can give the catcher practice with these types of throws. What sensors can you use so the robot knows if the ball has been caught? Write an algorithm to explain how your robot will work.

Test

1. What is a robot?
2. Give an example of a control system that might be used in someone's house.
3. What are proximity sensors? Why might a robot use proximity sensors?
4. Why are robots better at doing repetitive tasks than humans?
5. Describe something you have found out from your own research about robots – either how they are used now or might be used in the future.

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 activities 1–3.

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

2 Digital literacy: Make a web page

You will learn

- how to act respectfully when using a computer
- how to create a simple web page with text and images
- how to review and improve your web pages.

The world wide web (also called the 'web') is an important part of our lives. It gives us information that we need to work, learn and enjoy our leisure time. Anyone can create a web page. A web page lets you share your knowledge and experiences with people across the world. In this unit, you will learn how to create a web page.

Talk about...

What is respectful behaviour? How does respectful behaviour in the classroom make it easier for people to learn?

Should you behave any differently when using computers than you would in everyday life?

Learning outcomes: Create a simple web page with text and images; Review web content you have created to check if appropriate, and amend if needed; Act responsibly and with respect for others when using the computer

Class activity

web page logo
heading image link menu
web page editor respect
be safe

Look at these images of web pages. One is the first web page every created. The other is a modern web page. Can you spot which is which?

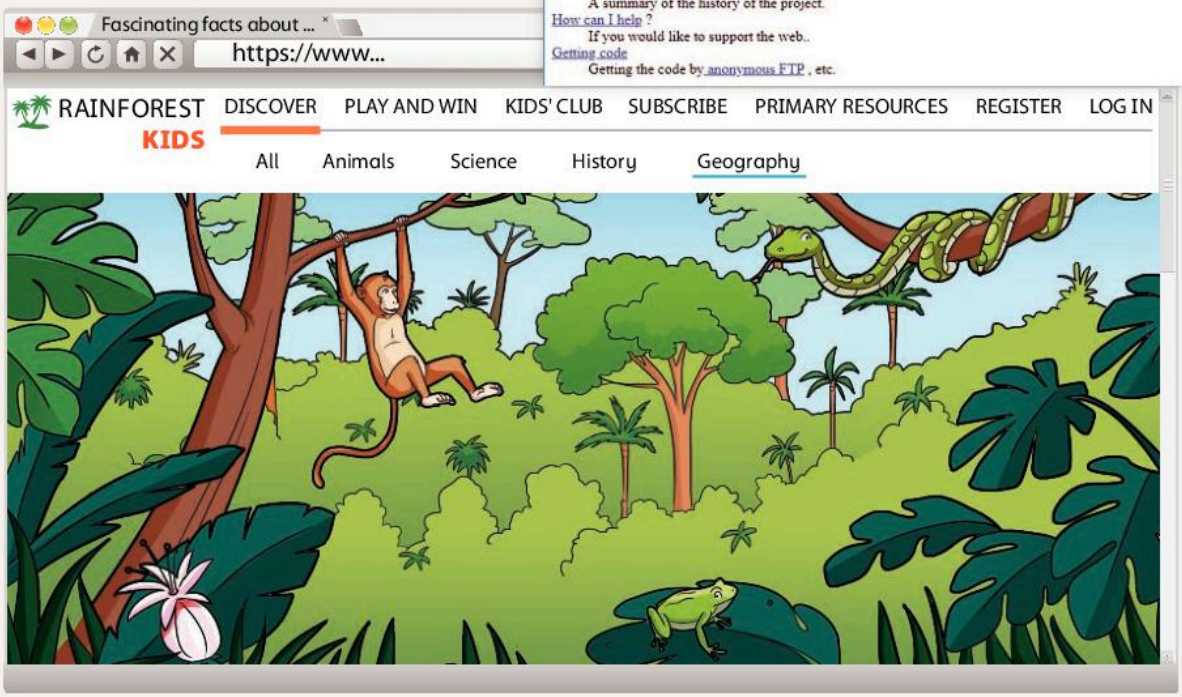
- How have websites changed since the web was invented in 1995?
- Do you think they have changed for the better?

World Wide Web

The WorldWideWeb (W3) is a wide-area hypermedia information retrieval initiative aiming to give universal access to a large universe of documents.

Everything there is online about W3 is linked directly or indirectly to this document, including an executive summary of the project, Mailing lists , Policy , November's W3 news , Frequently Asked Questions .

What's out there?
Pointers to the world's online information, subjects , W3 servers, etc.
Help
on the browser you are using
Software Products
A list of W3 project components and their current state. (e.g. Line Mode ,X11 Viola , NeXTStep , Servers , Tools , Mail robot , Library)
Technical
Details of protocols, formats, program internals etc
Bibliography
Paper documentation on W3 and references.
People
A list of some people involved in the project.
History
A summary of the history of the project.
How can I help ?
If you would like to support the web..
Getting code
Getting the code by anonymous FTP , etc.



Did you know?

The world wide web was invented in 1991. The very first web page has been saved. Look at the web page info.cern.ch and click on the first link in the list you see on the page. It is called 'Browse the first website'.

When you visit the site, you see the entire world wide web as it was in 1991. Today, there are nearly two billion websites!

2.1 Be a respectful computer user

In this lesson

You will learn:

- how responsible behaviour in the classroom makes it a better place to learn
- how being respectful is important when using the internet.

Making the classroom a better place to learn

Using computers for learning is fun. The classroom is a much better place to learn when we behave with respect. You should always behave respectfully towards your teacher and classmates. You should treat equipment with care. Remember, someone else will need to use the equipment after you have finished with it.

This unit gives you a general guide to behaving in the right way when using computers. Your school will have its own rules. Make sure you know them.

Behave respectfully towards your teacher

Working on computer projects often means you work on your own or in small groups. There will be times when your teacher needs to talk to the class. When this happens, stop typing and turn to face your teacher.

Rules for sharing computers and equipment

Remember that you are sharing computers and equipment with other students. Make sure equipment is tidy and in its correct place when you have finished using it. Here are some general rules:

- Be patient and take turns when sharing equipment.
- Do not use equipment for longer than you need to.
- Do not adjust equipment without permission.
- Tell your teacher if equipment is broken or needs attention.
- Leave equipment as you find it.
- Keep your work area tidy. Don't eat or drink at your computer. A spilled drink can be dangerous near electrical equipment.

Being responsible when using online technology

Online behaviour

- Be polite. Think about how you would feel if you received the email or message. Talk to people online as though you are face to face.
- Report anything you see that is frightening or upsetting.
- Help others to report problems.
- Report any behaviour that seems dangerous.

Be safe and help others to stay safe

If a classmate is having a problem using a software application or writing a program, it is good to help. If you are asked, offer an explanation but don't take over. Do not touch your classmate's computer. You could make the problem worse. Offer a solution and let your friend solve the problem.

Activity

Look at the picture of the untidy work area. How do you expect the computer room to look when you arrive to start your class? What problems does an untidy work area cause? List examples where you have found it difficult to work because of damaged or missing computer equipment.

Extra challenge

Work with a partner. Sit at the computer and ask your partner to help you with a problem. For example, you could ask them to explain how to create a new folder.

Let your partner explain the task and help you.

Then discuss how helpful your partner was. Were you confused at any point? Could your partner have explained anything better?

Now change positions so that you offer the help.

Think again

You notice a student is upset by something they have seen on-screen. How could you help them?

2.2 The parts of a web page

In this lesson

You will learn:

- to identify the parts that make up a web page.

Later in this unit you will create your own **web page**. Before you do that, think about what a web page is. What parts – or components – does it contain? The main web page components are shown below.

Components of a web page

Headings: These are short descriptions that tell you what a web page or text is about. **A heading** is larger than normal text so that it stands out.

Images: Most websites use **images** as well as text. Images can be photographs, drawings or cartoons. Images make a web page more interesting and easier to understand.

Links: When you click a **link** on a web page, you leave the page you are on and go to a new page. You use links to find new information on a subject. Links are also called hyperlinks. A hyperlink can be text or an image.

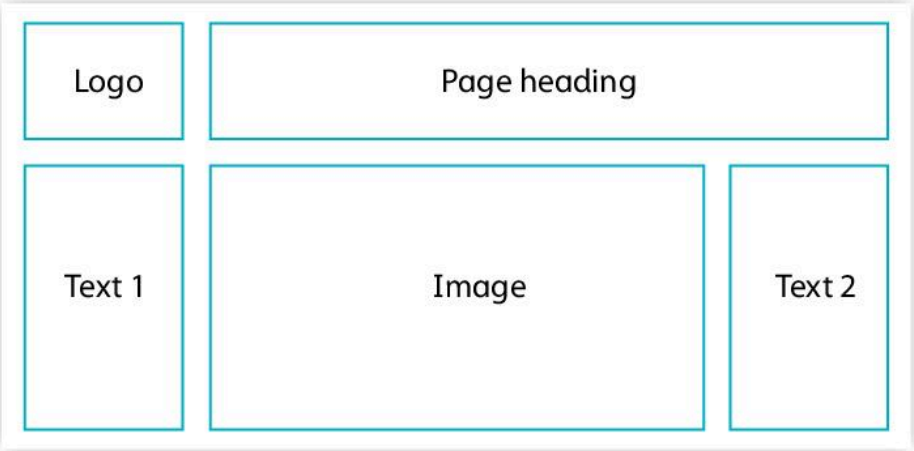
Logo: Most websites use a **logo**. The logo tells you who owns the website. If the website is owned by a company, the company logo will usually be at the top of the web page.

Menu: A **menu** helps you to find your way around the web pages that make up a website. Clicking on a menu option takes you directly to another part of the website.

Text: Text is used on a web page to give information or instructions. Text should be clear and relevant to the subject of the page. Too much text can put off a reader by making a page difficult to read.

Designing a web page

Before you start to create a web page, make a plan. Draw a simple outline plan showing how the components will fit together. Start with a rectangle to represent the computer screen. Then divide the rectangle into smaller sections to show where you will put images, text, and so on. This type of design is called a **wireframe**.



Activity

Your school wants to add a new page to its website. The page will tell parents about a new gymnasium that is being built. The page should include the school logo, a picture showing what the new gym will look like and text describing the gym. Design the page using a wireframe.

Extra challenge

You need to make some changes to your web page design. The design should now include the school logo, two images – showing the inside and outside of the gym, text describing the gym and a heading for the page.

Be creative

Some internet company logos have become very well known.

Design a logo that you can use on your website.

Think again

Open a web page that you use regularly at home or for school work. Identify examples of the six components of a web page.

2.3 Create a web page

In this lesson

You will learn:

- how to create a web page
- how to add text to a web page.

Web page editor

You use an application called a **web page editor** to create web pages.

Your teacher will tell you which web page editor to use at school. There are free web page editors that you can use at home. This unit uses a web page editor called Wix. Ask permission from an adult before using web editing software at home.

Sections of a web page

A web page has three sections.

Header: An area at the top of a web page. The header is used for a logo and the name of your web page.

Body: The main part of your web page. It contains the text and images you want to share.

Footer: An area at the bottom of the web page. If you look at the bottom of most web pages, you will see that the footer is used for links to site documents and policies. You won't use a footer in this unit.

Getting started

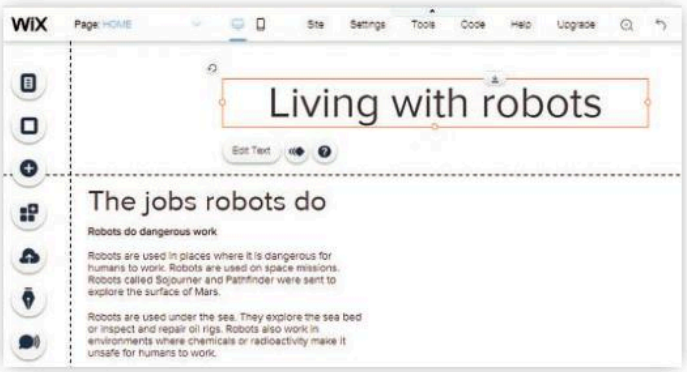
When you open your web page editor for the first time, you will be asked to create a new website. When you create the site, you will be asked to select a template for your site. Choose a blank template. Using a blank template will make it easier for you to learn the basics of web page design.

Adding text to your page

1. Choose the 'Add' option from the toolbar on the left of the screen.
2. Choose 'Text' from the 'Add' menu.
3. Choose a paragraph style from the list if you want to add text. Or choose a heading style to add a heading.
4. A box will appear on your web page. Drag the box to the position you want it on your page. Then type your text inside it.

An example web page

Here is an example of a web page. Notice that the main page heading, 'Living with robots', is above the dotted line. This means it is in the header. The rest of the text is in the body of the page – below the dotted line.



Activity

Design your own web page. It can be about robots or any other topic you choose. Choose a title for your page. Draw a wireframe plan that will show where you will place headings, text and images.

Be safe and responsible

When you publish a web page it can be read by anyone in the world. Do not include personal details on your web page. You must keep your full name, your address and phone number private. Do not include any information that will allow someone to identify or locate you.

Activity

Use a web page editor to create your web page. Add a page title, and text with a heading.

Extra challenge

Add a text box to your web page with the heading 'Did you know?' Do some research and add an interesting fact about your subject in the text box.

Think again

How many types of heading are used in the example web page? Why are some headings larger than others?

Digital citizen of the future

People often use websites to find out information. Others share information through their websites and social media pages. How will you use the web to share your knowledge in the future?

2.4 Add images to your web page

In this lesson

You will learn:

- how to add images to a web page.

Spiral back

Last year, you learned that it is important to respect copyright when using other people's content. You should only use content on your web pages if you have permission. A tool like Wix will provide you with images that you can use. You also learned how to find and use Creative Commons images. If you are in any doubt about copyright, ask your teacher.

Choosing the right images

Adding an image to a web page makes the page more interesting. It can also make the web page easier to understand. Make sure you choose images that work with the text on your web page.

Be safe and responsible

Be careful about the images you add to web pages. For example, do not add photos of yourself. If you feel there is a good reason to add an image of yourself, talk to an adult first. Never include images of friends or family without their permission.

An example web page

In the last lesson, you learned how to add text and headings to a web page. The 'Living with robots' example showed you how to position text and headings in the body and header of a web page.

Here is what the 'Living with robots' web page looks like when images are added. Two images have been added to the page. A logo has been added to the left of the web page heading. The logo is in the page header. An image has been added to the right of the text, in the body of the page.

How to add images to a web page

Here are the steps to follow to insert an image on your own web page. Remember, the web page editor used in this book is Wix. If you are using a different web page, menus and tool bars might be slightly different.

1 Choose the 'Add an Image' option from the tool bar on the left of the screen.

2 Use the 'Free Wix Images' option for this activity.

3 Select an image you want to add to your page.

4 Resize the image. Drag it into position.

Activity

Open the web page you created in the last lesson.

Add an image to the body of your page. Choose an image that works with the text on the page.

Extra challenge

In the last lesson you added a 'Did you know?' section to your web page.

Search the web for a suitable image for this section. Add the image to the 'Did you know?' section.

Think again

Here are two paragraphs from a web page about robots. Which image (A, B or C) would you choose to illustrate each paragraph?

- A robot arm has joints. The joints twist and bend like those in a human arm. A robot arm can do jobs that humans do with their arms.
- Robots are used to make cars. In a car factory, robots work together in teams. Each robot does its own job in turn.

A

B

C

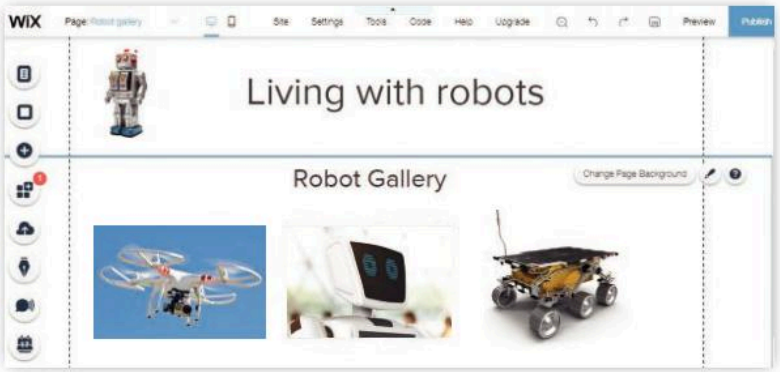
2.5 Add a new web page

In this lesson

You will learn:

- how to add a new web page to a website
- how to add a menu to a web page.

In this lesson you will add another web page to your 'Living with robots' website. This web page will be called 'Robot gallery'.

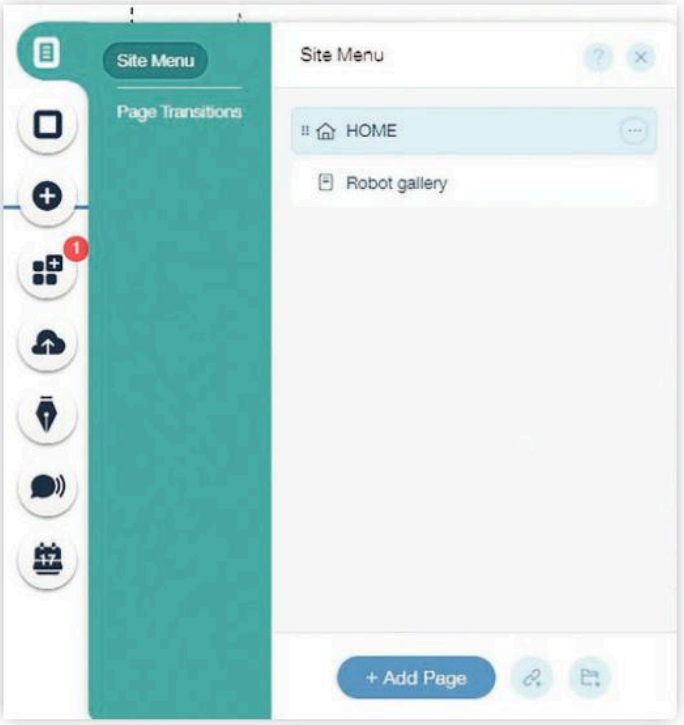


How to add a new web page

Here is how to add a new web page.

1. Choose the 'Menus and Pages' option from the tool bar on the left-hand side of the screen.
2. Click 'Add page' at the bottom of the screen.
3. Type in a name for your new page (our example is 'Robot gallery').

If you want to move between pages in the web page editor, choose the 'Menus and pages' option. Click on a page name to go to that page.



Your website

Your home page

You should see two web pages listed on the site menu – 'Home' and 'Robot gallery'.

A collection of web pages is called a website. You have now created a website! The first page you create is always called 'Home'. It is the home page. The home page is what people see first when they visit a website.

The page header

When you created the home page, you added the name of the web page and a logo in the page header.

When you create a new page, the information in the home page header is automatically inserted in the header of the new page.

How to link pages

When someone visits your website, they arrive at the home page. If you want your visitors to see 'Robot gallery', you must create a link to that page. Visitors click on the web link to visit your gallery. There are several ways to link from one page to another. In this lesson you will use a menu.

How to add a menu to a web page

1 Choose 'Add' from the tool bar on the left-hand side of the screen.

2 Select the 'Menu' option.

3 Select the menu style you want to add to your page.

4 A menu will appear on your web page. Drag it into the page header.

To see how your menu works, use the 'Preview' button. This button shows how your pages will look and work in a browser.

Activity

Create a new web page and give it a title. Add two or three images to the page. Add a heading to the body of the page.

Extra challenge

Add a menu to your home page. It should link your home page and your new web page.

Test that the menu works using 'Preview'.

Think again

Create a new page called 'Did you know?'. Research some additional facts about your chosen topic and add them to the page. Test the menu using 'Preview'.

2.6 Check your web pages

In this lesson

You will learn:

- how to review your web pages
- how to use a review to improve your web pages.

Reviewing your web pages

It is always important to review, or check, your work when you create a document. It does not create a good impression if there are mistakes in your work.

Checking your spelling

Your web page editor may have a **spellchecker**. The spellchecker will help you check the spelling and punctuation on your pages.

Check your text carefully even if your web editor has a spellchecker. Remember, the spellchecker won't find a word that you have spelt correctly but is not the right word to use in that sentence. For example, 'Robots never get bored by **there** work'. In this example, the correct word is 'their'.

Is your text clear?

