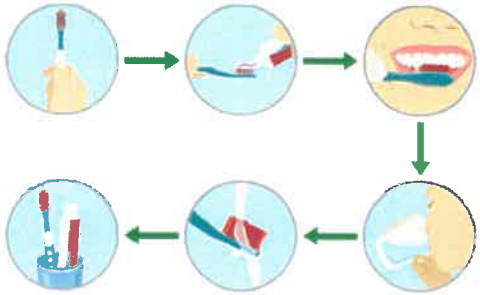
Knowledge Organiser: Programming — Procedures and Functions

Procedures and Functions

Summary—When writing programs, we should avoid long, repetitive code. Procedures and functions help to keep our programs simple and short.

A **procedure** is a small section of a program that performs a specific task. Procedures can be used repeatedly throughout a program.

A **function** is also a small section of a program that performs a specific task that can be used repeatedly throughout a program, but the task is usually a calculation. Functions perform the task and return a value to the main program.



What is a Procedure

A **procedure** is a small section of a program that performs a specific task. Procedures can be used repeatedly throughout a program.

Why use a Procedure

Program code is easier to read and understand when it is broken up into smaller sections. By breaking a program up into these sections, or procedures, code can be made shorter and simpler.

Writing a Procedure

Writing a **procedure** is extremely simple. Every procedure needs:
a name

the **program** code to perform the task

Running a Procedure

Once a **procedure** is named and written, it can be **called** at any point in the **program**.

To call a procedure in **Python**, simply use its name (include the brackets)

What is a Function

A **function** is also a small section of a program that performs a specific task that can be used repeatedly throughout a program, but the task is usually a calculation. Functions perform the task and return a value to the main program.

Both functions and procedures are small sections of code that can be repeated through a program. The difference between them is that functions return a value to the program where procedures perform a specific task.

Why do we use Functions

Program code is easier to read and understand when it is broken up into smaller sections. Functions and procedures can make code shorter, simpler and easier to write.

A function is just like a procedure, except that you wait to see what the result is.

Running a Function

Once a **function** is named and written, it can be **called** at any point in the **program**. 'Calling' a function means running it.

To call a function in **Python**, simply use its name (include the brackets) and the value it needs to use for calculation, plus a variable that the returned value will be stored in.

Key Vocabulary

Algorithm	A sequence of logical instructions for carrying out a task. In computing, algorithms are needed to design computer programs.
Calling	Starting or running a function or procedure.
Code	Instructions in a computer program.
Debug	The process of finding and correcting programming errors.
Function	A section of code that, when programming, can be called by another part of the program with the purpose of returning one single value.
Procedure	A section of computer code that performs a specific task.
Program	Sequences of instructions for a computer.
Python	A high-level programming language.
Statement	The smallest element of a programming language which expresses an action to be carried out.



Some images is copyright © 2016 BBC.

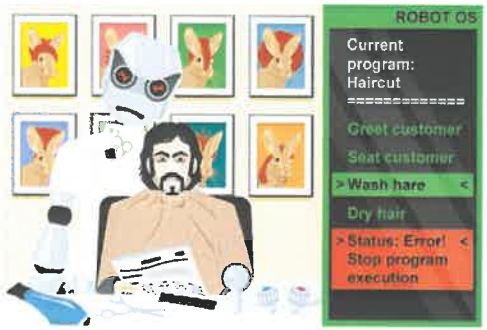
Knowledge Organiser: Programming — Writing Error Free Code

Writing Error Free Code

Summary
When writing programs, code should be as legible and error free as possible. **Debugging** helps keep code free of errors and documenting helps keep code clear enough to read.

When programs are written, it is likely that at least some errors will creep in. Errors in programs are often referred to as **bugs**.

It is vital that programs are as free of errors as possible. Errors can cause a program to produce unexpected results, or **crash**.



Syntax Error

Syntax errors will cause a program to crash or not run at all. The **program** may run until it encounters a syntax

Types of Syntax Error

More than one type of syntax error may exist. There may be:
incorrectly spelled **statements**
incorrectly spelled **variables**
missing punctuation (quotes, brackets, etc)

Any one or more of these errors may exist in a program, and each will cause the program to crash or not run at all.

Fixing syntax errors is simple. Spotting them is more difficult. Some of the most common syntax errors are **things that come in pairs**. Always check for speech marks and brackets first if your program

Fixing syntax errors is simple.
Spotting them is more difficult.

Logic Errors

Logic errors occur when there is a **fault** in the **logic** or **structure** of the problem. Logic errors do not usually cause a program to **crash.** However, logic errors can cause a program to produce **unexpected results.**

Types of Logic Error

More than one type of logic error may exist. Parts of the program may:
be in the wrong **sequence**
have the wrong **Boolean** expression
use the wrong **data type**
be missing altogether

Any one or more of these errors may exist in a pro-

Documenting Code

Code that is hard to read makes it difficult to understand what the **program** is trying to do. It also makes it difficult to understand the purpose of
any **variables, procedures** and **functions**. Errors are easier to fix when we understand the code we are reading through.