Does your text say what you want it to say? It will help the reader if you keep your sentences short. Try to make sure that each sentence makes a single point.

Ask someone else to read your text. They can tell you if anything you have written is confusing.

Images

Here are some things to think about when you review your images.

- Are the images high quality?
- Are the images the right size and in the right place on the page?
- Do the images make the text on the web page clearer? Do they work with the text on the page?

If you have used images by other people, make sure that you give credit to the owners.

How does your page look?

It is important that your website should look good. A modern web page editor will help you to create a professional-looking page. Here are some things to think about when you review your page.

- Have you used fonts and text size consistently? Don't use too many different ones!
- Do the colours you have used go well together? Avoid using too many bright colours.
- Do the headings help you to understand what the page is about?
- Are items on your page too squashed together? Are they too far apart?

Ask someone else to look at your page and answer these questions. They may have ideas that will help you to improve how your pages look.

Do the links work?

Check the links and menus on your pages to make sure they work. It should be clear where links are on your page and where they go to.

Activity

Use the guidelines in this lesson to create a checklist for reviewing a web page. The checklist needs to list the checks you will carry out. For example, 'Check spelling and punctuation'. Write the checklist using a word-processor.

Check your list with a classmate to make sure you haven't missed anything. Use the checklist whenever you are reviewing web pages.

Extra challenge

Work with a partner. Review each other's web pages.

Write a brief review of your partner's website and tell them what changes you think they should make.

Use your partner's review to make the changes you think are needed.

Explore more

Use the checklist to review the home page you have created. Ask someone in your family to take a look, too. Can they suggest ways to make it better? Make any changes you have talked about.

Check what you know

You have learned

- how to act respectfully when using a computer
- how to create a simple web page with text and images
- how to review and improve your web pages.

Test

Think of a web page you have made. Now answer questions about what you did.

1. Describe something you added to the web page.
2. Describe one mistake you found on your web page and how you corrected it.
3. Describe the text and images you used to make the web page.
4. Explain one improvement you made to your web page. How did the change make it better?
5. Describe your choice of headings on your web page. Give reasons for your choices.
6. Explain what changes or improvements you would make to your web page if you had more time.

Activities

Everyone in the class has made a web page. Look at a web page made by someone in your class.

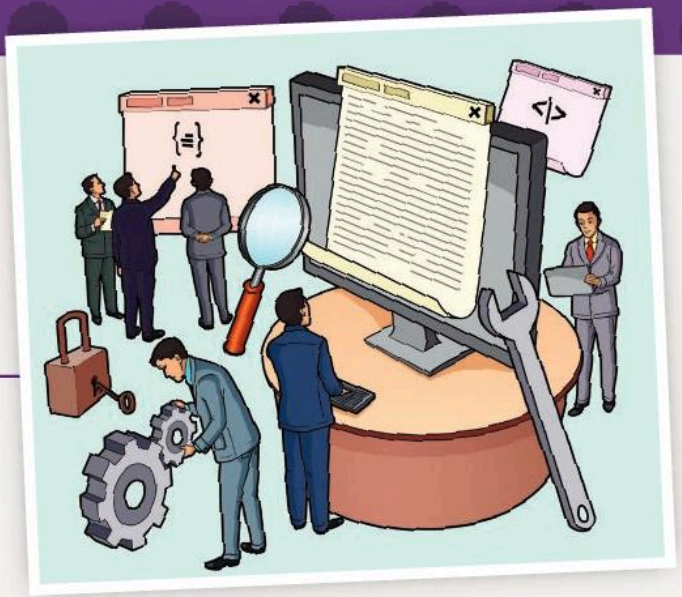
1. Use the web browser to look at the web page. Write a short description of what you see.
2. Write a short evaluation of the web page. Say something you liked about it, and something you would add or change if you could.
3. Use technology to send your feedback. For example, you might send an email or add a comment on the web page.

Self-evaluation

- I answered test questions 1 and 2.
- I completed activity 1. I worked politely when using the computer.
- I answered test questions 1–4.
- I completed activities 1 and 2. I worked responsibly online and with respect for others.
- I answered all the test questions.
- I completed all of the activities. I gave supportive feedback using online technology.

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: Algorithms and programs



You will learn

- how to make an algorithm to solve a problem
- how to find and fix errors in an algorithm
- how to adapt, improve and reuse algorithms
- how to turn an algorithm into a program.

In this unit you will make algorithms and programs to solve simple problems. An algorithm is a plan to solve a problem. It sets out the steps to solve the problem, in order. Programmers use algorithms to plan their work. The algorithm can be turned into a program.

Programmers often work very hard. They have to produce results quickly. It is also important to make programs that are free of errors and problems. Programmers work in ways that help them meet these challenges.

For example, they:

- reuse code they have made already
- adapt working programs to new uses
- work in teams and share program ideas.

You will learn more about these ways of working in this unit.

Talk about...

Almost all programmers work in teams to develop software. Many other jobs involve teamwork. Do you like working in teams? What are the best and worst things about working in teams?

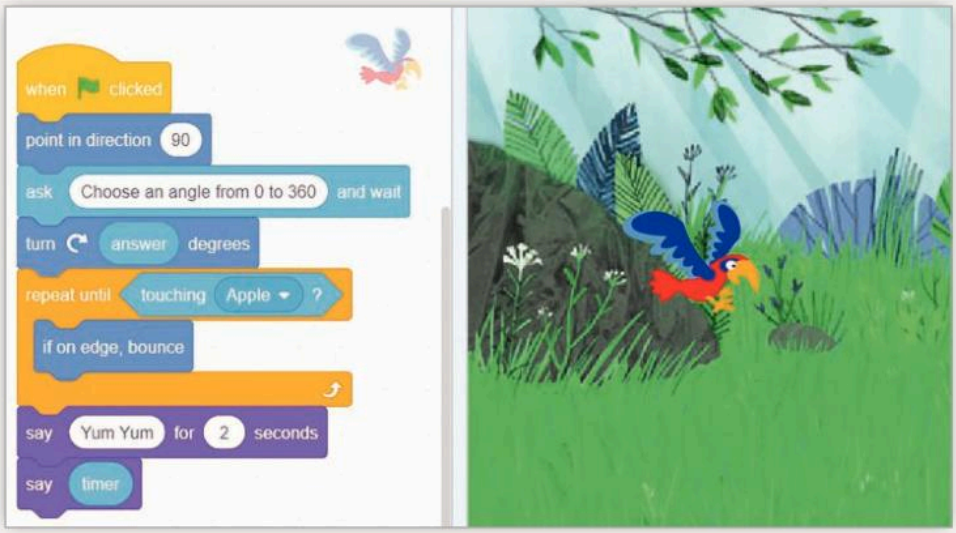
Learning outcomes: Make an algorithm to solve a problem using logical reasoning

trace requirement
algorithm reuse
repurpose average
counter loop

Class activity

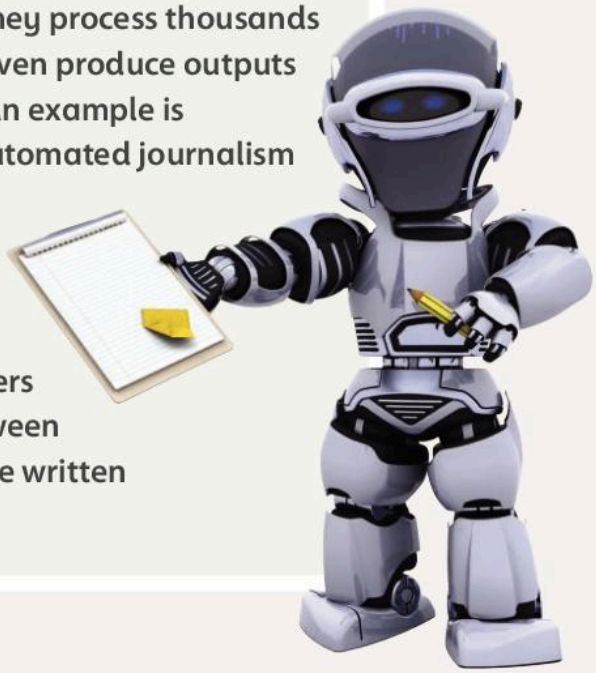
Work in a group or as a whole class. Have you used Scratch before? What programs have you made with Scratch? Make a list of all the Scratch programs that people in your class have made. If there is time, make posters or drawings to record your past work with Scratch.

If you have not used Scratch before, look at the Scratch website. Look at programs that other young learners have made. You can run the programs. You can also look at the program commands.



Did you know?

Some algorithms are very complex. They process thousands of inputs in complex ways. They can even produce outputs that seem to be written by a person. An example is algorithmic journalism (also called automated journalism or robot journalism). A computer algorithm will scan large amounts of data, pick out the key points and facts, and output a short newspaper article. Studies have shown that readers cannot always tell the difference between an article written by a person, and one written by an algorithm.



3.1 A simple plan

In this lesson

You will learn:

- how to simplify a problem
- how to plan a solution to a problem
- how to make a program to match the plan.

Spiral back

In this lesson you will plan a simple algorithm. Then you will make a Scratch program to match the algorithm. If you are not sure how to make an algorithm or a Scratch program, go back and review the work in Student Book 5.

Program requirement

When you plan and make a program, you must first know what the **program requirement** is. The requirement tells you what the program must do.

Here is an example of a requirement:

Calculate a total by adding ten numbers input by the user.

In this unit you will make a program that meets this requirement.

Start simple

Sometimes it can be difficult to know where to start with a problem. A good method is to start with a simple version of the problem. A programmer might make a program that solves the simple problem. Then they will add extra features.

That is the method you will use. Let's start by making a program that solves this simple problem:

*Calculate a total by adding **one number** input by the user.*

Plan the variables

The program must have variables. They will store the values.

- Scratch has a ready-made variable called 'answer'. It is one of the light blue 'Sensing' blocks. It stores the value input by the user.
- You will need one other variable. It will store the total. The total will start with the value 0.

Create an algorithm

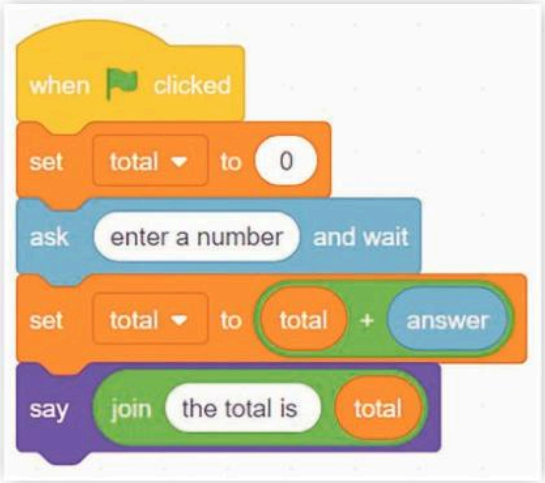
So, now you have made all of these decisions, you are ready to make an **algorithm** – the set of steps needed to solve a problem. The algorithm works as a plan when you make a program.

The algorithm looks like this:

```
Set total to 0
Input a number
Add number to total to give the new total
Output the total
```

Create a program

Now you will make a Scratch program to match this algorithm. You will need Scratch skills.



You must know how to:

- make a variable
- set the value of the variable
- use the 'ask' block to get input
- use operators to add and join values
- use the 'say' block for program output.

If you can't remember how to do these things, look back at Student Book 5.

The completed program is shown on the right.

Activity

Make the program shown on this page. Run the program and correct any errors. Save your work.

What colour are operator blocks? What two operators are used in this program?

Extra challenge

If you have time, adapt the program so that the user inputs two values. Each value is added to the total.

3.2 Extend the simple plan

In this lesson

You will learn:

- how to use a loop to extend the simple plan.

Program requirement

In the last lesson, you built a program to meet a simple requirement:

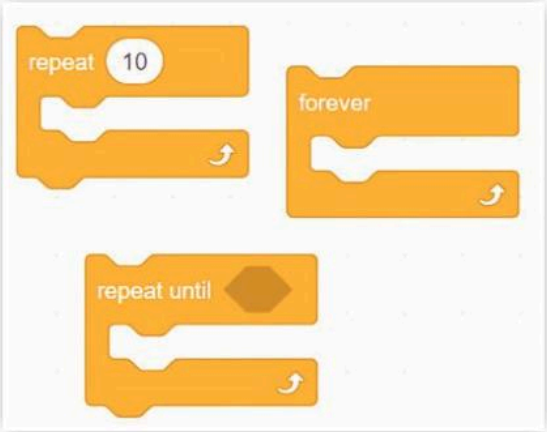
Calculate a total by adding one number input by the user.

Now you will extend it to meet the full requirement:

Calculate a total by adding ten numbers input by the user.

Loop

The program has to add a series of numbers. You want a program to carry out the same task over and over again. There is a program structure which is perfect for that task – a loop. Any command blocks inside a loop are repeated over and over again.



Several types of loop from Scratch are shown here.

You will use the type of loop that counts how many repeats there are. When this reaches the set limit, the loop will stop. This type of loop is called a **counter loop** or a **fixed loop**. Which block is the counter loop? Make sure you know.

Change the program plan

Which of the program commands go inside the loop? Which commands go outside the loop? If you think logically, you can work out the answer.

Command	If this command was repeated…	Decision
Set total to 0	The total would go back to zero every time.	Do not put this command inside the loop.
Input a number	The user would input several different numbers one after the other.	Do put this command inside the loop.
Add number to total…	Each number would be added to the total.	Do put this command inside the loop.
Output the total	The total would be displayed over and over again.	Do not put this command inside the loop.

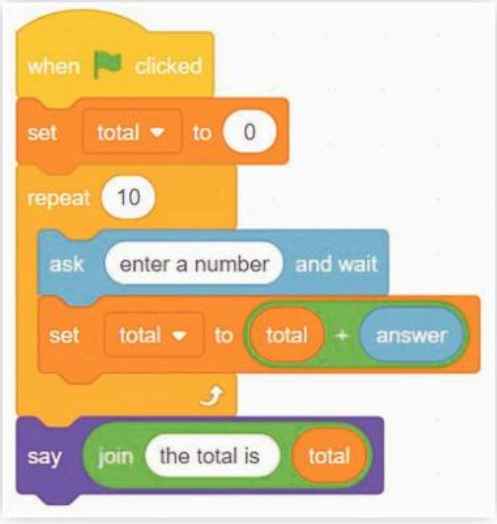
Revised algorithm

The new algorithm looks like this. The commands inside the loop are indented:

```
Set total to 0
Loop these commands 10 times
   Input a number
   Add number to total to give the new total
Output the total
```

Revised program

Here is the program that matches the revised algorithm. It includes a counter loop that counts to ten. You can count to any number you like. A smaller number will have fewer repetitions. This means the program will finish more quickly.



Activity

Make the program shown on this page with a loop. Run the program and correct any errors. Check the total, which is output. Is it the correct value, calculated from the numbers you input? Save your work.

Think again

You can make this program with a counter loop or a conditional loop. However, you couldn't use a 'forever' loop. Why not?

Extra challenge

Adapt the program so that it repeats until the total is greater than 50.

3.3 Check for errors

In this lesson

You will learn:

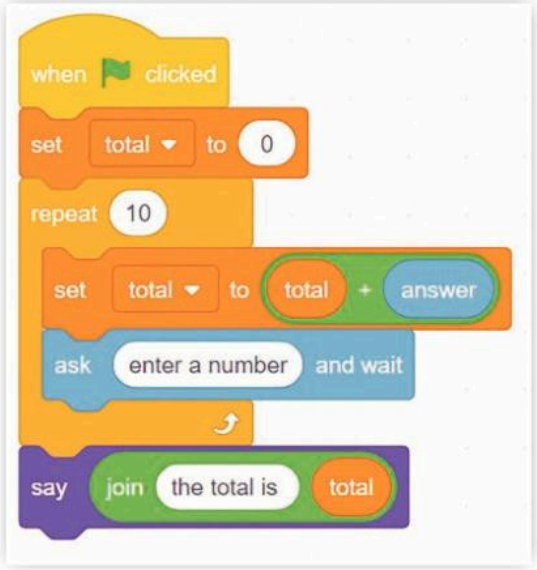
- how to check an algorithm for errors.

Deliberate mistake

Here is an algorithm with a deliberate mistake in it. Can you spot the error? What effect do you think this might have?

```
Set total to 0
Loop these commands 10 times
    Add number to total to give the new total
    Input answer
Output the total
```

Uncover the error



The image shows a Scratch program made from this algorithm. If you make and run this program, you will find that you do not get the correct result.

You can use logical reasoning to find the error in the algorithm. Think about each command and what it does. You must think about commands before the loop, inside the loop, and after the loop.

Before the loop

Before the loop, the total is set to 0. This is the right command. The total should be 0 at the start of the program, before any numbers have been added.

Inside the loop

Inside the loop, you will see commands for these two actions:

- Add the total to the answer to give the new total.
- Input the answer.

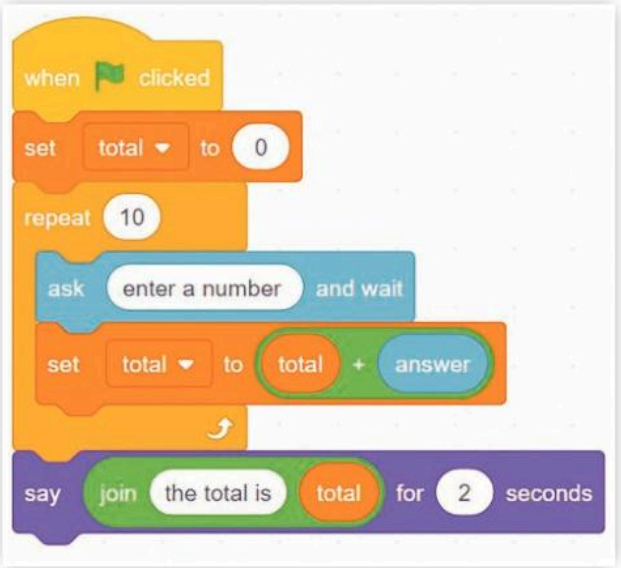
These commands are the wrong way round. The variable 'answer' is added before it is input.

After the loop

After the loop, the total is output. This is the right command.

Sequence

Using logical reasoning, we have shown that the error is inside the loop. The commands are the wrong way round. They are in the wrong sequence. **Sequence** means the order of commands.



- The first 'add' command will add nothing, because 'answer' has not been given a value.
- The final 'input' will not be added to the total.

Here is the algorithm with the commands in the correct sequence.

```
Set total to 0
Loop these commands 10 times
    Input number
    Add number to total to give the new total
Output the total
```

The image above shows a Scratch program based on this algorithm. This program works correctly.

Activity

1 Make the version of the algorithm with a deliberate error.

- Run the program.
- Enter the numbers from 1 to 10.
- What answer do you get? Is it the right answer?

2 Now make an algorithm with the problem fixed. Run the program and input the numbers from 1 to 10. What result do you get this time?

Extra challenge

The number at the top of the repeat loop sets the number of loops. What happens if you change this number to 0? See if you can work out the answer. Then try making this change to the algorithm and see if you were right.

Explore more

The first block in this algorithm resets the total to zero. What do you think would happen if this block were left out? Make a program with this error. Run the program a few times to see what happens. Explain what you found out.

3.4 Reuse and repurpose

In this lesson

You will learn:

- how to make changes to an algorithm so it meets a new requirement.

How to save time

Writing a computer program is a lot of work. So programmers reuse and repurpose their programs.

- **Reuse** means using the same program for a new task. For example, a programmer made a program to add up the bill in a restaurant. He reused the same code to add up the bill in a supermarket.
- **Repurpose** means making some changes to a program so it does a new thing. For example, a programmer made a program to add 17% sales tax to an invoice. She repurposed it for another business that wanted to add a 10% service charge to invoices. Objects can be repurposed too – in this picture, a plastic bottle has been repurposed as a growing container.

Advantages

Why do programmers reuse and repurpose? There are several advantages.

- **Saves time and work:** It takes less work and less time to make small changes to a program, compared to writing a whole new program.
- **Less risk of an error:** You have already checked the program for errors. You have used the program so you know it works. Repurposing the program is less risky than writing a new program.
- **Teamwork:** Programmers often work in teams. Sharing your programs with others is good teamwork.

New requirement

You have made a program that adds up a series of numbers to find the total. Now you will repurpose the program to work out the average.