When we **document** code, we make it easier to read and understand. We also document code in case:
we need to come back to it at a later date

Key Vocabulary

Boolean	A data type in computing which only has two possible values, true or false.
Bug	An error in a program.
Code	Instructions in a computer program.
Commenting	Adding one or more sentences to explain the purpose of a section of code.
Crash	When an application or operating system no longer responds. Crashes can be caused by software and hardware.
Data Type	The format in which a variable or constant holds data, such as 'integer' or 'string'.
Debug	The process of finding and correcting programming errors.
Documenting	Making code easier to understand by giving meaningful names for variables, procedures and functions and by adding comments to explain the purpose of instructions.
Execute	To run a computer program.
Function	A section of code that, when programming, can be called by another part of the program with the purpose of returning one single value.
Instruction	A single action that can be performed by a computer processor.
Logic error	A fault in the logic or structure of the problem.
Procedure	A section of computer code that performs a specific task.
Program	Sequences of instructions for a computer.
Programming Language	A language used by a programmer to write a piece of software.
Python	A high-level programming language.
Sequence	In computer programming, this is a set of instructions that follow on one from another.
Statement	The smallest element of a programming language which expresses an action to be carried out.

Some images is copyright © 2016 BBC.

Knowledge Organiser: Digital Devices

Digital Devices

It is easy to recognise that personal computers, laptops and mobile devices are computers, but computers are also hidden in many more devices. Because they are relied on so heavily, knowing what they are and how to use them is valuable. Digital devices may be input, output or storage devices , however on a basic level, they all operate through

Input Devices

An **input** device is any piece of computer **hardware used to provide data to a computer system.** Examples include:

- keyboard
- mouse
- scanner
- digital camera

Storage Devices

A **storage device** is a piece of computer equipment which can be **used to store data.** Examples include:

- Hard disk drive
- DVD disk
- USB stick
- Memory Card

Output Devices

An **output** device is any piece of computer hardware **used to communicate the results of data that has been processed.** Examples include:

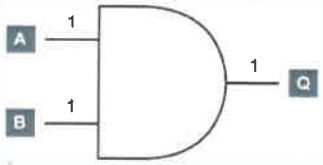
- monitor
- printer
- speaker

What is a Logic Gate

A circuit board can be found inside most digital devices. It is a flat, thin board that has tiny electrical **components** built onto it. Many electronic circuits have to make decisions. They look at one or more **inputs** and use these to determine the **outputs** from the circuit. The process of doing this uses electronic logic, which is based on digital switches called **gates.** Each input and output of the **logic gates** must be one of two states:

True or **1** or **on** **False** or **0** or **off**

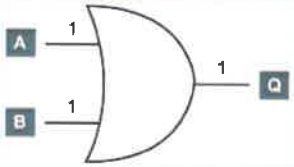
AND



An **AND** gate usually has two inputs. **AND tells us that both Input A AND Input B have to be 1 (or ON) in order for the output to be 1. Otherwise the output is 0.**

The Boolean expression can be written as **Q = A AND B.**

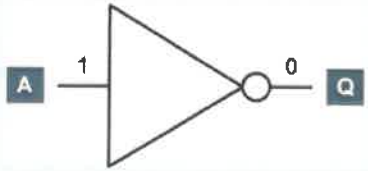
OR



An **OR** gate has two inputs. **OR tells us that EITHER Input A OR Input B has to be 1 (or ON) in order for the output to be 1. Otherwise the output is 0.**

The Boolean expression can be written as **Q = A OR B.**

NOT



A NOT gate has just one input. NOT tells us that Input A has to be 0 (or OFF) in order for the output to be 1. Otherwise the output is 0. A

The Boolean expression is written as Q = NOT A.

Key Vocabulary

Binary	A number system that contains two symbols, 0 and 1. Also known as base 2.
Boolean	A data type in computing which only has two possible values, true or false.
Component	Working parts of a product or system.
Hardware	The physical parts of a computer system, eg a graphics card, hard disk drive or CD drive.
Input	Data which is inserted into a system for processing and/or storage.
Logic Gate	Circuit components which take several inputs, compare the inputs with each other, and provide a single output based on logical functions such as AND, OR and NOT.
Output	Data which is sent out of a system.
Truth Table	Used to assess possible results of a Boolean algebra statement.

Truth Tables

Input A	Input B	Input Q
0	0	0
0	1	0
1	0	0
1	1	1

AND

Input A	Input B	Input Q
0	0	0
0	1	1
1	0	1
1	1	1

OR

Input A	Input Q
1	0
0	1

NOT

Some images © copyright © 2016 BBC.

Knowledge Organiser: Hardware and Software

Summary

Hardware—This is the physical parts of the computer. Software—These are the programs that run on a computer.

Input Devices. This is any piece of computer hardware used to provide data to a computer system. For example, Keyboard, mouse, scanner, digital camera or webcam.

Output Devices. An Output device is any piece of computer hardware used to communicate the results of data that has been processed. Examples could include a Monitor, Printer, Speaker or Headphones.

Storage Devices. A Storage Device is a piece of computer equipment which can be used to store data. Examples could include Hard Disk Drive, DVD Drive and USB sticks.

Logic Gates. Each input and output of logic gates must be one of two states. True or 1 or on. False or 0 or off. Logic Gates use Boolean operators, the most common are AND, OR and NOT.

Systems Software

Software makes hardware useful, it gives the computer the instructions they need to operate. When hardware runs software, it loads the software into its RAM.

There are two mains types of software. Systems software and Applications Software.

Systems software includes the Operating System, Drivers and Utility Software.

All computers must have an Operating System in order to operate.