Remember an **average** is worked out like this:

- calculate the total of a series of numbers
- divide by how many numbers there are.

How to change the program

You must adapt the program.

1 Make a new variable to store the average.

2 Calculate the average.

3 Output the average.

Make a new variable

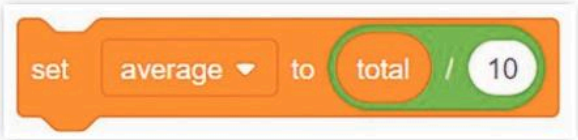
You know how to make a variable. Create a new variable for the program, with the name 'average'.

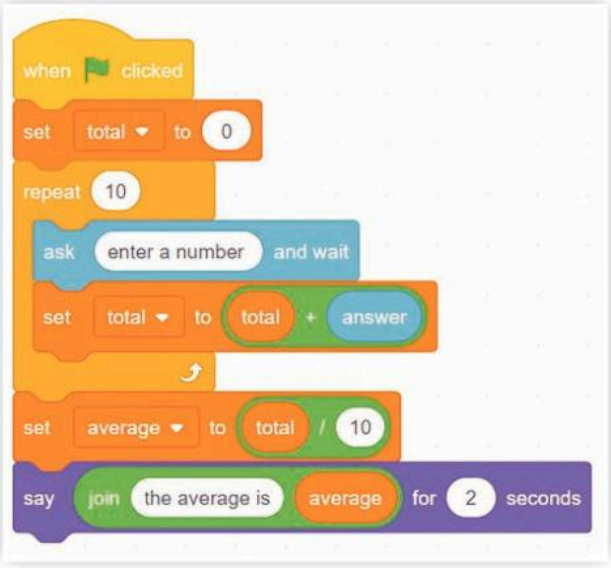
Calculate the average

You know that the average is calculated from the total divided by the number of inputs.

- The total is stored as a variable.
- The loop repeats 10 times, so the number of inputs is 10.

So the average is calculated as the total divided by 10. Here is the block that sets this value.





Output the average

Change the program so that instead of outputting the total, it outputs the average.

Activity

Make the program shown on this page. Run the program and correct any errors. Save your work using a new file name.

In your own words, explain the advantages to programmers of reusing and repurposing programs.

Extra challenge

This program lets the user input exactly 10 numbers. Adapt the program so the program asks the user how many numbers they want to input.

3.5 Turn an algorithm into a program

In this lesson

You will learn:

- how to turn an algorithm into a program.

Why use algorithms?

In this lesson you will be given a ready-made algorithm. You will see how a programmer turns an algorithm into a Scratch program.

There are lots of different programming languages as well as Scratch. An algorithm can be turned into a program in any programming language.

You have learned how programmers reuse and repurpose code. They can also reuse and repurpose algorithms. This saves time and makes it easier for programmers to share ideas.

How to turn an algorithm into a program

The algorithm

Here is the algorithm you will use:

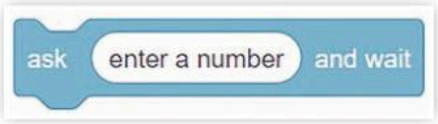
```
Set counter to 0
Loop until X is entered
   Input a value
   Add 1 to the counter
Output the counter
```

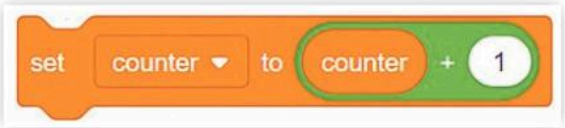
This program counts the number of values input by the user. The program stops if the user enters the value 'X'.

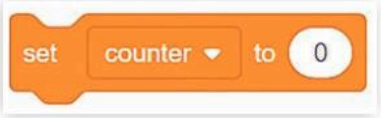
Match program blocks

These pictures show the program blocks you will need.

Match each block to one of the lines of the algorithm.







Build the program

If you have matched the blocks to the lines of the algorithm, you can now fit all the blocks together in the right order.

Activity

Make a program that counts the number of inputs. Use the blocks shown on the previous page.

Extra challenge

Adapt the program to add useful messages to the user.

- Tell the user that the program counts the number of inputs.
- Tell the user that they can stop the program by typing an 'X'.
- Alter the final message so that the sprite says 'Number of inputs' and the value of the counter.

Think again

This algorithm counts the number of values input by the user. That includes the final 'X' that stops the loop.

Imagine that the programmer did **not** want to include the input 'X' in the final count. How could you change the program to make this happen?

Digital citizen of the future

It is important that algorithms are designed to work fairly and treat all people equally. For example, a medical school used an algorithm to help decide who to admit for training to be a doctor. It was important that the algorithm was fair to all candidates. The inputs to the algorithm were factors such as exam results and voluntary work experience. The algorithm did not input factors such as family background or gender, which do not affect your suitability to be a doctor.

3.6 Combine and upgrade programs

In this lesson

You will learn:

- how to combine features from more than one program.

Combining algorithms

Programmers want to:

- save time
- reduce their workload
- reduce the risk of errors
- help others in their team.

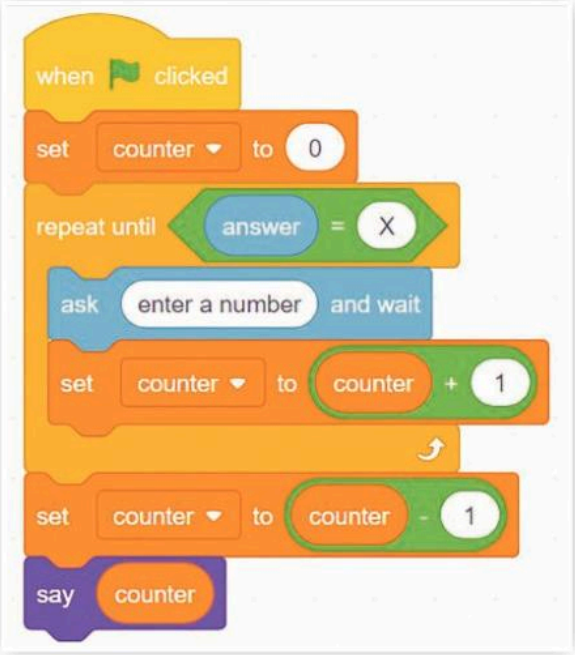
One way to do this is to 'borrow' commands and structures from programs they have already made. They can combine commands from more than one program to do a new task. In this lesson you will look at an example.

Programs you have made

In this unit you have made programs that:

- count the number of values entered using a conditional loop (Lesson 3.5)
- calculate an average using a counter loop (Lesson 3.4).

Now you will make a program that combines features from both of these programs. It will calculate an average using a conditional loop.



How to combine features from programs you have made

Program to count inputs

Load the program you made that counts the number of inputs.

You will see this version is adapted so that it does **not** count the final 'X' as one of the values. Make sure your program is adapted like this.

Add 'borrowed' commands

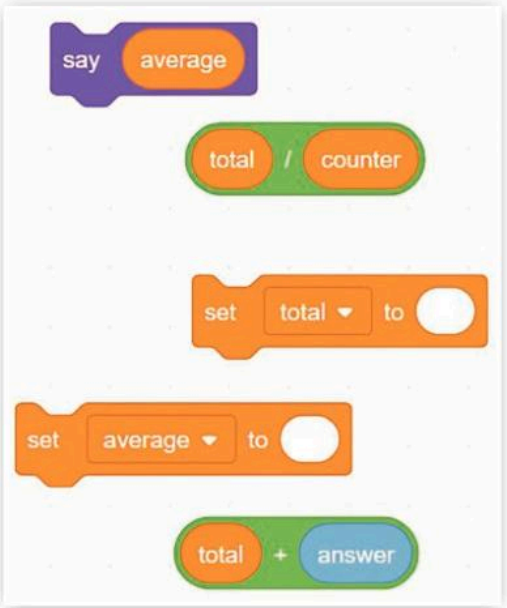
Now you will add some new commands to the program. You will 'borrow' commands you developed in your previous work. Here are the things you need to do.

- Create a variable called 'total' and a variable called 'average'.
- Set 'total' to 0.
- Add each input value to 'total' (inside the loop).
- At the end of the program, set 'average' to 'total' divided by 'counter'.
- Output the value of 'average'.

These are all things you have done before. Think back to remember how it is done.

The commands you will need

All the program blocks you need are shown on the right. Fit these blocks into the 'counter' program to complete the task.



The completed program

The finished program calculates an average using a conditional loop. It combines commands from the two programs that you have made before.

Activity

Make a program to calculate an average using a conditional loop. Run the program and correct any errors. Save your work.

Extra challenge

Adapt the program so that it outputs all three results:

- the number of values
- the total of all the values
- the average value.

Think again

Programmers 'borrow' code from programs they have already made. Say how this reduces the risk of errors.

Check what you know

You have learned

- how to make an algorithm to solve a problem
- how to find and fix errors in an algorithm
- how to adapt, improve and reuse algorithms
- how to turn an algorithm into a program.

Test

A student was asked to write a program. Here is the requirement:

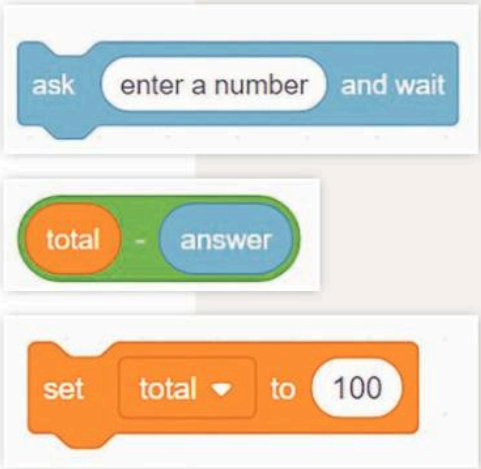
A total is set to 100. The user enters five numbers. Each number is subtracted from the total. At the end of the program the total is output.

The student decided to make a simplified version of the program first. Here is the algorithm they made:

```
Set total to 100
Subtract the input number from total
Ask user to input a number
Output total
```

This algorithm has an error in it.

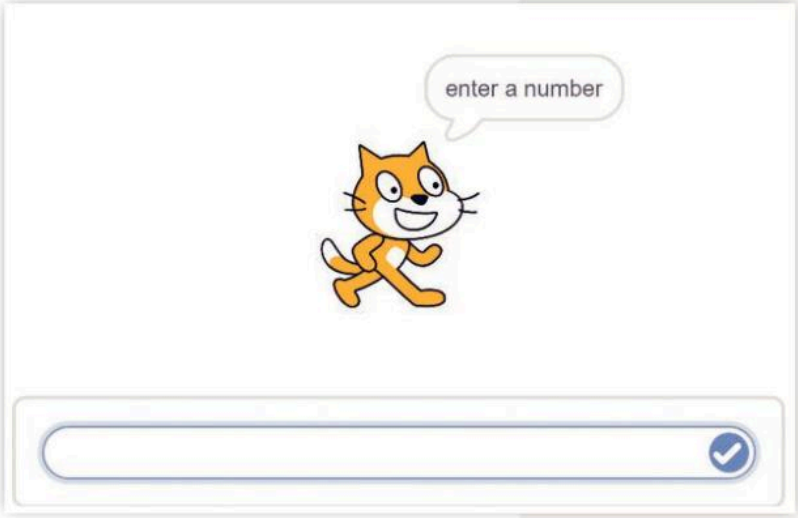
1. Write the algorithm without an error.
2. When the student turns the algorithm into a program, they will use these blocks. Say what each block does.



3. Make an algorithm that subtracts five numbers from a total.
4. Write an algorithm to subtract numbers from a total until the total is smaller than 0. Then output the total.

Activities

When you completed the test, you made an algorithm. Now you will turn the algorithm into a program.



1 Make a program to match the algorithm you made. If you made more than one algorithm, use the last one you made.

2 Extend or alter the program so that it subtracts numbers from the total until the user enters the number 0. Then the program outputs the final total.

3 Extend or alter the program so that it counts how many numbers have been subtracted and outputs this value at the end, as well as the final total.

Self-evaluation

- I answered test questions 1 and 2.
- I made a program that works.
- I answered test questions 1, 2 and 3.
- I did 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: The Frog Maze

You will learn

- how to make a program that controls the movement of objects on the screen
- how to use conditions and tests to make objects avoid obstacles
- how to store useful commands in modules
- how to assemble modules to create a working program.

In this unit you will make a simple computer game. A frog will go through a maze to get a birthday gift. It must avoid a snake which tries to catch it.

This unit introduces a new type of programming. You will make a **module**. A module stores several program commands. Using modules makes your programs shorter and easier to read.

Class activity

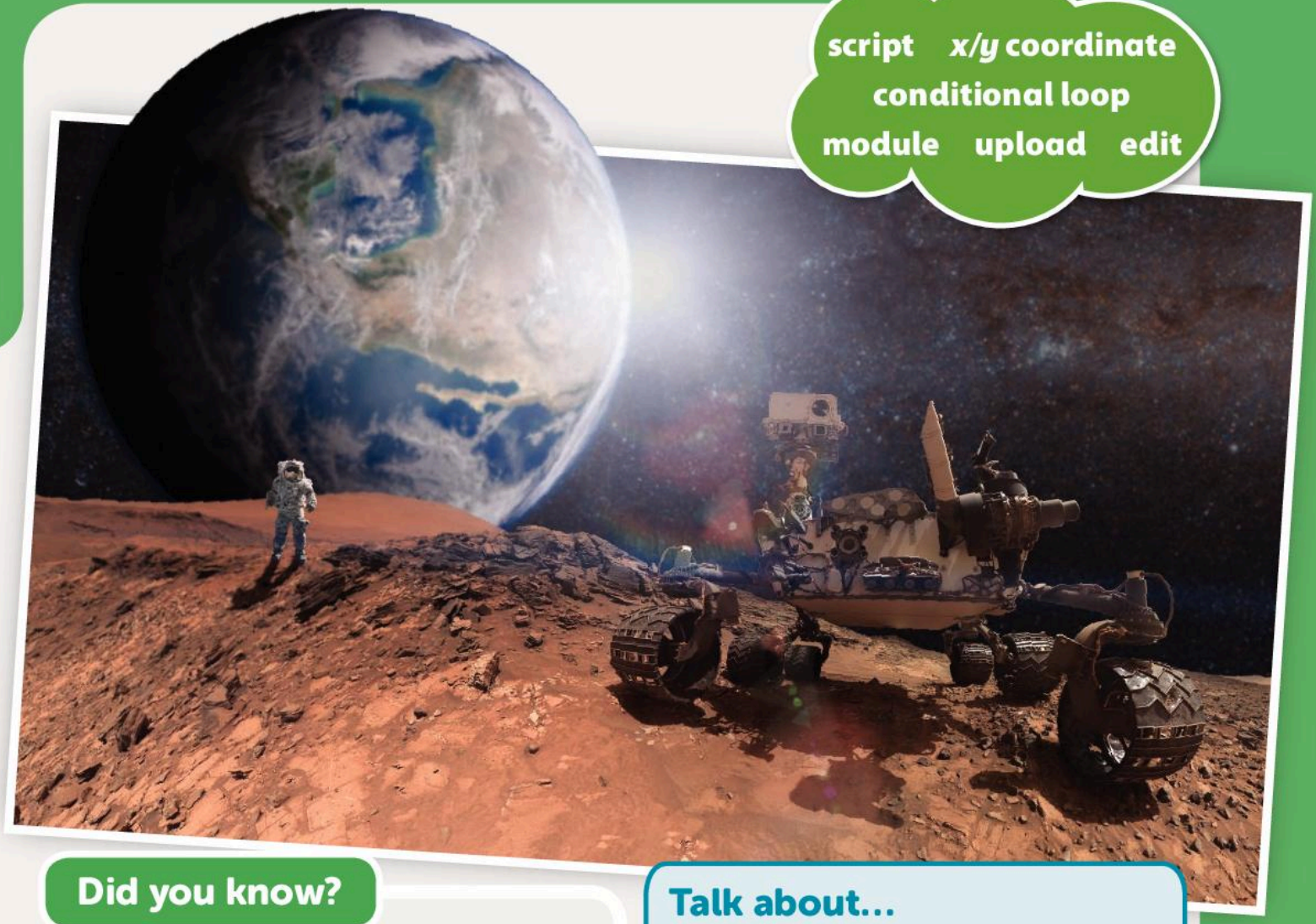
Work in pairs. Imagine that one of you is a robot. Write an algorithm with instructions that will guide the robot to walk from your classroom to the outdoor area of your school. Use your common sense to pick a suitable route and destination.

- One of you should read out the instructions in your algorithm. The 'robot' should follow the instructions. Remember, don't use your own judgement – just follow the algorithm.
- Review what happened. Did it work or do you need to improve the algorithm?
- Swap roles.

Write or report back to the rest of the class about your experiences.

Learning outcomes: Write a program that controls or simulates movement; Solve a problem by breaking it into smaller parts

script *x/y* coordinate
conditional loop
module upload edit



Did you know?

In this unit you will control the movement of objects on a computer screen. This is an activity that anyone can do if they have a computer.

However, Scratch also lets you write programs to control the movement of real objects; for example simple mechanical devices. For these features, Scratch has collaborated with LEGO® and with Micro:bit Educational Foundation.

To select and use these extra features, click on the 'Add Extension' button at the bottom left of the Scratch screen.

Talk about...

Robots controlled by programs can perform many of the actions of a person. Some people think that space exploration should be done by robots like this. There are good reasons. Space travel is dangerous for humans, and journeys to other planets can take a long time. Other people think that it is important for human explorers to make the journey.

What do you think? Would you volunteer for a space mission which might be away from Earth for years?

4.1 Frog movement

In this lesson

You will learn:

- how to control the movement of a sprite using *x*/*y* coordinates.

Spiral back

Last year, you created a game program where you steered a parrot to reach an apple. The position of each sprite was set using *x*/*y* coordinates. If you have forgotten what *x*/*y* coordinates are, look at Book 5. In this lesson you will use *x*/*y* coordinates to control the movement of a sprite.

Getting started

In this unit you will create a game where it is a frog's birthday. The frog tries to get his birthday gift. The picture below shows a suggestion for screen design for the first version of the game.

The example in this book uses two sprites – a frog and a gift. The backdrop is a night-time scene. You can choose any backdrop or sprites that you like.

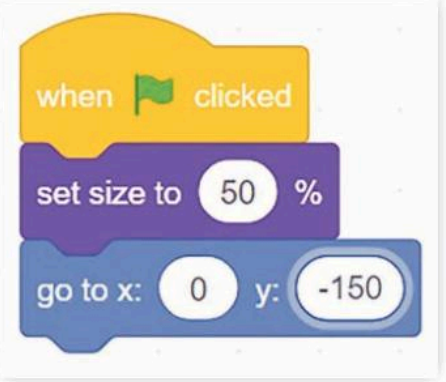
Starting values

Click on the frog sprites. Now you will write a short program which will set the starting position and the size of the frog. A **script** is a short program that controls one thing – for example a sprite. We will use 'program' to mean the whole game with several sprites, and 'script' to mean the code that controls one sprite.

The picture on this page shows the script for the frog. It will begin when the user presses the green flag. That will start the game.

How to get the frog to move

This script does not make the frog move. You will add commands so that the user can move the frog using the arrow keys. You learned in Unit 3 that it is good to start with a simple version. So you will make the frog move up the screen if the user presses the up arrow. Later you can add other commands so the other arrow keys work.



Change the *y* coordinate

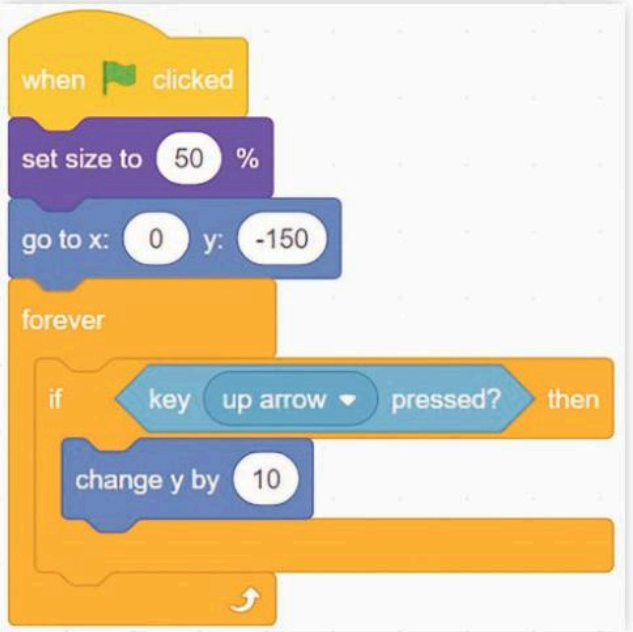
An ***x*/*y* coordinate** is a number value. It represents the position of a point on the screen. The *y* coordinate sets how far up or down the screen a point is. Points higher on the screen have a bigger *y* coordinate value. Increasing the *y* coordinate of the frog sprite will make it move up the screen.

These blocks will add 10 to the *y* coordinate if the user presses the up arrow.

Make this script by joining blocks together.

Forever loop

You want the frog to move not just once but over and over. So you need to put the movement blocks inside a forever loop. This picture shows the finished frog script.



The gift

The second sprite is a picture of a gift. You will make a script for this sprite. The gift does not move, but the script will set the starting values:

- Set size to 50%.
- Set the *x* value to 0. Set the *y* value to 150.

There is no picture of this script – see if you can make it yourself.

Activity

Choose sprites and a backdrop for your game.

- Make a starting script for the frog.
- Make a starting script for the gift.

Now run the program. If you press the up arrow, the frog will move on the screen. It moves towards the gift.

Digital citizen of the future

In this unit you will make a simple computer game. Games vary. Only play computer games that you enjoy. If a game isn't fun for you then you don't have to play it. Good friends respect those choices.

Think again

How would you adapt the script to make the frog move faster up the screen? Or to make it move more slowly?

Extra challenge

Extend the script so the frog moves down the screen if the user presses the down arrow.

4.2 Snake danger

In this lesson

You will learn:

- how to make the game respond to events
- how to use a conditional loop to control movement.

Add a snake

To make the game more exciting, you will add a danger. You will add a snake who can catch the frog and stop him reaching his birthday gift.

First, find and insert the snake sprite into the game. If you don't like snakes, you can choose something else! Remember, games should be fun.

Next, make a script to control the snake (or other 'danger'). It should include these commands:

- Start when the user presses the green flag.
- Set size to 50%.
- Set *x* and *y* coordinates to 0.

You can create this script without help, using the skills you have already learned.

Then add a 'forever' loop to make the snake move backwards and forwards across the screen. The 'rotation' block means the snake stays facing the right way.

If you play the game, you will see that the snake moves on the screen. The snake will keep moving forever (while the program is running).

Add a conditional loop

A **conditional loop** is a loop controlled by a logical test. The conditional loop block is shown on the right.

- Take out the forever loop, and use this type of loop instead.

There is a space at the top of the block. A logical test will fit inside this space. The commands inside the loop will repeat 'until' the logical test is True. In this case the snake will move until it 'catches' the frog. Then it will stop.

The snake catches the frog

If the snake touches the frog the loop will stop. You can use one of the light blue 'sense' blocks to check this. It is shown in the picture.

- Put the 'touching Frog' block into the conditional loop.

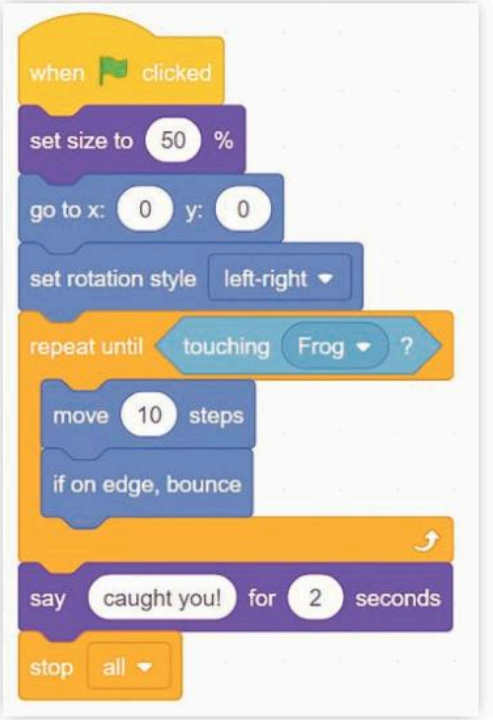
Now the snake will move until it touches the frog. If it touches the frog, the loop will stop.

What happens when the snake catches the frog?

Finally, you must decide what happens when the snake touches the frog. For example:

- the snake will say 'caught you!'
- the game will stop.

The completed script is shown on the right.



Activity

Add a snake to the game.

Write a script for the snake so that it moves left to right.

Use a conditional loop so that the game stops if the snake 'catches' the frog.

Extra challenge

Change the commands inside the loop so that the snake waits for two seconds, then jumps to a random position on the screen. Does this make the game easier or harder? For the rest of this unit, you may use either snake script.

Think again

The loop in this program starts with a test. What is the test? What happens if the test is True?

4.3 Birthday celebration

In this lesson

You will learn:

- how to use program commands to control visual and sound outputs.

Touch the gift

You will change the game so that if the frog touches the gift he 'wins'.

In the last lesson, you used a conditional loop. A conditional loop is controlled by a test. In the last lesson, the test was True if the snake touched the frog.

Now you will change the frog script. At the moment, the frog moves 'forever'. The frog script has a 'forever' loop. You will change this to a conditional loop. The loop will stop if the frog touches the gift.

Change the 'forever' loop to this conditional loop. You will have to pull the frog script apart, change the loop, and then fit it all back together again.

What happens if the frog touches the gift?

If the frog touches the gift, the game will stop.

Add these commands at the end of the script:

- The frog says "Happy birthday to me".
- The game stops.

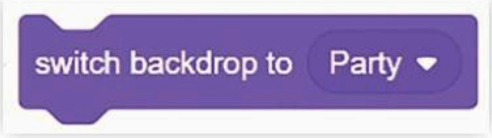
The command blocks should come after the conditional loop.

Change the backdrop

If the frog touches the gift, the backdrop will change. You should change your backdrop into a party picture.

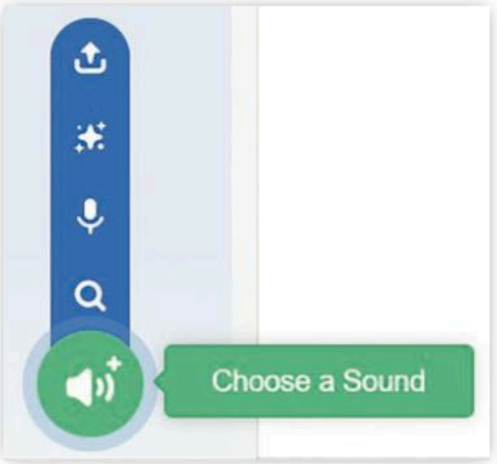
- Click on the 'Choose a Backdrop' button.
- Find the backdrop called 'Party'. Click to select it.

There are now two different backdrops in the frog game. This command block lets you set a backdrop, as shown on the right.



Add this block to the frog script. Add the block twice.

- At the start of the script, select the backdrop you want to start with (for example, 'Woods').
- At the end of the script, select the backdrop 'Party'.



Choose a sound

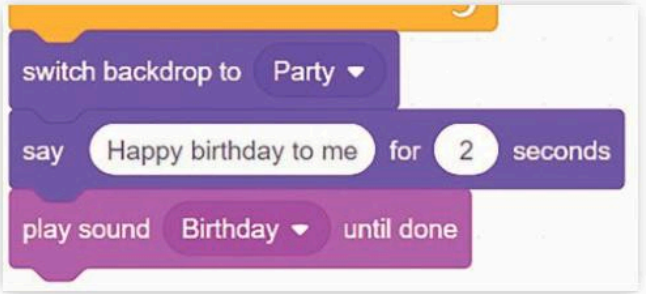
Now you will add a happy sound to the game. Across the top of the screen are three tabs that say 'Code', 'Costume' and 'Sound'. Click on the 'Sound' tab.

At the bottom of the screen, find a button like the one shown in the picture.

Click on 'Choose a Sound'. You will see a wide range of sounds to choose from. We liked one called 'Birthday', but you can pick any sound.

Complete the program

When the frog reaches the gift, three things must happen: the backdrop changes to a party, the frog says 'Happy birthday to me' and a happy sound plays. The blocks to make this happen are shown on the right.



Make sure these blocks come at the end of the frog script, after the loop has finished.

Activity

Change the frog script so that if the frog touches the gift:

- the game stops and the frog says 'Happy birthday to me'
- the 'Party' backdrop appears
- the program plays a party sound or a birthday tune.

Extra challenge

Change the snake script to play a gloomy or scary sound if the snake touches the frog. You may also change the backdrop.

Think again

A student put the 'play sound' block inside the loop by mistake. What will happen when he plays the game?

4.4 Movement module

In this lesson

You will learn:

- how to store commands in modules.

Press the up arrow

The position of the frog on the screen is set by *x*/*y* coordinates. When you press the up arrow, the value of the *y* coordinate increases by 10. This makes the frog move up the screen. The command is shown here.

You already made this command – it is inside the conditional loop.

The command includes a logical test. The logical test checks if the user has pressed the up arrow. The drop-down menu will let you change this to a different key.

Other arrows

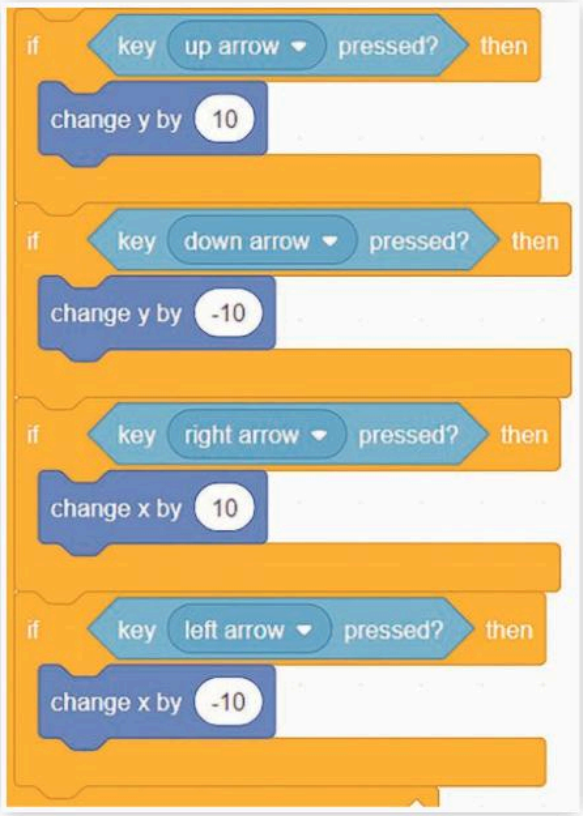
You can make other versions of this command. As the table shows, different keys will have different effects.

Key press	Change
Up arrow	Change *y* coordinate by 10
Down arrow	Change *y* coordinate by –10
Right arrow	Change *x* coordinate by 10
Left arrow	Change *x* coordinate by –10

The up and down arrows change the ***y* coordinate**. The left and right arrows change the ***x* coordinate**.

Create four different versions of the command and fit them together as shown here.

Put these blocks inside the loop of the frog script. Now the frog will move in any direction, controlled by the four arrows.



Create a module

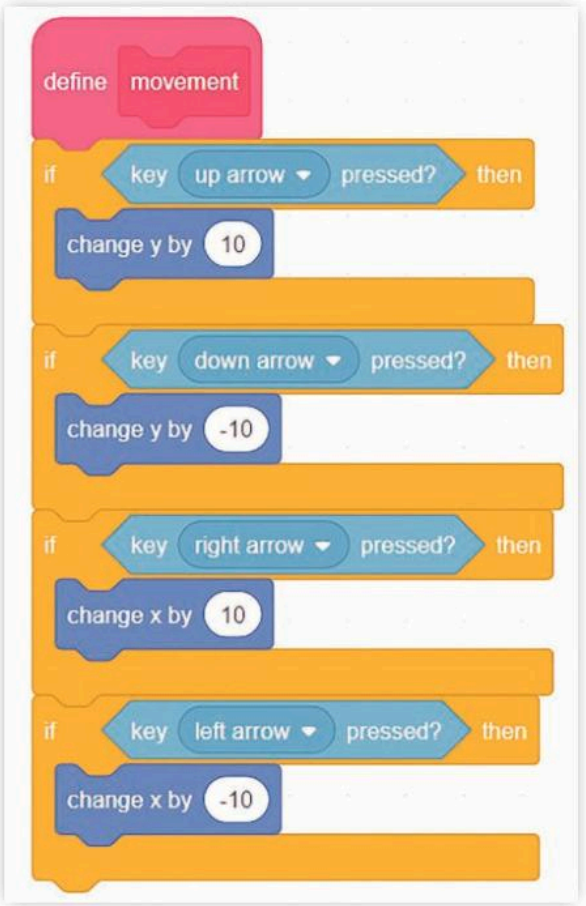
The four movement blocks you made take up a lot of space. The frog script is now very big. To make the script smaller and easier to read, you will make a module. A module stores commands. This module will store all the movement commands.

In Scratch, a module is a new block. Now you will make this block.

Click on the red 'My Blocks' dot. Then click the button that says 'Make a Block'.

Now you can make a new block. This block will store the movement commands. A good name for this block is 'movement'. Type the name and click OK.

Choose module names carefully. Choose a name that reminds you what commands are stored in the module.



Add commands to the module

A red block should appear in the script area. It is the new block you made. Now you can store the 'movement' blocks in this module. Drag all the movement blocks out of the frog script and join them to the new block you made.

Now you have made a module. In the next lesson you will see how to use the module in the program.

Activity

Make the movement module shown in this lesson. Your program will not run properly until you complete the next lesson in this unit. Save your work ready for next time.

Think again

Two students did this lesson. One called their new module 'arrowmove' and one called it 'mymodule'. Which is a better name? Can you say why?

Extra challenge

Make a module called 'startup'. Put the start commands from the frog script into this module. Those are the commands that set the size, backdrop and position of the frog.

4.5 Using the new module

In this lesson

You will learn:

- how to use modules to simplify a program.

How modules work

In the last lesson, you made a module. A module stores a set of commands. The module has a name chosen by the programmer.

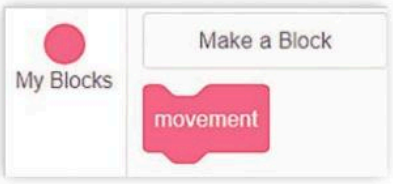
A programmer can use a module in their programs. When the computer sees the name of a module in the program, it will carry out all the commands stored in the module.

Advantages of modules

Most programming languages use modules. In some programming languages, modules are called procedures or functions. Storing commands in modules has many advantages for programmers:

- It makes programs shorter and easier to read.
- Programmers can use a module in more than one part of their program. That saves time and effort.
- Programmers can use modules in new programs. They can share them with other programmers. That makes everyone's work easier.

Modules in Scratch



A module is like a mini-program. It stores a series of commands. The module you made in the last lesson stores all the movement commands.

In Scratch, modules are called 'My Blocks'. Click the red dot that says 'My Blocks'. You will see one block available to use. It is called 'movement'. This is the block you made in the last lesson.

You have made a new Scratch block! You can use this block just like any other Scratch block. What will this block do? It will do all the commands that you put into the module.

Use the module in your program

In the last lesson, you took all the commands out of the forever loop. You put them in the 'movement' module. The red 'movement' block stands for all these commands.

Drag the red 'movement' block into the frog script. Now the script looks like this picture.

When you run the game, the frog will move. All the arrow keys will work. All the different commands that you made will work. That single red block stores all the commands.

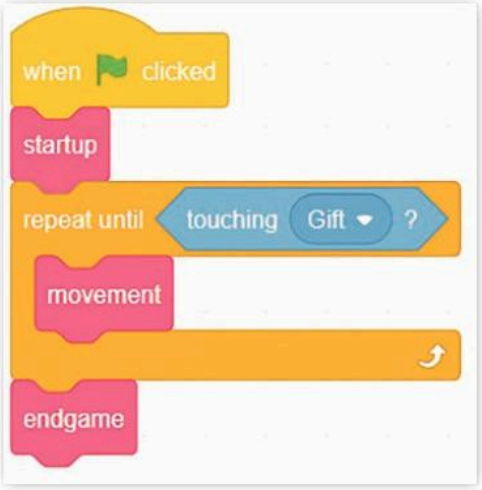
Make more modules

If you have time, make and use more modules in your games program.

For example, make a module called:

- 'startup' to store all the starting commands that come before the loop in this module
- 'endgame' to store all the commands from the end of the game.

Use the modules you make in the frog script. That should make it short and easy to read.



Activity

Use the 'movement' module you made to simplify the frog script.

Explore more

Make a poster or leaflet explaining what modules are and why they are used in programming. Use the ideas set out in this lesson. You can make the poster using word-processing software. Then you will be able to include screenshots of your code as illustrations.

Extra challenge

Make 'startup' and 'endgame' modules and use them in your program.

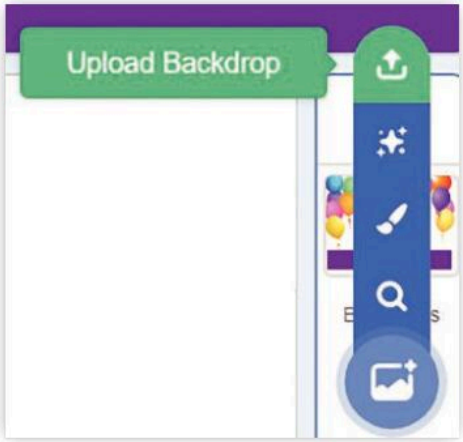
4.6 Adding the maze

In this lesson

You will learn:

- how to make an object avoid obstacles
- how to use your most advanced skills to make a challenging game.

Making the maze game



In this lesson you will make a more complex version of the frog game using many programming skills. Make sure you have completed all other work first. Do as much of this activity as you can.

New backdrop

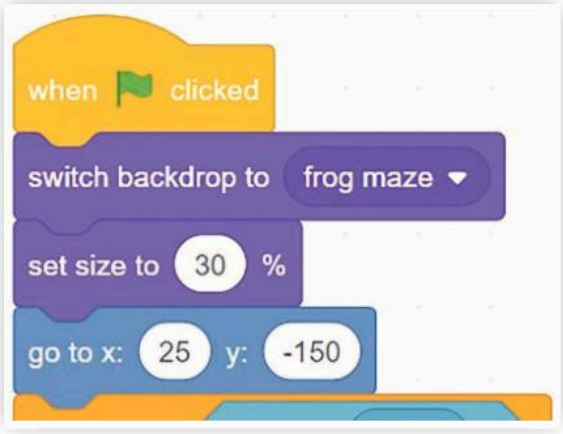
You will replace the backdrop of the game with a new picture. It will show a maze. The frog has to go through the maze to reach the gift.

We have provided a picture of a maze that you can use. Or you can make your own. See the Be creative section for some ideas.

Open the 'Backdrop' menu and choose the top option: 'Upload Backdrop'. To **upload** means to copy a file from your computer to the Scratch website.

Select the maze image. This image is now the backdrop to your game.

Edit the scripts



Edit means make changes to a script or other file. You must edit the scripts you have made, so they work with the maze.

Frog script

Change the frog script to load the maze backdrop and put the frog in the right place. The values in this image match the maze we provided. If you use a different maze, you may need different values.

Note: If you made a startup module, you will need to edit that instead.

Movement module

Change all the movement numbers from 10 to 2 so the frog goes more slowly. This will make it easier to control.

Gift script

Set the *x/y* coordinates of the gift to –25 and 150. That will put it at the end of the maze. If you use a different maze you may need different values.

Sense if the frog touches the maze wall

You will change the frog script so the frog can't go through the walls of the maze. Look at the light blue 'Sensing' blocks. Find the block which senses if the frog touches a colour. Click on the colour.