The Operating System provides the User Interface, it manages the memory in the computer. It also looks after system security including user names and passwords.

Examples of Operating Systems include Windows, Linux, Mac OSX, Android and iOS.



Applications Software

Applications Software is used to carry out tasks on a computer. For example these could include a Word Processor, Spreadsheet, Database or Graphics Software.

This sort of software is called general purpose software because they can carry out many different tasks using those applications.

Applications are also used on Smart phones to to many different types of things. For example Social Media Applications, messaging or music and Video sharing applications.

The word 'App' is short for Application software. This is a piece of software that allows us to do an everyday task.



Key Vocabulary

Application	A software program that allows a user to perform a specific task.
Driver	A piece of software which controls a peripheral device.
Graphics Software	An application that assists a user in creating or editing images
Hardware	The physical parts of a computer system e.g. a graphics card, hard disk drive or CD drive.
Instruction	A single action that can be performed by a computer processor.
Operating System	The software that manages the hardware and software resources in a computer system.
RAM	Random Access Memory. Memory that is used for processing tasks. It is volatile, this means when the power supply is turned off the information stored in the memory disappears.
Smartphone	A mobile phone with a powerful processor that is capable of running applications and accessing the internet..
Software	The programs, applications and data in a computer system that aren't physical
User Interface	The means by which a user interacts with a computer or device.
Utility Software	These are known as utilities, e.g. virus protection or a firewall
Web Browser	Software that displays web pages
Word Processor	An application used to write, edit and format text.

Knowledge Organiser: Components of the CPU

Summary

Computers require input hardware, processing hardware and output hardware. The hardware that defines a computer is the **CPU** and **memory**. Without these a computer could not function. The CPU and memory work together to run programs.

CPU - executes programs using the **fetch-decode-execute cycle.**

Memory - stores program operations and data while a program is being executed. There are several types of memory, including: **registers, cache, RAM** and **virtual memory**.

Storage - stores programs and files long term, even when they are not in use. Devices such as hard drives, USB memory sticks or SD cards are used to store files such as photos, music and software applications long term.

An **input device** is any piece of computer hardware **used to provide data to a computer system.** Examples include: keyboard, mouse, scanner, digital camera and webcam.

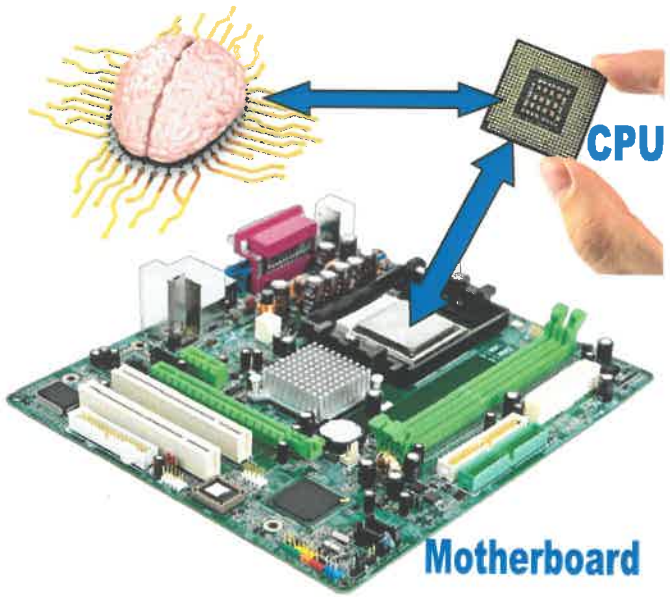
An **output device** is any piece of computer hardware used to communicate the results of data that has been processed.

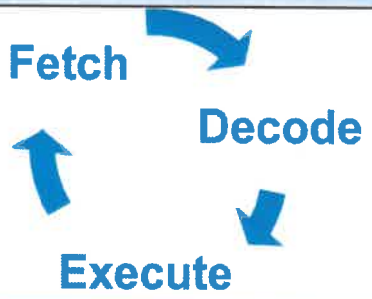
Central Processing Unit

The **Central Processing Unit** or **CPU** is arguably the most important component of a computer.

You can think of the CPU is being like the brain in a human.

It is responsible for all of a computer's processing.





The Fetch – Decode – Execute cycle

The **CPU** operates by repeating three operations:

FETCH – causes the next instruction and any data involved to be fetched from main memory

DECODE – decodes the instruction to make sure it can be carried out

EXECUTE – carries out the instruction

Repeat...



Key Vocabulary

Clock speed	The speed of a computer CPU, measured in hertz.
Cache	A piece of temporary memory. It can refer to a part of the RAM, storage disk, CPU, or an area for storing web pages.
CPU	**C**entral **P**rocessing **U**nit - the brains of the computer that processes program instructions. Also called a **microprocessor**.
Execute	To run a computer program.
GHz	Gigahertz. One billion hertz per second = one gigahertz. This is a measure of frequency and is used to describe bus speeds and CPU clock speeds.
Hardware	The physical parts of a computer system, e.g. a graphics card, hard disk drive and CD drive.
Motherboard	The circuit board inside a computer that houses the CPU, memory and connections to other devices.
RAM	Memory that is constantly being written to and read from. It does not retain its contents without a constant supply of power, i.e. when a computer is turned off, everything stored in its RAM is lost.
Registers	The section of high speed memory within the CPU that stores data to be processed.
Software	Software is the programs that run on a computer.
Virtual memory	A section of a computer storage drive which is temporarily used as RAM.