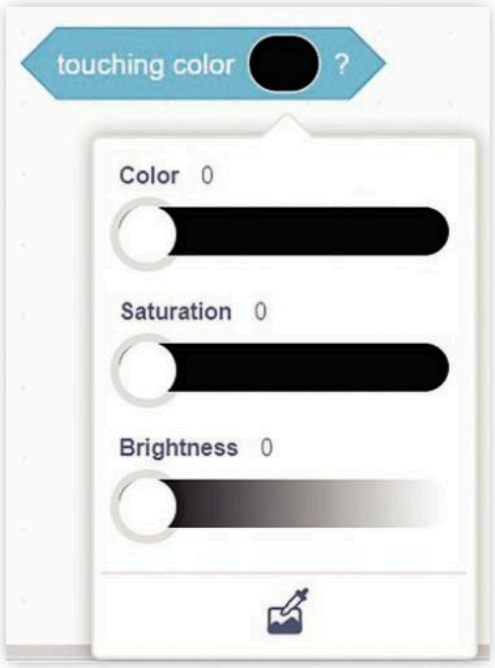
Set all the colour values to 0. Now the test will be True if the frog touches the black walls of the maze.

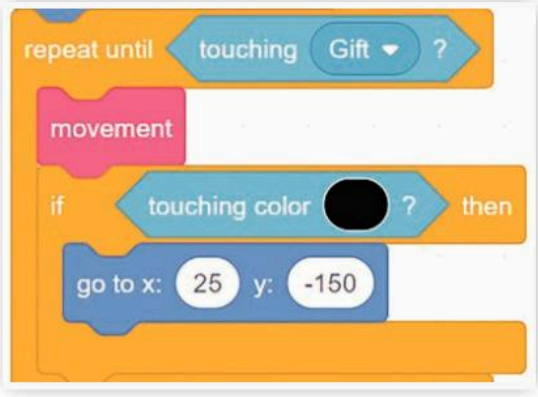
Go back to the start

Put the 'Sensing' block into an 'if' structure. If the frog touches the wall then it must go back to the start of the maze. The frog started with *x/y* values of 25, –150. Use a 'go to' block to send the frog back to that place.

The final picture shows the completed 'if' structure. Put the structure inside the frog script.

Now you can play the game.





Activity

Make the Frog Maze game as shown in this lesson.

Extra challenge

Change the Frog Maze game so that if the frog touches the maze wall, it also makes a sound.

Be creative

There are many ways to get a maze picture. Find a picture on the internet. Make a picture using graphics software. Or draw or paint a maze, and scan the page to make a computer file.

Explore more

Develop a new idea for a computer game. You can use ideas from the Frog Maze game, such as avoiding dangers. Write a full description of it.

Check what you know

You have learned

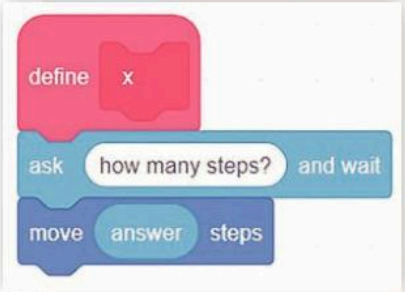
- how to make a program that controls the movement of objects on the screen
- how to use conditions and tests to make objects avoid obstacles
- how to store useful commands in modules
- how to assemble modules you have made to create a working program.

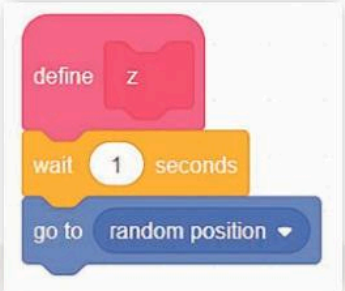
Test

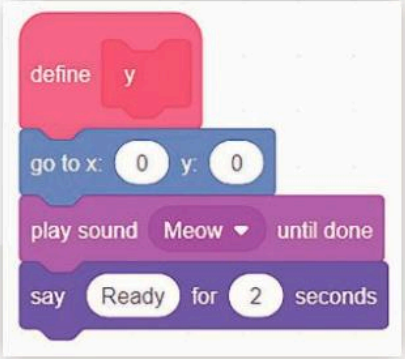
A student wants to make a game where a spaceship flies about in space. A 'hyperjump' feature will make the spaceship jump to a new space location.

Here are three different modules. Each module stores different commands. The modules are called x, y and z.

1. The student wants to make the spaceship jump to a random place on the screen. Which module should she use?
2. The student wants to make the spaceship jump to a random place when the user presses the space bar. What would the script look like? Either describe the script in words or say what blocks you would use.
3. Look again at modules x, y and z. Write a description of what each module does.
4. Suggest better names for each module.







Activities

This game is called Dodge the Starfish. The crab swims in the sea. If it touches the starfish, the game is over.

1. Choose a backdrop and two sprites. Make a script for the starfish so that it moves on its own around the screen. Use a forever loop.
2. Make a script for the crab. The user controls how the crab moves. You can choose how to control the crab – using the keyboard or the mouse. Put the movement commands inside a conditional loop so the crab moves until it touches the starfish.
3. Make a module to store actions that will happen after the crab touches the starfish. For example, the backdrop might change, the crab might jump to a random location, or there might be a sound. Give the module a suitable name. Put the module block into the crab script in the right place.

Self-evaluation

- I answered test question 1.
- I completed activity 1. I made the starfish move.
- I answered test questions 1 and 2.
- I completed activities 1 and 2. I made both sprites move.
- I answered all the test questions and completed all the activities.
- I completed all the activities. I made and used a module.

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

5 Multimedia: Our school survey

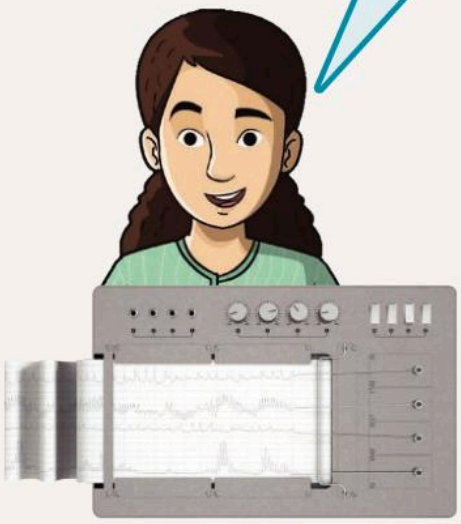
You will learn

- how to plan a survey and data collection
- how to use digital devices and applications to collect data
- how to use software applications to analyse data
- how to use software applications to create and deliver a presentation of your survey results.

In this unit you will plan and carry out a survey in your school. You will use a digital device to collect data. You will use a spreadsheet to store and analyse the data. You will use presentation software to share your survey results and analysis.

Talk about...

In this unit you will investigate the noisiest and busiest part of your school. What parts of your school are noisy and busy? What times of day are most busy? Can you agree on a single time and place to investigate?



Collecting data with analogue and digital devices

People have designed and used machines to collect data for many years.

Chart recorders are **analogue** devices that were used to record different values (sometimes called parameters) on rolls of paper.

A polygraph machine is a kind of chart recorder. A polygraph is sometimes used as a lie detector, as shown in the image. Each line on the graph paper shows the measurement of one parameter. Polygraph machines also record parameters for breathing, sweating and blood pressure.

Nowadays, analogue chart recorders are often replaced by digital data loggers. The most famous kind of data logger is a flight **data recorder** (FDR). A FDR is fitted in most aircraft. It is a multi-channel recorder. It records at least 88 different parameters during a flight.

Learning outcomes: Use technology to collect or record data; Work as a team using technology (for example to give a group presentation)

quantitative data
qualitative data
data logger transcribe
slide transition
audio file handout

These include:

- air speed
- altitude
- engine power
- position of control levers and switches in the cockpit.

The data can be used for servicing and investigating incidents.

A FDR is often called a 'black box', but as the image shows it is painted orange, so it can be more easily found in an accident.

Data logging with your own device

Many people have a device that they use as a data logger: a smartphone or tablet.

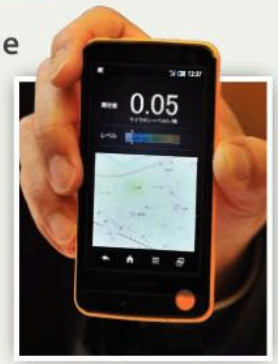
Smartphones have built-in **sensors** that record data all the time. These include:

- GPS sensor to sense the device's location by measuring the distance to satellites in space
- accelerometer and gyroscope to sense the device's position and direction of movement
- microphone to sense sound
- camera and ambient light sensor to sense light
- proximity sensor to sense objects nearby
- magnetometer to sense the phone's direction, such as a compass.

Class activity

Above this activity is a list of six sensors you might find in a typical phone. You will investigate the noisiest and busiest part of your school and interview people to see what they think. Which of the phone sensors might help you? How might you use them? If you know of any other sensors and apps that might be helpful then note those down.

Did you know?



The Sharp Pantone smartphone has a Geiger counter to sense background radiation. The phone was released at a time when many people in Japan wanted to measure radiation for health reasons. The coloured button switches on the radiation sensor.

5.1 Plan a survey

In this lesson

You will learn:

- how to plan a survey
- how to identify the kind of data you want to collect
- how to choose technology to help you collect data.

Spiral back

In Student Book 3, you learned how to use a presentation application. Last year, you learned about data and graphs. In this unit you will combine your skills to record, analyse and present information to an audience.

Plan a survey

Before you can plan your survey, you must decide what you want to find out. This is called your survey goal. You should be able to write down your survey goal in one or two sentences.

The survey goal in this unit is:

Find out the busiest and noisiest areas of our school at different times of the day. Find out what people feel about the space and noise at these times.

Choosing the data you need to collect

When you have decided on your survey goal, choose what data you need to collect to meet the goal. There are two types of data.

- **Quantitative data** is numerical. It is used to measure and compare. In the example, the quantitative data is numbers of people and noise levels.
- **Qualitative data** is descriptive. In the example, qualitative data will describe how classmates feel about the noise levels in school.

Choosing a survey method

Once you know what data you need, you can choose the way to collect it. This is called your survey method. Survey methods include:

- questionnaires, which can be on paper or online – they are used to collect quantitative and qualitative data
- interviews – they are used for qualitative data.
- measuring devices (digital or analogue) – they are used to collect quantitative data.

In the unit survey, digital measuring devices and interviews will be used to collect data.

Choosing technology for a survey

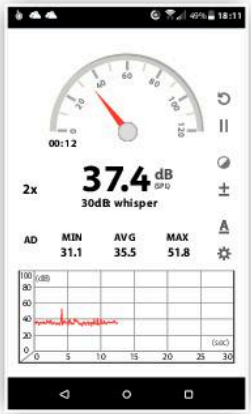
To make sure you choose the right technology, think about devices that can record the data you need. Think about specialist devices. Think about how you can use devices such as smartphones and tablets.

In this unit, technology will be used in three different ways:

Data to be collected	Technology
Number of people moving around the school	A smartphone counter app and a stopwatch or timer app
Level of noise in the places where you are counting people	A smartphone sound meter app
People's thoughts about the level of noise	A smartphone voice recorder app

This is a smartphone counter app. The app helps you quickly count people or objects, for example cars.

This is a smartphone sound meter app. It measures and displays sound levels using the phone's microphone.



Activity

Plan your survey.

- Write down your survey goal. Make sure you include information about location and times.
- Write down the data you will need to collect for your survey. Find quantitative and qualitative data types.
- Investigate the technology you can use to collect data. Are there any apps that can help you?

Think again

Surveys create a lot of data. Think about how you can record this data so that you can use it to meet the survey goal.

Extra challenge

How will you carry out your survey? Write down some ideas on how you could prepare for the survey and record your data.

5.2 Prepare and carry out a survey

In this lesson

You will learn:

- how to prepare for a data collection survey
- how to record data for analysis
- how to carry out a survey using digital data collection tools.

Prepare for a survey

To prepare for your survey, you need to form a team and agree your roles. This survey involves using digital data collection tools.

Agreeing team roles

As the table shows, in this survey there are four roles in each survey team.

Team role	Duties
Surveyor 1	Use the counter app to count the number of people at the survey location.
Surveyor 2	Use the sound meter app to record the sound level at the survey location.
Time keeper	Use the stopwatch to make sure the sound and data collection start and end at the same time.
Interviewer	Use the voice recorder app to record people's answers to a survey question.

You also need to agree how to record your data for analysis later.

Record data for analysis

Quantitative data

In this survey, there are two quantitative data items:

- the number of people
- the noise level.

You need to agree a sample time for this data. If you collect data at different times or in different places, you need to use the same sample time so that you can compare the results. In the example in this unit, a sample time of one or two minutes would work.

You will use a spreadsheet to record your data.

First, decide on the headings you will need.

Create your own template, on paper, to record data. You can transfer the data to the spreadsheet later.

My school survey

Time	Place	Number of people	Noise level recorded		
			Minimum	Maximum	Average
09:30	Corridor	21	37db	62db	58dB
12:30	Corridor	76	67db	83db	74db

Qualitative data

The qualitative data is the recorded answers to a survey question. You can use the voice recorder app on a smartphone or tablet. Ask each interviewee for permission before you record their answer.

Write down the question before you start. Your question could be:

How do you feel about the noise level in this location?

This is called an open question. This means people can't answer with just 'yes' or 'no'. The answers should tell you more about how people feel.

Activity

Prepare your survey.

- Form a team. Agree the roles you will each take. Agree where you will survey and at what time(s).
- Create a paper template to record the quantitative data during the survey.

Extra challenge

Add an extra question to your interview. Make sure it is an open question.

Think again

Every group in your class will carry out a survey. You might all work at the same time on different days. Why is it important that everyone does the survey in the same place, and at the same time of day?

Activity

Carry out your survey.

- Use your chosen devices to record your data.
- Take your sample of data: measure the sound level for two minutes and count the number of people passing the surveyor at the same time.
- Record interview answers with at least three people.
- Record your data on your template.

5.3 Analyse your data

In this lesson

You will learn:

- how to transfer your data to a spreadsheet
- how to create a graph that helps your audience understand the data.

In the last lesson you carried out your survey. This is called **fieldwork**. You collected your data. In this lesson you will analyse your data.

Present your data so you can analyse it

The first step is to copy your data into a format that helps you analyse it.

Transferring quantitative data

In this lesson, you will transfer your survey data into a spreadsheet. The example in this lesson uses a formatted data table that helps you make charts.

Download the spreadsheet 'Survey data'.

3 To add a new graph, select all the data in the table. Click on 'Recommended Charts' on the 'Insert' menu. Choose a chart from the list.

2 The graph will change when you enter a new value.

1 Copy the data from your paper template.

Transcribing qualitative data

Copying qualitative data after a survey is called transcribing.

Data accuracy is very important when you **transcribe** data. Your data should tell the truth, so it needs to be correct. You can type the words into a table or write them.

Download the document 'Interview transcription'.

My interview transcription

Time	Place	Answer
09:00	Corridor	I don't mind it being a little bit loud in the corridor because I like to say hello to my friends before classes start.
12:30	Corridor	Sometimes it gets very noisy at lunch time. Everyone is excited and in a rush. I like it a little quieter when I want to study.

Listen to your voice recordings. Write or type out the interview answers.

Add the place where the recording was made.

The time of the recording is usually in the display when playing the audio file. Adding the time will help you match the data with the quantitative data your team collected.

Analysing your data

Use the spreadsheet's data table and charts to analyse the data. This should help you answer questions, such as:

- When is our school busiest?
- When is it noisiest?
- Is our school always noisier when it is busier?

Use your transcribed interview answers to answer questions, such as:

- Are people more happy or less happy when it is noisy and busy?
- What other feelings do people have about the noise levels?

Activity

Transfer your quantitative data using the instructions above.

Transcribe your qualitative data.

Review your data with your survey team. Write down your conclusions.

Extra challenge

In the next lesson, you will start to make a presentation.

Think about the survey conclusions you have reached with your team. Put your conclusions in the order you think they should be presented. Ask your team if they agree with your plan.

Think again

Can you think of more data you could add to the transcript template? How could the data help you analyse your survey?

5.4 Create your presentation

In this lesson

You will learn:

- how to create a slideshow for your presentation
- how to add slides and text
- how to add images to your slides.

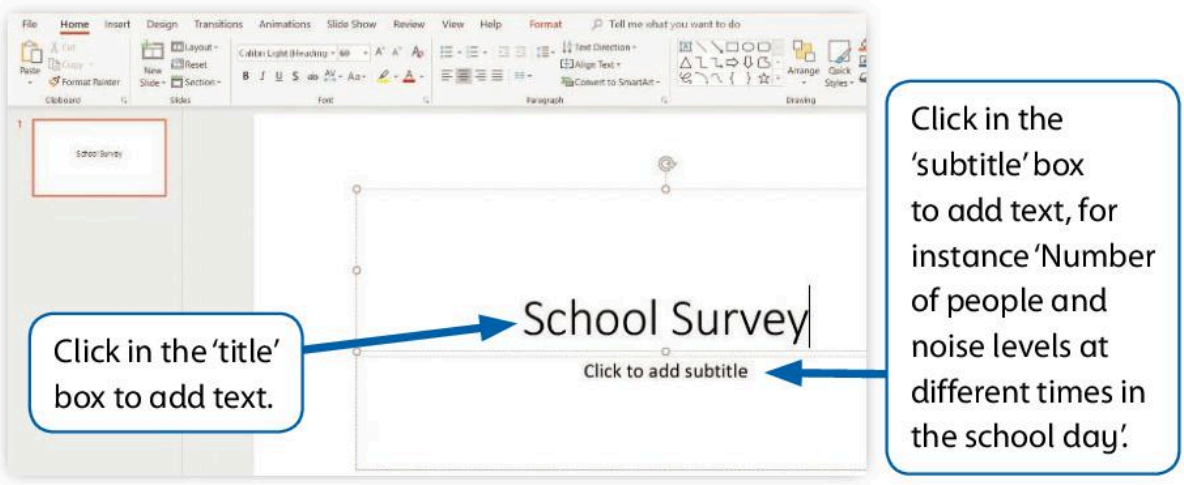
Create a presentation

You have collected and analysed your survey data. You will now use Microsoft PowerPoint to create a short presentation with text, images, charts and sound.

Start your presentation with a title and an introduction. These two **slides** will just have text on them.

Add a title slide

When you start a new presentation, Microsoft PowerPoint gives your first slide a special layout for a title. Click in each box to write your 'title' and 'subtitle'. Your title and subtitle need to fit in each box. You can also put your name in the subtitle box.



Add more slides

Once you have completed your title slide, add an introduction slide. Your introduction slide should help your audience understand what your presentation is about.

Click on 'New Slide' and choose the 'Title and Content' layout. You can put a heading at the top of the slide and your introduction text in the bottom box.

Use the 'Layout' menu (in the 'Slides' section) to change the layout of each slide. Some layouts allow you to put content side by side. Others let you include an image with a caption.

How to insert images on your slides

You can add an image by using the 'Insert' menu to import a picture you have already saved on your computer. The image might be:

- a drawing or graphic you made in another program
- a photograph you have saved on your computer.

You can add an image using clip art, or make your own image using the 'Shapes' menu.

You can also insert a screenshot from another program that is open at the time. Use the 'Screenshot' button to import it directly to your slide.

Activity

Using your notes from Lesson 5.3, create your presentation using PowerPoint or a similar app.

Use a variety of slide layouts. Add the chart(s) you created in Lesson 5.3 to the presentation.

Save your work.

Extra challenge

Explore the 'Shapes', 'Icons' and 'SmartArt' menus on the 'Insert' tab. Find a place in your presentation for one or more of these illustrations. How does your new illustration help your audience understand your presentation?

Be creative

Add some more images to your presentation. Use clip art or draw a map of the location where you carried out your survey. Try adding some labels.

Think again

What makes a presentation interesting?

Think about when someone has presented information to you. What makes it fun to listen to/watch?

5.5 Improve your presentation

In this lesson

You will learn:

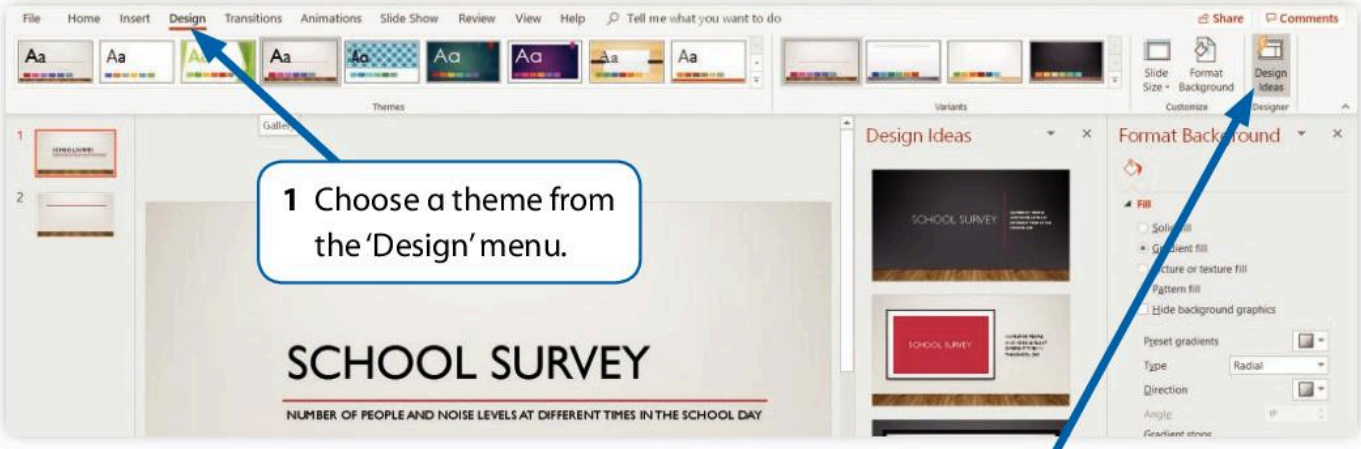
- how to apply a theme to your presentation
- how to add transitions
- how to add audio files.

Once you have got the content of your presentation right, you can add **transitions** and design themes to give it a professional look.

Apply a theme

Themes change the colours and fonts. They add patterns and borders to slides.

If you apply a theme, it is applied to all the slides in the presentation. So check each slide to make sure the theme works with your content.



Add transitions

Transitions are animations that occur during the change between slides. They can add interest – but can also be distracting. Experiment to find the best ones.

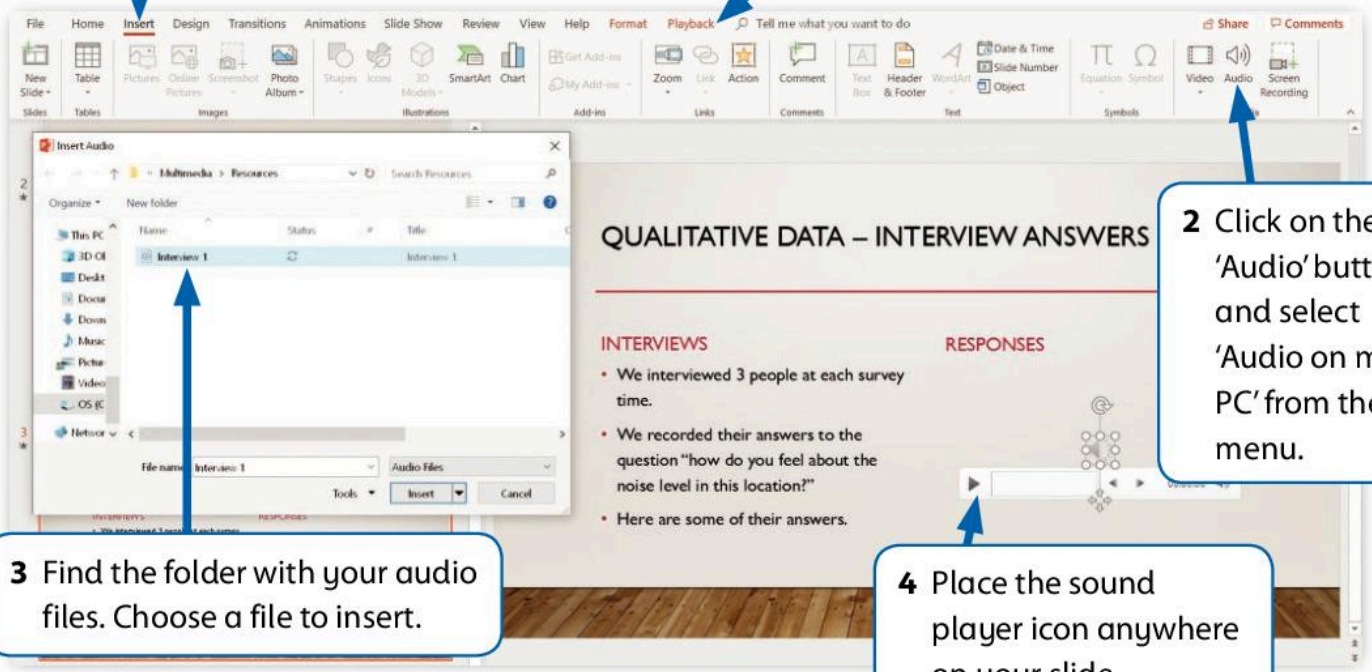
Choose a transition from the 'Transitions' menu. Your choice will be applied to the slide in the main window. You can apply the same transition to every slide or choose different transitions.

Add an audio file

You can add **audio files** to a slide in your presentation. The audio file will play when you show the slide.

1 Select the 'Insert' menu.

5 You can use the 'Playback' menu on the 'Audio Tools' ribbon to choose how the sound will be played in your presentation.



2 Click on the 'Audio' button and select 'Audio on my PC' from the menu.

3 Find the folder with your audio files. Choose a file to insert.

4 Place the sound player icon anywhere on your slide.

Choose 'Start: When Clicked On' to control the sound player when you are delivering your presentation.

Activity

Apply a theme to your presentation. Check all the slides. Does the design make them easy to read? Find the theme that works best.

Add a transition between each slide. Save your work.

Extra challenge

If you made audio recordings of interviews during your survey, choose the best one. Insert it in your presentation.

Explore more

You have created your presentation and it is ready for you to deliver. Ask your family and friends if they have ever given a speech or a presentation. What tips do they have for you?

5.6 Deliver your presentation

In this lesson

You will learn:

- how to print handouts
- how to use your slideshow to give a presentation to your class.

When it is time to deliver your presentation, your slideshow must be ready on the computer you will be using. Here are some things to remember:

- Think about what you are going to say. You can write notes on paper. You can also add notes to each slide in PowerPoint (see below). Write short notes that help you remember the most important points.
- If you have time, practise your presentation alone or with your team. You will feel more confident.
- If you are using a projector or large screen, stand to the side so that everyone can see.
- Talk slowly and clearly so that everyone in the room can hear you.
- You will need to change the slides as you go along. Click the left mouse button to change slides.

Handouts can help your audience understand and remember your presentation. They may be a printout of your slides or something else.

Create notes for your presentation

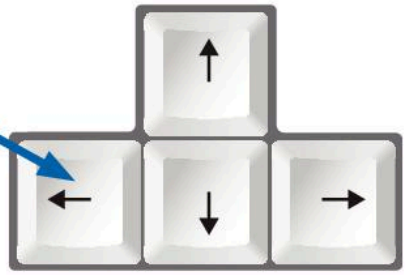
To add a note to a slide, click on the Notes icon at the bottom of the screen. Type your notes into the box. Your audience will not see your notes. You can print the notes page to help you deliver your presentation.

Deliver your presentation

To start your presentation, click on 'From Beginning' on the 'Slide Show' tab.

Use the left mouse button to change to the next slide.

If you go too fast, use the left arrow key on the keyboard to go back one slide.

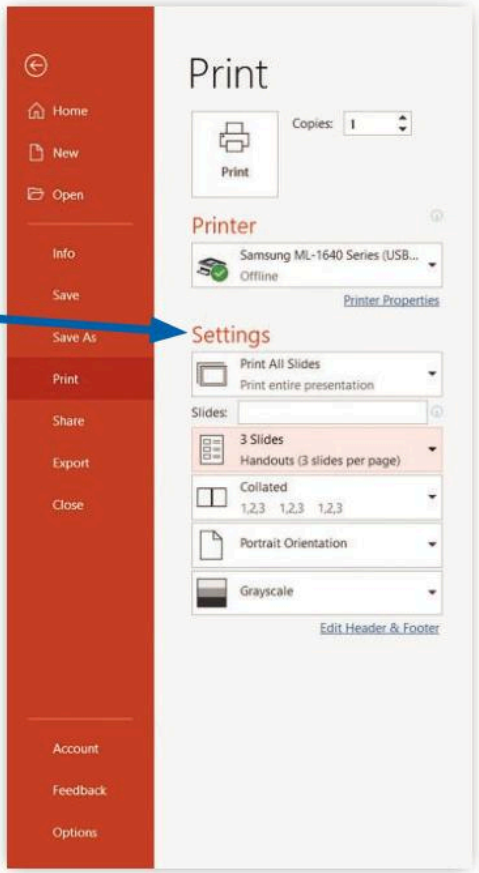


Remember, you need to click the mouse button to show the animations on your slides.

Print handouts for your audience

If you want to print handouts, choose 'Print' from the 'File' menu.

From the 'Settings' drop-down menu, choose a 'Notes Page' or 'Handouts' option.



Activity

Open your presentation on the computer you will be using to present your slideshow.

Deliver your presentation to your classmates. Talk slowly and calmly. Good luck!

Extra challenge

If your computer is connected to a printer, print handouts of your presentation for your classmates. Which 'Handouts' option works best for you?

Think again

Think about the information in your slideshow. How would you share it with people who can't be in the room with you?

Digital citizen of the future

We work with people from across the world. The presentation skills you have learned in this unit will help you communicate with friends and colleagues from all over the world.

Check what you know

You have learned

- how to plan a survey and data collection
- how to use digital devices and applications to collect data
- how to use software applications to analyse data
- how to use software applications to create and deliver a presentation of your survey results.

Activities

Download the 'Weather survey data' spreadsheet.

The spreadsheet contains data from a survey of rainfall and temperature.

1 Make a presentation to share this weather data with your classmates. Add these slides:

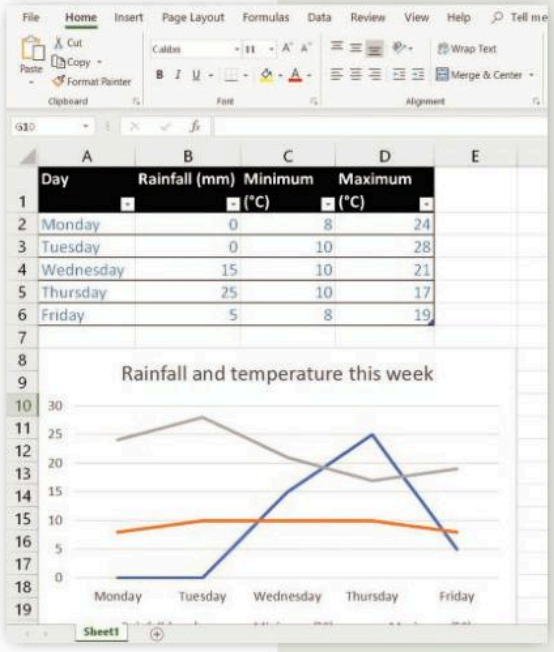
- A title slide showing the name and subject of your presentation.
- A slide explaining what the survey is about.

2 Add a slide showing a chart of the weather, based on the data in the spreadsheet.

- You can copy and paste the chart from the spreadsheet.
- If you have time, you can make changes to the chart.

3 Add a final slide that explains the data in the graph in no more than three bullet points.

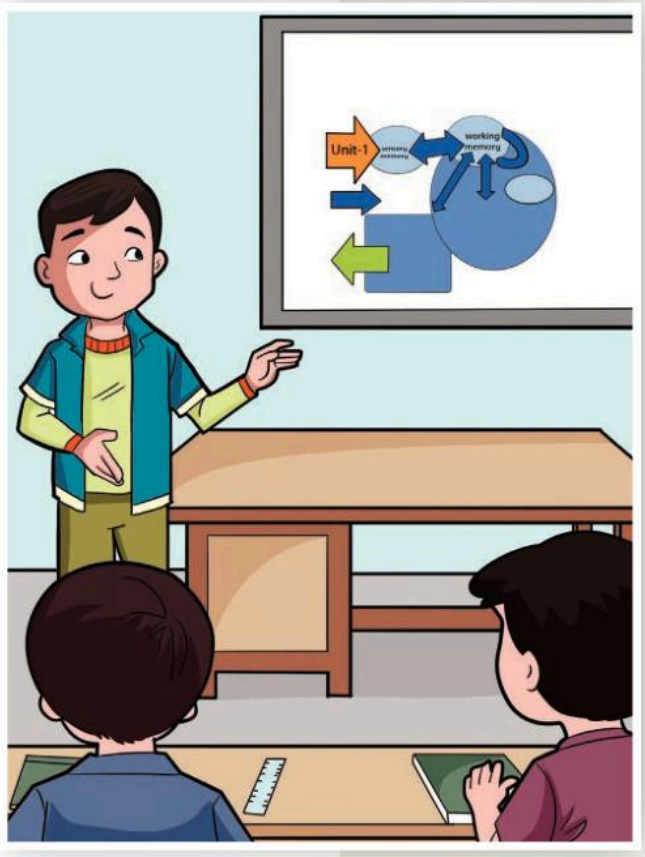
- Make sure you save your work.
- If you have time, print out some handouts for your presentation and share them with the class.



Test

In this unit you have collected data to find something out. You worked in a team. Answer these questions, thinking about the work you did.

1. Name one way that you can use technology to collect data.
2. Say one thing you did to help with the survey activity.
3. Describe how you used technology to collect data in your survey.
4. How did you help with the presentation?
5. Describe the work of your whole team and the things you found out through your survey.
6. Describe all the ways you helped your team. What decisions did you help to make?



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?

6 Numbers and data: Amir's parcels

You will learn

- how to store data in a structured table
- how to sort records into alphabetical order
- how to filter a data table to show selected data
- how to use data validation to check for errors
- how to use lists to make data entry easier
- how to use spreadsheet formulas to do calculations
- how to use logical tests to produce results.

A spreadsheet is a type of software application (app) that stores information in tables. Businesses use data tables to store information. Using a spreadsheet makes storing data quicker, easier and more accurate. Spreadsheets also help users do more with their data. Users can sort, filter and make calculations with their data. This helps them manage their business and make plans for the future.

In this unit you will use a spreadsheet to store information about a company called Amir's Parcels-by-Bike. The company delivers packages by bicycle in a busy city. The spreadsheet you will make stores information about the packages the company delivers. The spreadsheet will help Amir manage his company.

Talk about...

Accurate data is important to all businesses. Without accurate data, businesses can't make the right decisions.

What is the risk if computers store inaccurate data? Think of the kinds of business problems that may happen if there is inaccurate data in a business computer system.

Learning outcomes: Use software to structure, sort and filter data

Class activity

data table validation
filter sort primary key
list validation validation criteria
logical test calculated field

The project in this unit is about a business that delivers parcels. This kind of business is sometimes called a courier service.

- Investigate the courier or parcel delivery services in your area. How much do they charge to deliver a parcel?

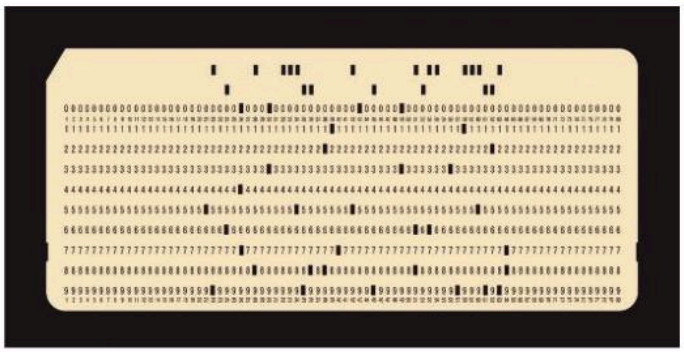
Now work in groups. Imagine that you manage a business that delivers parcels. What data would you store in a spreadsheet? Think about the information that you would need to manage your business. Here are some examples:

- how many parcels you had delivered
- how far your delivery workers had cycled
- how much you should charge your customers.

Discuss your findings with the rest of the class.

Did you know?

The earliest machines for data storage held information on paper cards with holes punched in them. A machine found the data you wanted by putting metal rods through the punched holes. Later, punch card machines could store about 80 characters on a single card (about 80 bytes of data). It would take 200 million cards to store the data that you can fit onto a modern 16GB memory card!



6.1 Build a spreadsheet data table

In this lesson

You will learn:

- how to explain what a data table is
- how to format a spreadsheet as a data table.

Spiral back

Last year, you learned how to use spreadsheet functions and formulas to help a business plan new products. In this unit you will learn how to use spreadsheet functions to record and analyse data to help a business manage its day-to-day operations.

In this unit you will work with data about a parcel delivery company. The data you will use is provided in a spreadsheet.

Download the spreadsheet 'Amir deliveries data'.

What is a data table?

'Data' is the name for facts and figures. A spreadsheet app lets you organise data in a structured format called a **data table**.

When data is formatted as a data table, you can carry out different tasks.

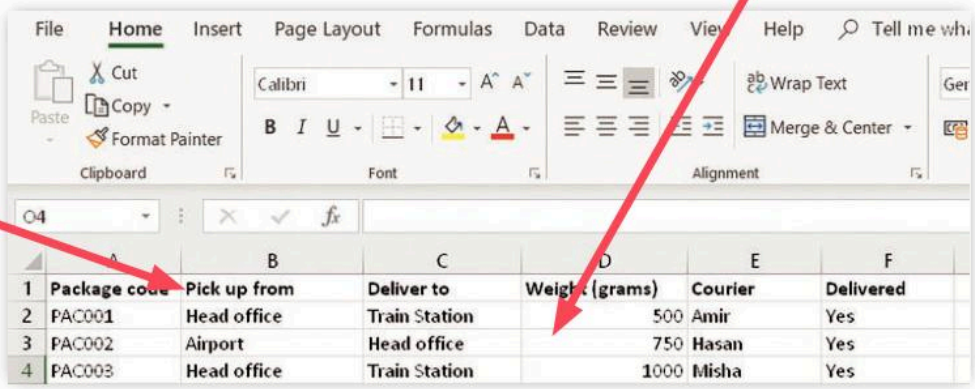
How data becomes information

On its own, data may not mean much. If you know how to sort, format and analyse the data, it becomes meaningful.

Data = 2 0 1 0 0 9 2 1 → Information = 21 September 2010

The rows of a table are called **records**. Each record has several fields. A record stores all the data about one thing, event or person. In this table, each record has data about one delivery.

A column in a data table is called a **field**. Each field stores one fact. The heading of the column is the field name. It explains what data is in the field.



	A	B	C	D	E	F
1	Package code	Pick up from	Deliver to	Weight (grams)	Courier	Delivered
2	PAC001	Head office	Train Station	500	Amir	Yes
3	PAC002	Airport	Head office	750	Hasan	Yes
4	PAC003	Head office	Train Station	1000	Misha	Yes

Data tables have a field that holds a different value in each record. This is called the primary key. You can use the **primary key** to identify each record. In this data table, the primary key field is 'Package code'. The package code begins with the letters PAC followed by a number.

How to make a data table

To turn a spreadsheet into a data table, you must select all the cells that contain data.

1 Drag the pointer across all the cells in the spreadsheet. Now they are highlighted.

2 Click to 'Format as Table'.

3 Choose the colour and style for the table.

A window like this will open:

1 This shows you the range of cells you have selected.

2 Your data includes headers in the top row. Make sure this box is ticked.

3 Click on 'OK'.

Activity

Open the spreadsheet 'Amir deliveries data'.

Select all the cells that make up the spreadsheet.

Format these cells as a data table.

Save your work.

Extra challenge

Experiment with different table colours and styles. Which do you like best? Why?

Think again

In this data table, every parcel has a unique code number.

Why does the business use code numbers to identify parcels?

	A	B	C	D	E	F
1	Package code	Pick up from	Deliver to	Weight (grams)	Courier	Delivered
2	PAC001	Head office	Train Station	500	Amir	Yes
3	PAC002	Airport	Head office	750.75	Hasan	Yes
4	PAC003	Head office	Train Station	1000	Misha	Yes
5	PAC004	Law centre	Bank	-250	Amir	Yes
6	PAC005	Housing agency	Head office	250	Misha	Yes
7	PAC006	Warehouse	Shop	100	Nadia	Yes
8	PAC007	Law centre	Train Station	1000	Amir	No
9	PAC008	Head office	Law Centre	500	Hasan	No

6.2 Sort and filter

In this lesson

You will learn:

- how to sort a data table into alphabetical order
- how to filter a data table so that only some of the records are shown.