Binary Units

Remember the units used in the binary system.

1 byte =	8 bits
1 Kilobyte =	1024 bytes
1 Megabyte =	1024 Kilobytes
1 Gigabyte =	1024 Megabytes
1 Terabyte =	1024 Gigabytes

Knowledge Organiser: The CPU, FDE Cycle and Networks

Summary

CPU—The purpose of the CPU (Central Processing Unit) is to process data.

The CPU is made up of three main components. These are the **Control Unit**, the **Immediate Access Store** and the **Arithmetic and Logic Unit.**

Control Unit—This controls the flow of data within the system.

Immediate Access Store—This is where the CPU holds all the data and programs that are currently being used.

Arithmetic and Logic Unit — This is where the CPU performs the arithmetic and logic operations. So arithmetic calculations e.g. 5 + 2 = 7. The Logic part deals with logical comparisons, for example 5 > 3.

CPU

The speed of a Computer is very dependant on the CPU. There are several key factors which effect how fast the CPU can carry out instructions.

CLOCK SPEED

The CPU can only carry out one instruction at a time. Although to the user it may appear that the computer is carrying out multiple tasks at the same time, this is only because the CPU operates at an incredibly fast speed. The speed that the CPU carries out instructions is called the clock speed.

CORES

A standard CPU is made up of one core. However most modern CPUs have Dual, Quad or even Oct cores. The more cores a CPU has the faster it can operate. So a Dual core processor could fetch and execute two instructions at the same time, a quad core four instructions and an Oct core eight instructions at the same time.

CACHE

The Cache is a small block of memory built onto the processor. The Cache will store the most commonly used instructions and data. This speeds up processing instructions as the data is



FETCH-EXECUTE CYCLE

The Fetch—execute cycle is the basic operation of any computer . It simply means that an instruction is fetched from the memory and then it is executed. The instruction is fetched from the main memory or hard drive and then places it in a temporary memory called the immediate access store. Once this has happened the instruction will need to be decoded. Once the instruction has been decoded than it it will be executed by the CPU.

Key Vocabulary

ALU	Arithmetic Logic Unit. A component of the CPU that performs arithmetic and logic
Control Unit **CU**	The component of the CPU that manages instructions.
Data	Units of information, for example Boolean, integer, characters.
Execute	To run a computer program.
Hardware	The physical parts of a computer system
Immediate Access Store	Keeps the data that is required straight away by the Control Unit
Input	Data which is inserted into a computers system for processing or storage
Register	Part of the high speed memory within the CPU that stores data to be processed.

NETWORKS

A Network is defined as two or more computers joined together.

Some computers may be joined using a server, this is a central hub for all of the computers. The individual computers that are connected to the server are called clients.

If a computer is not part of a network it is called a stand alone computer.

Networks allow the sharing of hardware, for example printers, sharing of software across all computers on the network and files can be accessed by logging onto any computer connected to the Network.

The biggest Network in the world is called the internet.

Knowledge Organiser: Networks *Explore how data travels the world*

Summary

A **network** is created when more than one device is connected together.

A network can be a small collection of computers connected within a building (e.g. a school, business or home) or it can be a wide collection of computers connected around the world.

The main purpose of networking is to share data between computers.

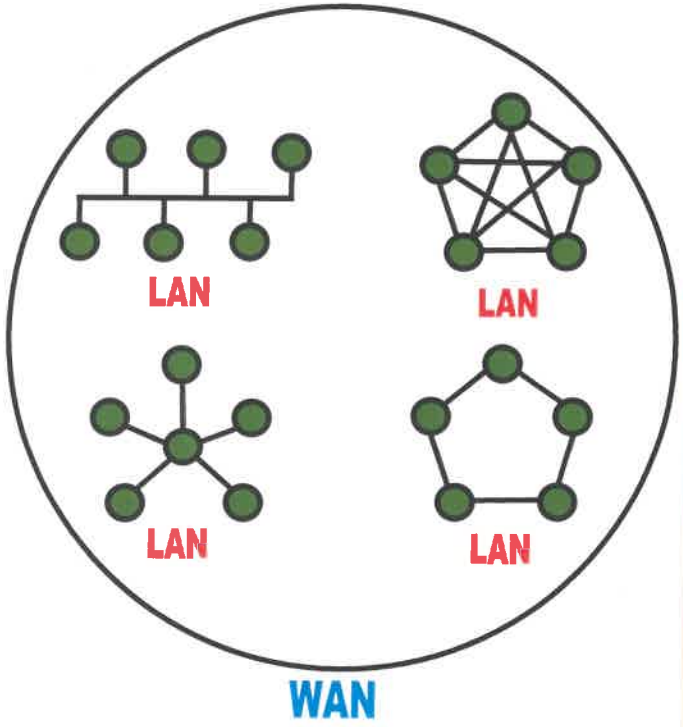
A file has to be broken up into small chunks of data known as **data packets** in order to be transmitted over a network. The data is then re-built once it reaches the destination computer.

Protocols are used to control how data is transmitted across networks. They are a set of rules for how messages are turned into data packets and sent across networks.

Bandwidth measures the amount of data that can transfer through a communications channel over a given period of time.

LANs and WANs

A school network is usually a **LAN.** LANs are often connected to WANs, for example a school network could be connected to the internet. **WAN**s can be connected together using the internet, leased lines or satellite links.