What is sorting?

Sorting means putting the records of the data table in order. You must choose which fields to use to sort the table.

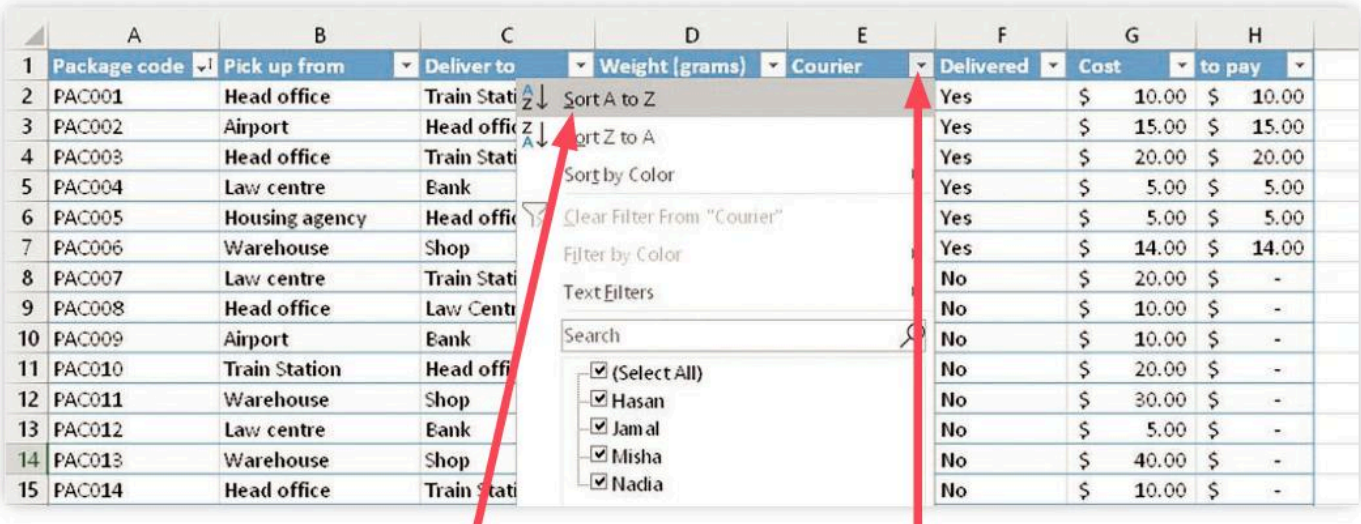
There are two main types of sorting: alphabetical and numerical order.

You must choose what the sort order should be – ascending (starting with the lowest value) or descending (starting with the highest value).

How to sort a data table

The field called 'Courier' stores the name of the courier who will deliver the parcel. You will sort the data table into courier order. All the deliveries for each courier will be grouped together.

Sorting the data is easy because you have formatted it in a table.



	A	B	C	D	E	F	G	H
1	Package code	Pick up from	Deliver to	Weight (grams)	Courier	Delivered	Cost	to pay
2	PAC001	Head office	Train Stati			Yes	$ 10.00	$ 10.00
3	PAC002	Airport	Head offic			Yes	$ 15.00	$ 15.00
4	PAC003	Head office	Train Stati			Yes	$ 20.00	$ 20.00
5	PAC004	Law centre	Bank			Yes	$ 5.00	$ 5.00
6	PAC005	Housing agency	Head offic			Yes	$ 5.00	$ 5.00
7	PAC006	Warehouse	Shop			Yes	$ 14.00	$ 14.00
8	PAC007	Law centre	Train Stati			No	$ 20.00	$ -
9	PAC008	Head office	Law Centr			No	$ 10.00	$ -
10	PAC009	Airport	Bank			No	$ 10.00	$ -
11	PAC010	Train Station	Head offi			No	$ 20.00	$ -
12	PAC011	Warehouse	Shop			No	$ 30.00	$ -
13	PAC012	Law centre	Bank			No	$ 5.00	$ -
14	PAC013	Warehouse	Shop			No	$ 40.00	$ -
15	PAC014	Head office	Train Stati			No	$ 10.00	$ -

2 Choose 'Sort A to Z' for an ascending sort order.

1 Click on the arrow next to the heading 'Courier'.

Now the table is sorted in order of courier name. To return the table to its original order, sort again using the 'Package code' field.

What is filtering?

Filter means selecting records from a data table so you can see them. You choose the records you want to see. You use rules called filter criteria. The spreadsheet app will show you only the records that match your filter criteria.

The records that don't match your criteria are still in the data table. They are hidden. When you turn the filter off, the records appear again.

How to add a filter

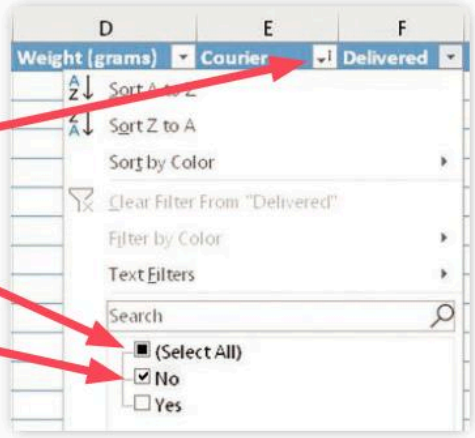
One field in the data table is called 'Delivered'. You will filter the data table so you only see the records that say 'No'. These are the parcels that have not been delivered yet.

You can add and remove filters using tick boxes.

1 Click the arrow next to the heading 'Delivered'.

2 Click to remove the tick next to 'Select All'.

3 Click to add a tick next to 'No'. Then click on 'OK'.



Now the data table only shows the records with the value 'No'.

E	F
Courier	Delivered
Amir	No

The filter symbol tells you that data in this column has been filtered.

Activity

Open your saved spreadsheet 'Amir deliveries data'. Sort the table in order of courier name.

Use a filter so that the table only shows the parcels that have been delivered.

Turn off the filter. Sort the table in 'Package code' order.

Think again

How could sorting and filtering data help Amir manage his business?

Extra challenge

Sort the table in order of parcel weight.

Sort the table in order of where the parcels will be delivered to.

Use a filter so that only Hasan's deliveries are shown.

6.3 Use validation to check your data

In this lesson

You will learn:

- how to explain why data checks are important
- how to use validation to check data.

Why are data checks important?

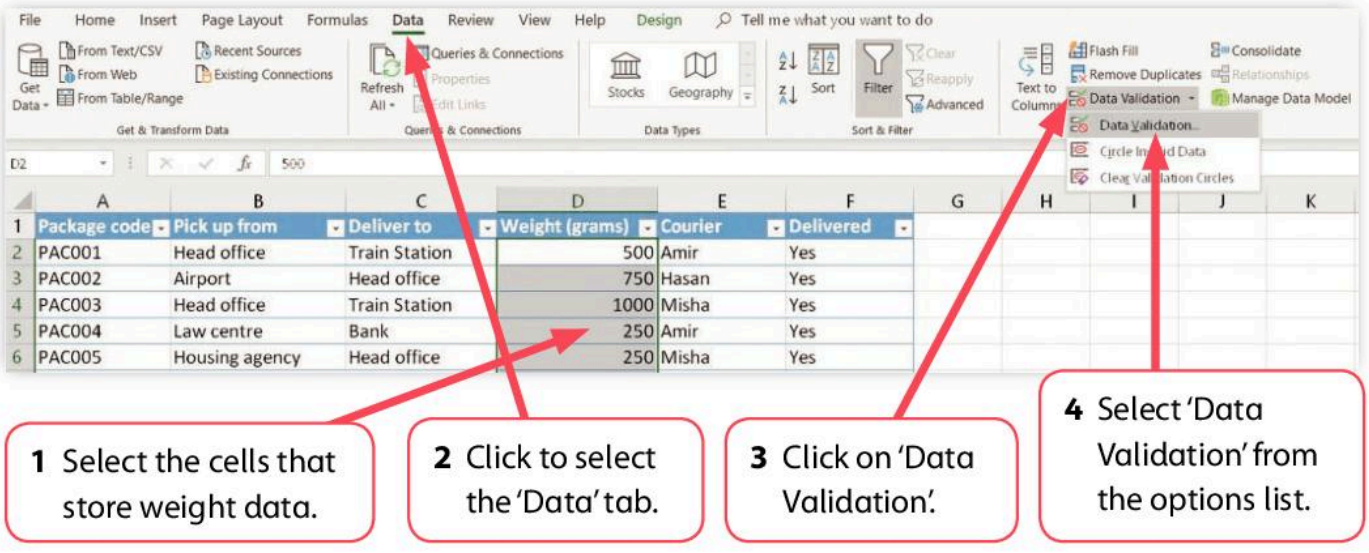
Data accuracy is important to businesses. Errors can cause big problems. They can be difficult to find when there is a lot of data. This is why software apps have data checks that can help you spot mistakes.

One type of data check is **validation**. Validation means that you decide on rules called **validation criteria**. The software checks all data against the criteria. If data does not match the criteria, then you know the data is wrong.

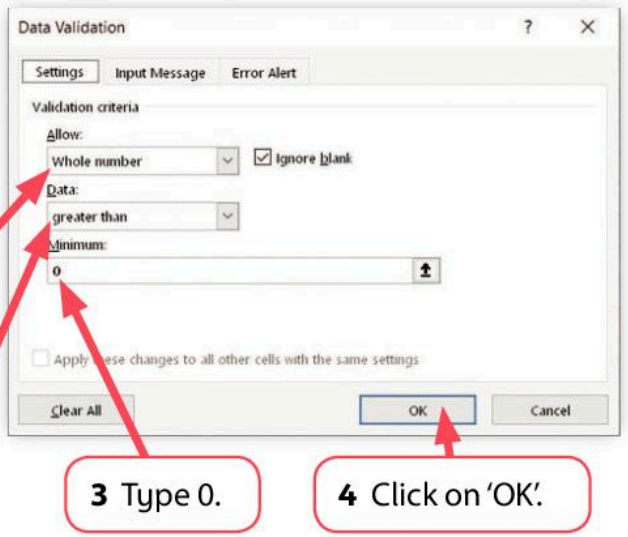
How to add data validation to a data table

You will use number-based validation to check the 'Weight' field. You will check the data using two criteria.

- The weight must be a whole number.
- The weight must be bigger than zero.



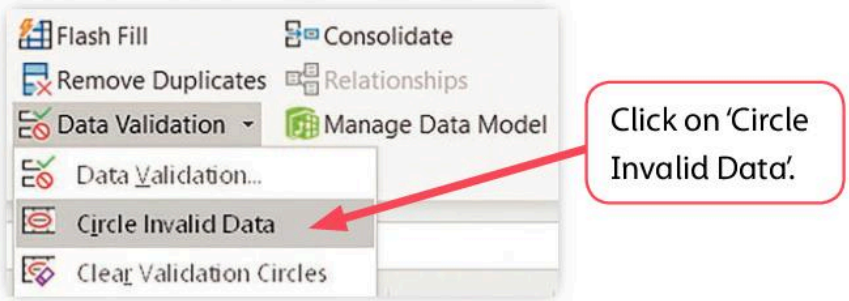
This window lets you add your validation criteria. First, you will set the rule that all weights must be whole numbers. Next, you will set the rule that weights must be bigger than zero.



How to check for errors

You can get the application to show you any invalid data.

The app finds and highlights all the cells with data that breaks the validation criteria.



Activity

Open your saved spreadsheet 'Amir deliveries data'. Add validation to the data as shown in this lesson.

Find the invalid data.

Replace the circled numbers with new data that does not break the validation criteria. The circles will disappear as the data is valid.

	A	B	C	D	E	F
1	Package code	Pick up from	Deliver to	Weight (grams)	Courier	Delivered
2	PAC001	Head office	Train Station	500	Amir	Yes
3	PAC002	Airport	Head office	750.75	Hasan	Yes
4	PAC003	Head office	Train Station	1000	Misha	Yes
5	PAC004	Law centre	Bank	-250	Amir	Yes
6	PAC005	Housing agency	Head office	250	Misha	Yes
7	PAC006	Warehouse	Shop	100	Nadia	Yes
8	PAC007	Law centre	Train Station	1000	Amir	No
9	PAC008	Head office	Law Centre	500	Hasan	No

Think again

Amir changed the validation check. He decided that every value in the weight column must be over 100 grams. One parcel fails this check. Give the code number of this parcel.

Extra challenge

The app will stop you from typing invalid data. It shows an error message. Explore what happens when you enter invalid data.

6.4 Use data lists

In this lesson

You will learn:

- how to use list validation
- how to use lists to simplify data entry.

Data entry or input is an important task for any business.

There can be problems with typing data.

- It can take a long time to input all the data.
- It is easy to make a mistake. A business needs to avoid data errors.
- The data may be inconsistent. This means that data may not be entered in the same way every time.

What is list validation?

List validation makes data input simpler. The user does not need to type data. They choose the right value from a list.

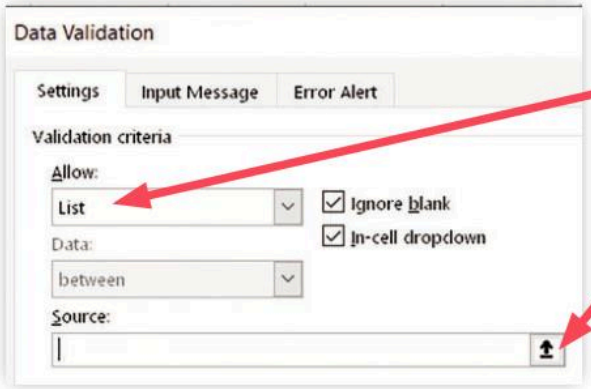
This has many advantages.

- Choosing from a list can be quicker than typing.
- There is less chance of making mistakes.
- The data is consistent (the same every time).

How to add list validation

You will add list validation to column C of your data table.

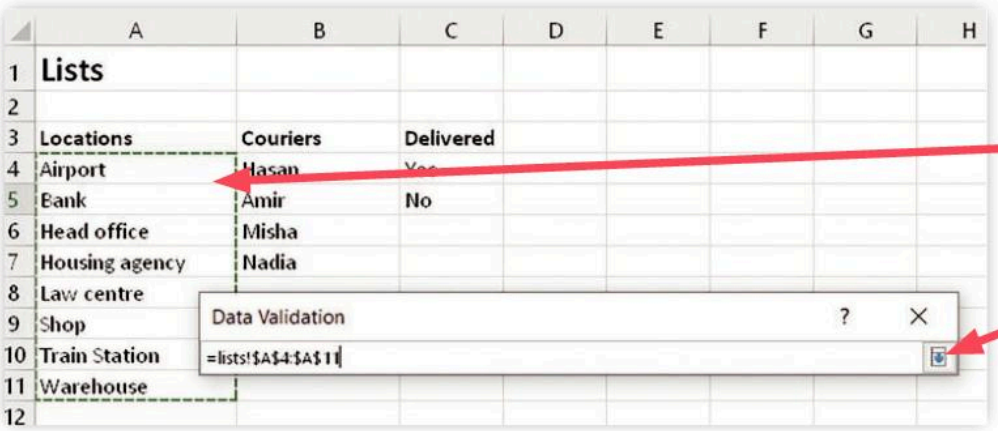
- Click on the 'C' at the top of column C. This will select the whole column.
- Open the 'Data' tab and click on 'Data Validation' like you did in the last lesson. The 'Data Validation' window will open.



1 Choose 'List'.

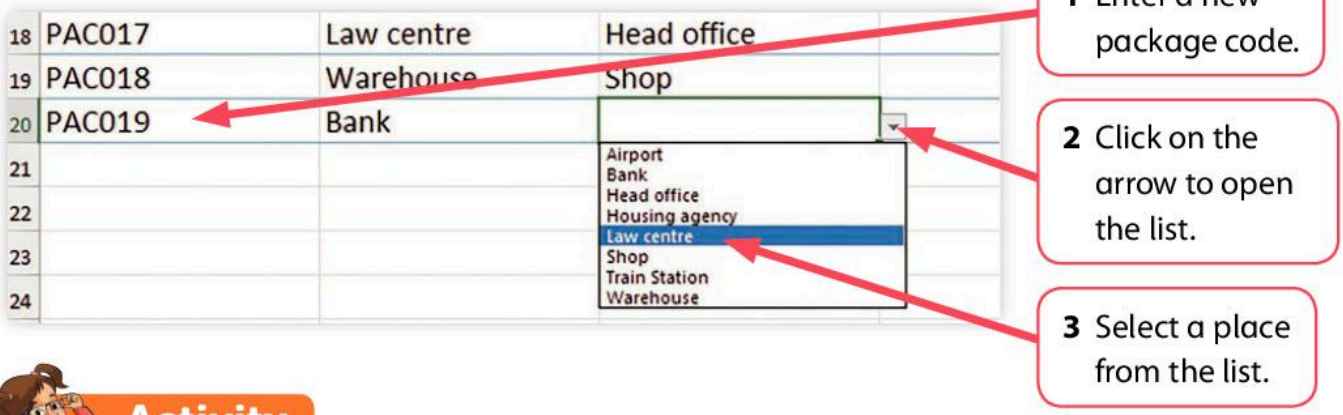
2 Click on this button to select the fields that will make your list.

Now you will select the list of places. Click on the 'Lists' tab at the bottom of the screen to open the worksheet with the lists.



How to use a data list for data entry

You can add a new record to the table.



Activity

Open your saved spreadsheet 'Amir deliveries data'. Add list validation to your table.

This picture shows new records for the data table. Add these records.

PAC019	Bank	Law centre	1000	Amir	No
PAC020	Warehouse	Airport	250	Hasan	Yes
PAC021	Train Station	Head office	500	Hasan	No

Save your work.

Extra challenge

Add list validation to the 'Delivered' and 'Courier' fields. The values are stored on the 'Lists' worksheet of the spreadsheet. Input new records to the data table, using the lists you have made.

Think again

Some columns in the data table could not use dropdown lists. Give an example.

6.5 Calculations

In this lesson

You will learn:

- how to describe what a 'calculated field' is
- how to add calculations to a data table.

Spiral back

In Student Books 4 and 5 you used formulas to do calculations in spreadsheets. You can create formulas in data tables in the same way.

You have made a data table. You have entered data into the fields by typing and by choosing from a list. Now you will add a **calculated** field.

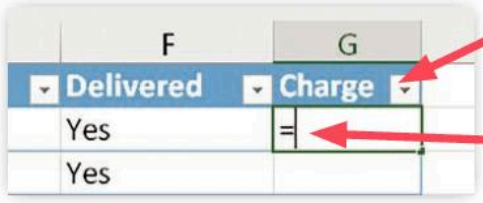
You will enter a formula and the app will work out the answer for every record in the table.

How to add a formula

In Amir's business, the charge for delivering a parcel is calculated by weight of the parcel. Amir uses the formula $\frac{\text{weight}}{50}$.

This calculation will appear in a new field called 'Charge'. You will enter a formula into this field.

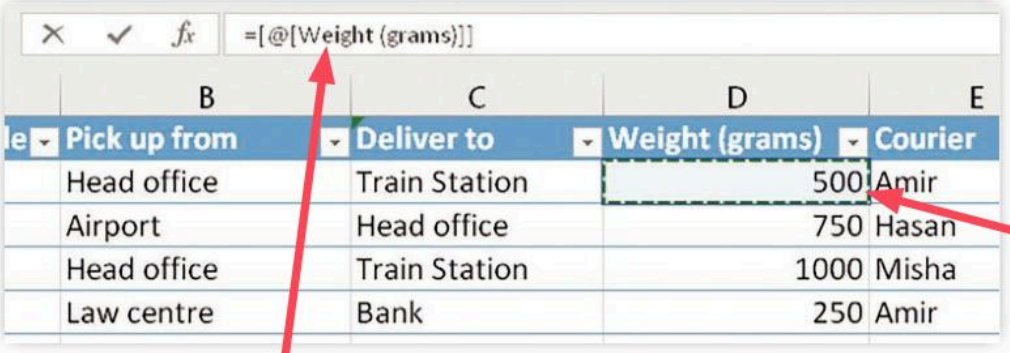
Remember: Every formula begins with an equals sign.