Advantages of networks

- Sharing devices such as printers saves money.
- Site (software) licences are likely to be cheaper than buying several standalone licences.
- Files can easily be shared between users.
- Network users can communicate by email and instant messenger.
- Security is good - users cannot see other users' files unlike on stand-alone machines.
- Data is easy to backup as all the data is stored on the file server.

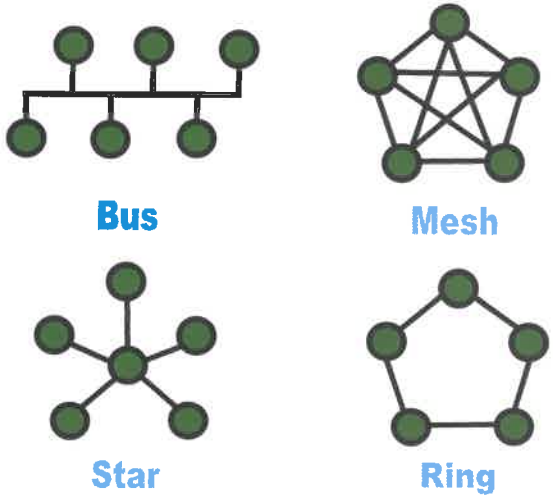
Disadvantages of networks

- Purchasing the network cabling and file servers can be expensive.
- Managing a large network is complicated, requires training and a network manager usually needs to be employed.
- If the file server breaks down the files on the file server become inaccessible. Email might still work if it is on a separate server. The computers can still be used but are isolated.
- Viruses can spread to other computers throughout a computer network.
- There is a danger of hacking, particularly with wide area networks. Security procedures are needed to prevent such abuse, e.g. a firewall.

Key Vocabulary

File server	A networked **computer** that provides shared storage, it can be accessed by workstations on the same network.
Input device	Input devices, like a keyboard, allow us to put raw data in a computer which it processes to produce outputs.
LAN	**L**ocal **A**rea **N**etwork covers a small area such as one site or building, e.g. a school or a college.
Licence	A legal agreement between the company who published the software and the end user covering areas such as copyright.
Network	A network is a number of computers linked together to allow the sharing of resources.
Output device	A device used to output data or information from a computer, e.g. a monitor or printer.
Server	A computer that holds data to be shared with other computers. A web server stores and shares websites.
WAN	**W**ide **A**rea **N**etwork covers a large geographical area. Most WANs are made from several LANs connected together.
Workstation	A computer connected to a network.

Topologies



Knowledge Organiser: Internet communication

Summary

The **internet** has revolutionised the way we work and play. It allows us to communicate, to share data and to seek information in a matter of seconds. All this is possible through the use of computers and networks.

The internet is a global **network** of computers. All computer devices that are connected to the internet form part of this network. Added together, there are billions of computers connected to the internet, all able to communicate



World Wide Web

The internet is a global **network** of computers. The World Wide Web is the part of the internet that can be accessed through **websites**.

Websites are accessed using a **web browser**. A browser is a program designed to display the information held on a website. Every website has an address at which it can be found, a bit like a house address.

A website's address is known as its **URL**. A website can be visited by typing its URL into a web browser. Each address contains the prefix 'http:' which tells the computer to use the **hyper text transfer protocol** for communicating with the website.

Transferring information via the internet

The internet is a global network of computers, some of which are called **web servers**. A web server is a computer which holds websites for other computers linked to the internet to access.

Sending information to a web server is known as **uploading**. Receiving information from a web server is known as **downloading**.

When information is uploaded to, or downloaded from, a web server it is broken up into tiny pieces called **data packets**. Each packet is a very short communication between the **client** computer and the web server.

HTML

All web pages on the internet are created using a language called **Hypertext Markup Language** (HTML).

This example HTML code displays a message on a webpage:

```
<html>
    <body>
        <h1>Hello world</h1>
        <p>This is my first webpage</p>
    </body>
</html>
```

Using Email to communicate

Two of the main ways in which people use the internet for communication are **email** and **VoIP**.

Email (short for electronic mail) is the digital equivalent of sending a letter.

A **video conference** is live video streamed over the internet so that people can communicate face. **Voice over Internet Protocol (VoIP)** is the technology that allows us to video conference. Many companies offer free VoIP services, including Skype, Apple Facetime and

Key Vocabulary

Bandwidth	Bandwidth measures the amount of data that can transfer through a communications channel over a given period of time.
Browser	An application used to view web pages, eg Internet Explorer or Google Chrome.
Client	The computer on a network that request data from a server.
Data Packet	A piece of data sent over a network. Messages have to be broken down into binary data packets before they are transferred.
Downloading	To copy a file from the internet onto your computer or device.
Email	Electronic mail. A method for sending messages and files to other people.
HTML	HyperText Markup Language. The language used to write and display web page documents.
HTTP	Hypertext Transfer Protocol - a request/response standard. Web browsers send requests and websites or servers respond to requests.
Hyperlinks	A link in a document or webpage that connects to another location.
Internet	A global network connecting millions of computers.
Network	A group of interconnected computers/devices.
Online	Connected to and using the internet.
Streaming	Data that is sent in pieces. Each piece is viewed as it arrives, eg a streaming video is watched as it downloads.
URL	Uniform resource locator - Each web page address on a network is written as a URL. It contains the IP address and the DNS and web page details.

Some images is copyright © 2016 BBC.

Knowledge Organiser: Mobile App Development

Graphical User Interface (GUI)

- **Visual interface made up of graphics**
- **Users can navigate around the user interface logically**
- **Contains Icons to select programs**



WYSIWYG

What you see is what you get



- Users can see exactly that way a documents looks on screen.
- This is used for word processing, Web development and Mobile App development software.

Native Apps

- Native Apps are programmed for and installed on a specific mobile platform.
- They are downloaded to and stored on the Mobile phone
- They can make full use of the Mobile Phone features, for example GPS location, camera, speakers, contact lists, microphone, sensors..

Web Apps

- These are used from the internet
- They are not downloaded onto the Mobile Phone
- No Application data is stored on the device
- The functionality is limited as the phones features, camera, GPS and sensors can't be used.

Vocabulary

- APP — Software application for mobile devices
- GPS—Global Positioning System
- Icons—Small computer graphics
- Intuitive—Means easy to understand
- Menu—Presents the user with a list of options
- Haptic—Relates to a sense of touch
- Platform—The name given to the computer (hardware) and operating system (software) on which applications can be run.
- Triangulation—Establishing mobile location from three or more transmitters
- URL—Uniform Resource Locator—This is the address of the webpage.



Map features

- Mobile phone Apps often use location data to enhance the functionality of the App.
- GPS—Global positioning system.—this is very accurate and can pin point a device to one metre.
- Triangulation is used from three or ,ore cell towers (where the transmitter is located) so that the location of the phone can be found.



Advantages

- Show your position on a map
- Give turn by turn directions
- Provide local suggestions is search results
- Help you to find a lost device
- TAG where a photo was taken.

Disadvantages

- Loss of privacy with a persons location always known
- Users may not wish to be 'Tracked'
- Tagged images can reveal your location to others
- The battery on the mobile phone will drain more quickly with GPS data being constantly updated

Knowledge Organiser: HTML and Website Development *Design and code a modern website*

Summary

Hyper Text Markup Language (HTML) is a basic programming language for building web pages. It uses a set of **predefined tags** that the **web browser** then interprets and renders/displays.

The World Wide Web Consortium (W3C) is responsible for standardising HTML and releasing updated specifications that revise existing tags, introduce new tags and identify deprecated tags and should be the first reference point for any matters concerning HTML.

Common browsers include: Internet Explorer, Firefox, Safari, Opera and Chrome.

Using HTML to create websites

HTML can be written in specialist software, or in a simple text editor like Notepad. As long as the document is saved with the **file extension**'.html' it can be opened and viewed as a webpage from a browser. This example HTML code displays a message on a webpage:

```
<html>
    <body>
        <h1>Hello world</h1>
        <p>This is my first webpage</p>
    </body>
</html>
```

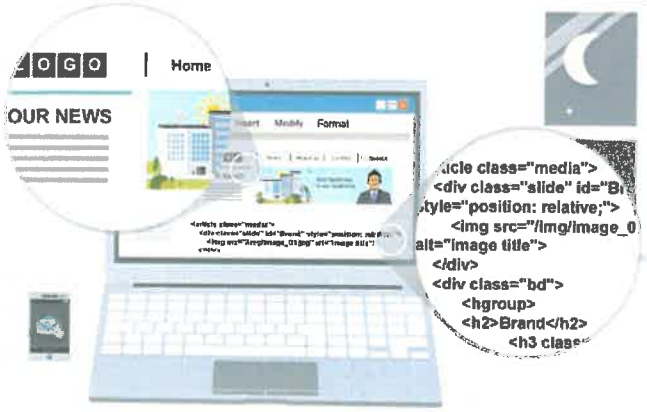
The code uses tags to describe the appearance of the information:

<html> states that the document is a HTML document

<body> states that the information appears in the body of the page

<h1> states that the following text appears as a prominent heading

<p> states that this is the beginning of a new paragraph



Some images is copyright © 2016 BBC.

Working with HTML

HTML is typically written (or generated) in two ways:

- using a plain text editor, e.g. Notepad, Notepad++ or TextPad.
- using a **What You See Is What You Get (WYSIWYG)** editor, e.g. Dreamweaver, iWeb or SeaMonkey.

Plain text editor

An editor like Notepad offers greater control over the code when compared to a WYSIWYG editor because each and every character that forms the HTML and the resulting web page is hand typed. The disadvantage is it's a slow process.

WYSIWYG editor

The web page is designed, and the content written and styled, using a series of tools. The page can be previewed in the default browser from within the **application**. This makes building a web page much faster because only a limited knowledge of HTML is needed as it's automatically generated.

Versions of HTML

Over the years there have been several versions of HTML. Each successive version has more tags, allowing the programmer to build more advanced webpages. The latest version, released in 2012, is HTML5.

Key Vocabulary

Browser	An application used to view web pages, e.g. Internet Explorer or Google Chrome.
Hyper Text Markup Language (HTML)	Hyper Text Markup Language. The language used to write and display web page documents.
Hyperlink	A link in a document or webpage that connects to another location.
Internet	A global network connecting millions of computers.
Web browser	An application that displays web pages.
Web Page	A page designed for, and viewed in, a web browser.
Website	A web page or group of web pages hosted on one web server and viewed in a web browser, usually maintained by a person, group or organisation.

?