	F	G
	Delivered	Charge
	Yes	=
	Yes	

1 Type the name of the field.

2 Enter an equals sign (=) in the first row of the data table.

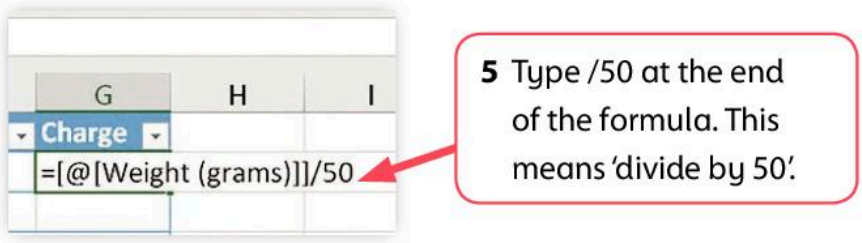


=[@[Weight (grams)]]

	B	C	D	E
le	Pick up from	Deliver to	Weight (grams)	Courier
	Head office	Train Station	500	Amir
	Airport	Head office	750	Hasan
	Head office	Train Station	1000	Misha
	Law centre	Bank	250	Amir

3 Click on the first cell in the 'Weight' field.

4 The app puts the name of the field into the formula. This means that the calculation will be made for all the data in this field.

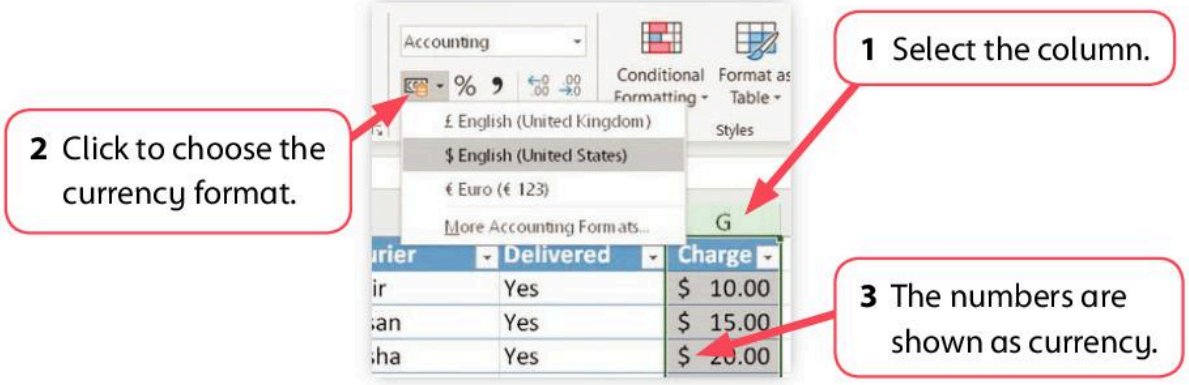


The charge should have been added to every record in the table.

How to format a field as currency

You can change the format of any data field. The format of a field means the way that the data in the field is displayed.

You will use the currency format. This shows numbers as money values.



Activity

Open your saved spreadsheed 'Amir deliveries data'. Add a 'Charge' field to the data table. Use a formula to calculate the charge for each delivery.

Change the format of the 'Charge' field to a currency format.

Extra challenge

Make changes to the weight of the parcels. See how the app calculates a new charge based on the new weight.

Explore more

Talk to family and friends about what they think is a fair price for parcel delivery. Does this match the prices in your spreadsheet?

6.6 Use a logical test

In this lesson

You will learn:

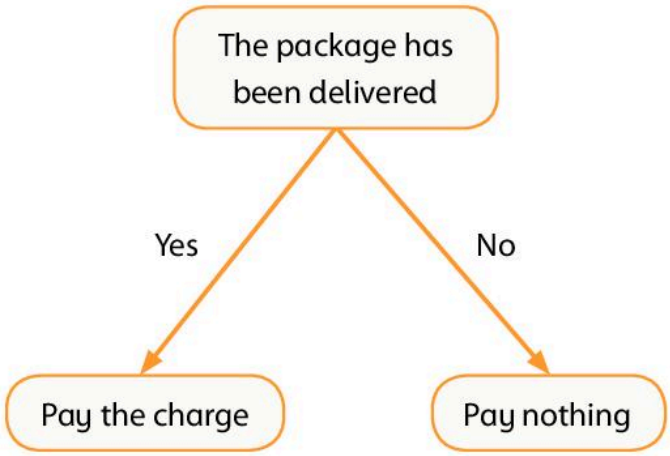
- how to use a logical test to make a choice
- how to use an IF command in a data table.

In the last lesson you used a formula to work out the charge for delivery of a parcel. The customer does not have to pay the charge until the parcel has been delivered.

In this lesson you will use a **logical test** to work out how much each customer has to pay today.

A logical test checks a statement and finds out if it is True or False. A logical test on the statement 'This parcel has been delivered.' can have the answer

- True (yes, it has been delivered) or
- False (no, it has not been delivered).



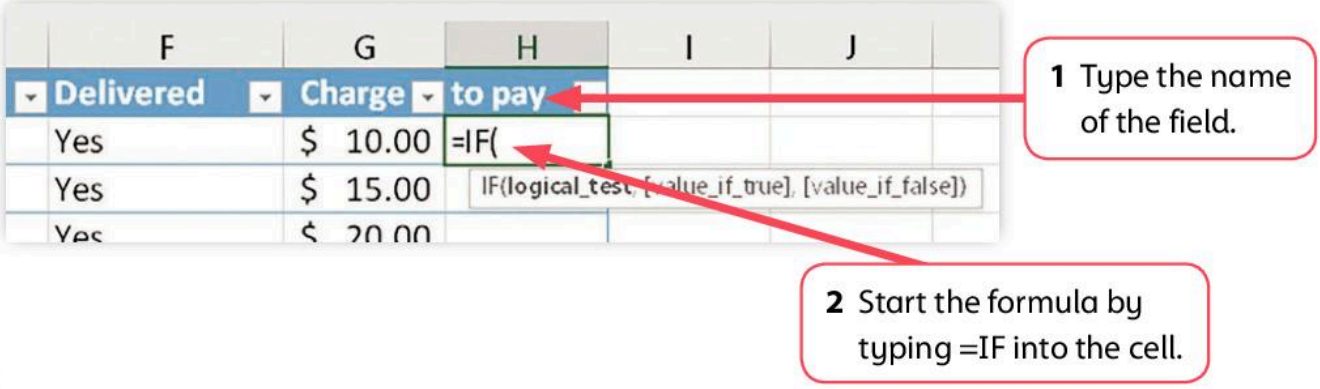
You can use an IF formula in your data table to calculate different values depending on the result of a logical test.

The field that holds the answer to the logical test is the 'Delivered' field. It holds the answer 'Yes' or 'No' for each parcel.

How to start an if formula

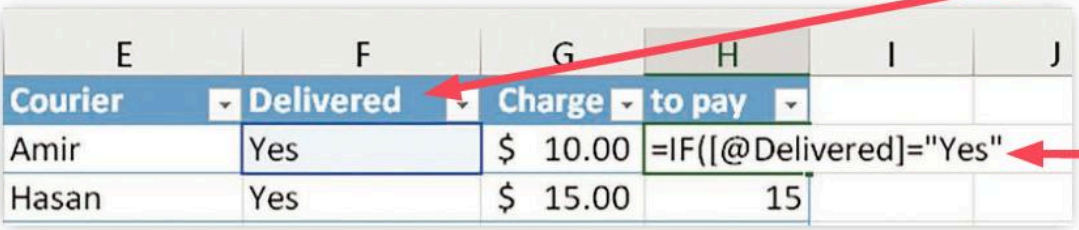
You will add a field to the end of the data table. It is called 'to pay'. This field will show how much the customer has to pay.

To start the formula, type: =IF(



How to enter the logical test

The test is: 'Is the answer in the "Delivered" field equal to "Yes"?' The formula in the data table will look like this: =IF([@Delivered]="Yes"



	E	F	G	H	I	J
	Courier	Delivered	Charge	to pay		
	Amir	Yes	$ 10.00	=IF([@Delivered]="Yes"		
	Hasan	Yes	$ 15.00	15		

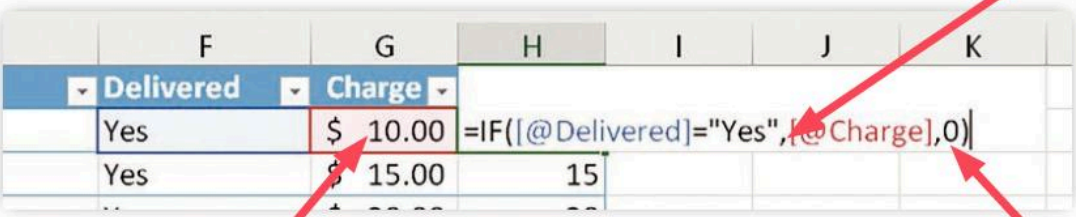
1 Click on the 'Delivered' field. The field name will be added to the formula.

2 Type ="Yes" to finish the logical test.

How to complete the formula

You have entered the logical test for the formula. Now you must add the choices the application should make, depending on the result of the test. The choices are:

- Set the value of the cell to the value of the charge.
- Set the value of the cell to zero.



	F	G	H	I	J	K
	Delivered	Charge				
	Yes	$ 10.00	=IF([@Delivered]="Yes",[@Charge],0)			
	Yes	$ 15.00	15			

1 Type a comma.

2 Click on the 'Charge' field.

3 Type another comma. Then type 0 (zero). Finish with a close bracket.

Press 'Enter' when you have finished the formula.

Activity

Open your saved spreadsheet 'Amir deliveries data'. Add a new field to the data table. The heading is 'to pay'.

Use an IF formula in this field to show the amounts customers need to pay for every parcel in the table.

Format the data in the 'to pay' as currency.

Extra challenge

Change some of the values in the 'Delivered' field from 'No' to 'Yes'. What happens?

Think again

Amir has to wait for the courier to return to the office and tell him that he has made the delivery.

How could Amir's business use technology to speed up getting this information?

Check what you know

You have learned

- how to store data in a structured table
- how to sort records into alphabetical order
- how to filter a data table to show selected data
- how to use data validation to check for errors
- how to use lists to make data entry easier
- how to use spreadsheet formulas to do calculations
- how to use logical tests to produce results.

Activities

	A	B	C
1	Student code	First name	Second name
2	S001	Ibrahim	Khan
3	S002	Jamal	Hussain
4	S003	Zara	Abidi
5	S004	Kurtis	Lincoln
6	S005	Leela	Masood
7	S006	Katie	Morgan
8	S007	Sian	Boyle
9	S008	Jatinder	Singh
10	S009	Martin	Green
11	S010	Yvette	Leland

1 Here is an example of a class register which a teacher made using a spreadsheet app.

a Create a spreadsheet like this, using the names of students in your class. Add no more than 10 rows of data.

b The teacher turned the register into a data table. Format the data as a data table.

c Sort the data table alphabetically by students' second names.

d Save and print the sorted table.

2 The teacher will use the spreadsheet to mark if students are present. The school secretary will phone every student who is absent to check if they are ill.

a Add a new column to your class register spreadsheet, with the heading 'Monday'.

b Enter data in the Monday column: '1' if the student is present, and '0' if the student is absent. Put these values into the cells of this column.

c Filter the spreadsheet so that you only see students who are absent. Print out the spreadsheet so you can send a list to the school secretary.

3 **Extension activity:** If you have time, add more columns to show attendance on other days of the week. Use list validation to give the teacher a drop-down menu to mark each student with a '1' or '0'.

Test

These questions relate to the spreadsheet you made for the activities. Print this out or look at it on the screen as you answer the questions. If you have not completed the activity, use this spreadsheet as an example.

	A	B	C	D
1	Student code	First name	Second name	Monday
2	S001	Ibrahim	Khan	1
3	S002	Jamal	Hussain	1
4	S003	Zara	Abidi	1
5	S004	Kurtis	Lincoln	0
6	S005	Leela	Masood	1
7	S006	Katie	Morgan	1
8	S007	Sian	Boyle	0
9	S008	Jatinder	Singh	1
10	S009	Martin	Green	1
11	S010	Yvette	Leland	1

1. The spreadsheet shows data about students. Which items of data does it store about each student?
2. In the activity you sorted students in order of their second names. What other sort order could you use? If you used this order, which student would come at the top of the list?
3. The teacher wants to send the school secretary a list of students who are absent on Monday. The school secretary will phone their homes to see if they are ill. Explain how you can use the spreadsheet to print out a list for the secretary.
4. What extra item of data could you add to the table to help the secretary do her job?

Self-evaluation

- I answered test question 1.
- I started activity 1 by entering data into a spreadsheet.
- I answered test questions 1 and 2.
- I completed activity 1, making a sorted data table.
- I answered all the test questions.
- I completed activity 1 and 2.

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

Glossary

actuator component of a robot that turns instructions from the controller into movement

algorithm a set of steps to solve a problem. It must show the right steps. The steps must be in the right sequence. You can use an algorithm to plan a program

analogue a measurement that varies continuously within a range. For example, temperature is an analogue measurement. Temperature doesn't jump from one precise number value to another, but changes in a smooth curve

artificial intelligence the ability of a computer to think and learn in a way similar to humans

audio file a computer file containing digital data about a sound recording

average worked out by calculating the total of a series of values and dividing by the number of values in the series

body an area in the middle of a document or web page. The body contains most of the page content

bumper a sensor used in robots to detect contact with another object

calculated field a spreadsheet cell that contains the result of a calculation or formula. For example, a 'Total' field showing the sum of a column of numbers

collaborative robots robots that work together as a team to complete a task. Sometimes collaborative robots work with humans

computer technician a person who installs and repairs computers. A technician installs software on computers

conditional loop a loop which is controlled by a logical test. In Scratch, the conditional loop repeats until the test is True

controller the component in a robot or control system that carries out programmed instructions. The controller receives messages from sensors and sends instructions to actuators

control loop an algorithm used in a control system. For example, a control loop can maintain temperature at a level set by the user. Sensors feed back information to a controller. The controller uses the information to maintain a constant temperature

control system a system used to control an environment or process. Examples include temperature control systems and security systems. Control systems use a controller and sensors

counter loop a loop controlled by a counter. When the loop has repeated a set number of times it will stop. It is also called a counter-controlled loop or a fixed loop

data logger a device or application that collects data from sensors and stores it. Data loggers usually collect data at set intervals, for example, 'once per minute'

data table a set of data in a spreadsheet with a header row. Data tables help you to sort and filter data more easily

drone a robot that flies. A drone can be designed like a helicopter or a plane

edit make changes to a file; for example,s change the commands in a program or script

feedback loop see control loop

field a column of a data table that stores a single item of data

fieldwork work done away from an office. Surveys are often conducted by fieldworkers. Fieldworkers often use portable devices during fieldwork

filter a spreadsheet function that lets you see only data that you are looking for

fixed loop another name for a counter loop

footer an area at the bottom of a document or web page. The footer contains information such as the date and page number. The footer is automatically repeated on every page in a document or website

handout a printed version of a presentation. The handout shows all the slides together on one or more pieces of paper

header an area at the top of a document or web page. The header can contain the title of the document and a logo. In a web page, the header can contain a menu. A header is automatically repeated at the top of every page in a document or website

headings short descriptions that tell you what a website or a piece of text is about. A heading is larger than normal text so that it stands out

images photographs, drawings or cartoons. Images make the content of a document or web page more interesting and easier to understand

import bringing data into an application so that you can work with it and use it

links when you click on a link, you leave the page you are on and go to a new page. Links are also called hyperlinks. A link can be a piece of text or an image. Links are often made to stand out. For example, a text link may be blue and underlined

list validation restricts data input to a list of choices

logical test a question in a spreadsheet formula that can be answered with True or False

logo websites use a logo to tell you what the site is about. A logo can also tell you who owns the website. If the website is owned by a company, it will have the company logo at the top

menu helps you find your way around software applications or websites. Clicking on a menu option takes you directly to another part of the software application or website

module stores a group of commands. The programmer makes the module and gives it a name. The module can be included in a program. When the computer sees the name of the module, it will carry out all the stored commands

nanobot a microscopic robot that is small enough to work in confined spaces. Nanobots are experimental – they are being developed for medical use. One day they may be used to fight disease inside the body

pressure sensor a sensor that can be used to detect changes in pressure. Pressure sensors are often used in home alarm systems to detect intruders

primary key a piece of data that is unique to one record or one row in a data table. A code number is often used as a primary key

proximity sensor a sensor that can detect distance between a robot and another object. Proximity sensors can be used to detect movement

qualitative data data that cannot be counted. Qualitative data describes things

quantitative data data that can be counted, for example, the amount of rainfall in a day

record a row of a data table that stores all the information about one thing or event or person

repurpose a program can be repurposed. This means that with some changes it can be used for a new purpose. This saves time and is less work for the programmer. It also reduces the risk of errors because the program has been used before

requirement a program requirement sets out what the program has to do. You need to be clear about the requirement before you begin to make an algorithm

reuse a program can be reused. This means a program is used a second time. Like repurposing, this is easier and more reliable

robot a machine that can be programmed to carry out a task automatically

robot arm a robot in the form of a human arm. A robot arm has joints that can bend and twist so that it can do jobs a human would normally do

robot car a vehicle that can drive without human control. Also called a self-drive car

satellite navigation (satnav) a system used to guide cars and other vehicles to a destination chosen by the driver

script a short program that typically controls one thing

sensor a component used in a robot or control system to sense events in the environment. For example, a thermometer is a sensor that detects temperature in the air or a liquid

sequence the sequence of commands in an algorithm is the order in which the commands happen

slide a screen shown during a presentation. Each presentation is made up of one or more slides that are shown one after the other

smart home a home where functions such as lighting, heating and alarms can be controlled remotely using the internet

software engineer a programmer. A software engineer writes programs for computers and keeps the programs up to date

sort spreadsheet applications can sort data into an order you choose

spellchecker a function in most software applications. A spellchecker checks spelling and grammar and suggests corrections

temperature sensor a sensor used in robots and control systems to detect changes in temperature

trace an algorithm when we trace an algorithm, we follow through all the commands and make a note of what the results are. This is a way to check for errors

transcribe copying information from one format to another. For example, copying the words from a voice recording by typing them into a word-processed document

transition a kind of animation that is shown when a presentation slide changes

upload copy a file from your own computer to another computer. In Scratch, this means you copy it to the Scratch web server

validation data validation means checking that data is correct. You can use validation to stop mistakes when data is entered into a spreadsheet

validation criteria the rules you set for data validation. For example, a rule for a number cell could be 'the number must be greater than 0'

vibration sensor a sensor used in robots and control systems to detect vibration

web page a page on the web that contains information on a subject. Web pages contain text and images. Web pages can also contain video, audio and animations. Web pages often contain links to other web pages

web page editor application software used to make web pages

wireframe an outline design used in the design of web pages and other documents. A wireframe shows where major elements of a page will appear on-screen or in a printed document. It is a useful planning tool

***x/y* coordinate** two numbers that set the position of a point on the screen

***x* coordinate** the number that sets the left/right position of a point on the screen

***y* coordinate** the number that sets the up/down position of a point on the screen

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 2019

British Library Cataloguing in Publication Data

Data available

ISBN 978-0-19-849784-4

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 China by Golden Cup

Acknowledgements

The publisher and authors would like to thank the following for permission to use photographs and other copyright material:

All photos © Shutterstock, except: **p15(l):** Archive Photos/Getty Images; **p19:** BSIP SA / Alamy Stock Photo; **p21:** web page © WC3; **p35:** Marmaduke St. John/Alamy Stock Photo; **p52(t):** Licensed from mazegenerator.net; **p69**: YOSHIKAZU TSUNO/AFP/Getty Images; **p85(bl):** zionbalkon/Fotolia.

Cover illustration by Patrick Hruby. All other artwork by QBS Media Services Inc.

Oxford International Primary

Computing 6

Student Book

This **new edition of *Oxford International Primary Computing*** has been designed to teach students the vital computing skills they need for today's digital world. Across the six levels, students grow into digital citizens who have 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. 9781382007450 9781382007467

Leading on to:

9780198497851 9780198497868 9780198497875 9781382007474

OXFORD UNIVERSITY PRESS

How to get in touch:
web www.oxfordprimary.com
email schools.enquiries.uk@oup.com
tel +44 (0)1536 452610
fax +44 (0)1865 313472

ISBN 978-0-19-849784-4
9 780198 497844