What happens when I view a web page?

What is WWW?

Where are all the websites stored?



Websites are stored on web servers all over the world.

Web pages contain different types of information including images, text and multimedia.

Each website contains web pages which are navigated to via hyperlinks.

There is no central storage of websites.

The World Wide Web is a huge collection of websites that we can access using the Internet.

Knowledge Organiser: Programming — Iteration

Summary

Iteration is the process of repeating steps. Iteration allows us to simplify algorithms by repeating certain steps . This makes the program run faster, is easier to code and the program works more efficiently. Iteration allows us to simplify our algorithm by stating that we will repeat certain steps until told otherwise. The program is then simpler because it doesn't have to include lots of unnecessary steps. This is implemented in programming using **FOR** and **WHILE** statements.

FOR and **WHILE** statements are used in programming for iteration, they are also known as **loops,** because when a program iterates it 'loops' to an earlier step.

Count Controlled Loops—These use the FOR loop to control how many times the program will loop.

Condition Controlled Loops. These use a condition to **control how many times the program loops**. When the condition is met the program stops looping.

Condition Controlled Iteration

A condition—controlled loop continues looping until a condition is met. A condition controlled loop is used when there is no way of knowing how many times the iteration or LOOP will need to occur.



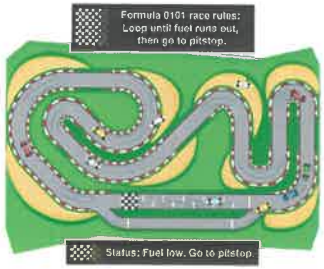
These types of loop are called **WHILE LOOPS.**

A Condition Controlled or WHILE LOOP will keep looping or iterating until the desired condition is met. E.g True or False.

Count Controlled Iteration

Count Controlled Iteration is used when the number of times that the program need to loop is known . The program will then count how many times the code has looped using a counter coded into the program until it reaches its set limit.

These are called **FOR LOOPS**



Condition Controlled uses WHILE Loops

Example code for a Condition Controlled Loop (WHILE LOOP)

```
password = "blank"
WHILE  password does not equal "iloveCode"
      INPUT password
OUTPUT  "Password correct"
```

Steps that are part of the loop are indented. Indentation

Count Controlled uses FOR Loops

Example code for a Count Condition Controlled Loop (FOR LOOP)

```
total = 0
for count in range(5):
        number = int(input("Type in a number: "))
        total = total + number
print("The total is: ")
print(total)
```

Steps that are part of the loop are indented. Indenta-

Key Vocabulary

Algorithm	A sequence of logical instructions for carrying out a task. In computing, algorithms are needed to design computer programs.
Condition	In computing, this is a statement or sum that is either true or false. A computation depends on whether a condition equates to true or false.
Counter	A device which stores (and/or displays) the number of times a particular event or process has occurred.
Implement	To put into effect.
Indenting	Adding spaces or tabs in front of blocks of code, making it easier for programmers to see which parts of the code relate to each other.
Instruction	A single action that can be performed by a computer processor.
Iteration	In computer programming, this is a single pass through a set of instructions.
Program	Sequences of instructions for a computer.
Programming	The process of writing computer software.
Pseudocode	Also written as pseudo-code. A method of writing up a set of instructions for a computer program using plain English. This is a good way
Python	A high-level programming language.
Selection	A decision within a computer program when the program decides to move on based on the results of an event.
Sequencing	Creating a set of instructions to complete a task.
Statement	The smallest element of a programming language which expresses an action to be carried out.

When designing programs, there may be some instructions that need repeating. This is known as iteration, and is implemented in programming using FOR and WHILE statements.

Some images is copyright © 2016 BBC.

Knowledge Organiser: Programming—Boolean Logic

What is Boolean Logic

Programs use simple comparisons to help make decisions. **Boolean logic** is a form of algebra where all **values** are either True or False. These values of true and false are used to test

the **conditions** that **selection** and **iteration** are based around.

Boolean logic uses algebra and algebraic expressions. We use these expressions in algorithms and programs.

Expression and Boolean Equivalent

Equals	=
Greater than	>
Less than	<
Greater than or equal to	>=
Less than or equal to	<=
Does not equal	<>

Boolean Expressions use Algebra

Boolean expressions are represented using **Algebra.**

E.g. 5 < 10

X < 10

X < y

Statements 1 uses numbers, whereas statement2 and 3 use variables.

Statement	Expression	Boolean value
y > x	y is greater than x	**True.** When x is 5 and y is 10, then y is greater than x.
x < y	x is less than y	**True.** When x is 5 and y is 10, then x is less than y.
x = y	x equals y	**False.** When x is 5 and y is 10, then x does not equal y.
x<>y	x does not equal y	**True.** When x is 5 and y is 10, then x does not equal y.

Boolean Expressions — AND, OR and NOT

As well as using simple comparison, Boolean logic can also use the **expressions AND, OR** and **NOT**:

- AND considers two (or more) conditions. The result is **True** if the result of **all** comparisons is **True**.
- OR considers two (or more) conditions. The result is **True** if **either** comparison is **True**.

NOT requires a condition to be False.

Using AND in Python

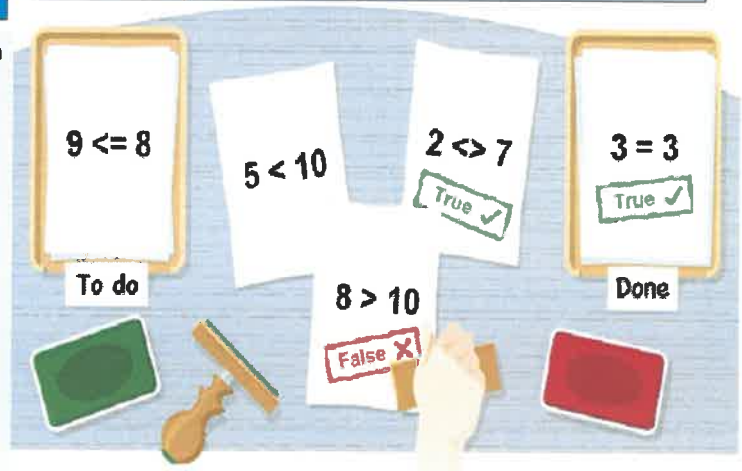
Sometimes we need a program to do something based on the results of two conditions. This simple program prints out a different message depending on the result of two tests.

If each test's score is greater than or equal to 5, then a message is output saying "You have passed!". Otherwise, a message is output saying "Sorry, try again.".

```
score1 = int(input("What did you score in test 1?"))
score2 = int(input("What did you score in test 2?"))
if score1 >= 5 and score2 >= 5:
print("You have passed!")
else:
print("Sorry, try again.")
```

Key Vocabulary

Boolean logic	A form of logical algebra which works only with two values, true or false.
Condition	In computing, this is a statement or sum that is either true or false. A computation depends on whether a condition equates to true or false.
Expression	A combination of symbols representing a value. Expressions are used in programming languages and applications.
Iteration	In computer programming, this is a single pass through a set of instructions.
Program	Sequences of instructions for a computer.
Python	A high-level programming language.
Selection	A decision within a computer program when the program decides to move on based on the results of an event.
Statement	The smallest element of a programming language which expresses an action to be carried out.
Value	A numerical amount denoted by a specific term, eg the value of x is 10.
Variable	A memory location within a computer program where values are stored.



Knowledge Organiser: Programming — Arrays and lists

Arrays and Lists

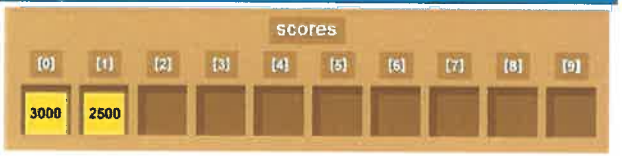
Summary

When writing programs, it is useful to use **arrays** and **lists** as they simplify programs by storing related data under one name. Programs can get very complex without using **arrays.** An Array will reduce the amount of code required to perform a complex task making the program easier to write and more efficient.

An Array is a series of memory locations or 'boxes' (think of a **box** in as a variable in which to hold data). Each **box** will hold a single item of data, but with each box sharing the same name. All the data in an **Array** must be of the same Data Type.

Lists are Data structures similar to arrays that allow data of more than one data type. Some programming languages such as BASIC and Java allow the use of arrays. Others such as Python do not and only allow lists.

Arrays use elements



Each box in the array is referred to as an element. In the example above you can see an Array is similar to a row of cells. The individual elements in the array are numbers 0 to 9 because computers start counting at 0.

Arrays are named like variables. The number in brackets determines how many data items the array can hold. So the Array score (9) could hold 10 items of data to be

Using Arrays

To store a **data** item in an **array**, the **element** that the data will be stored in needs to be referenced.

For example, in the score array **score(10)**, the following would store the score 3000 in the first element:

score[0] = 3000

The following would store the score 2500 in the second element:

score[1] = 2500

Array code

score_0
score_1
score_2
score_3
score_4
score_5
score_6
score_7
score_8
score_9

Example code for an Array of scores (for the example above) without using an Array.

score(9)

Simplified code

Why use Arrays

A **variable** holds a single item of data. There may be a situation where lots of variables are needed to hold similar and related data. In this situation, using an array can simplify a program by storing all related data under one name.

This means that a program can be written to search through an array of data much more quickly than having to write a new line of code for every variable. This reduces the complexity and length of the program which makes it easier to find and debug errors.

Key Vocabulary

Array	A set of data values of the same type, stored in a sequence in a computer program. Also known as a list.
BASIC	A group of general-purpose, high-level programming languages that are relatively easy to use. BASIC stands for Beginner's All-purpose Symbolic Instruction Code.
Data	Units of information. In computing there can be different data types, including integers, characters and Boolean. Data is often acted on by instructions.
Data Structure	The way that data is stored in a database or program.
Data type	The format in which a variable or constant holds data, such as 'integer' or 'string'.
Element	An individual component of an array.
Java	A popular high-level computer programming language.
Python	A high-level programming language.
Variable	A memory location within a computer program where values are stored.

Some images is copyright © 2016 BBC.

Knowledge Organiser: Searching Algorithms

Designing an Algorithm

An algorithm is a plan, a logical step-by-step process for solving a problem. Algorithms are normally written as a flowchart or in pseudocode.

The key to any problem-solving task is to guide your thought process. The most useful thing to do is keep asking 'What if we did it this way?' Exploring different ways of solving a problem can help to find the best way to solve it.

Why do we need Searching Algorithms

Searching for data can be very difficult. Searching algorithms, such as serial search and binary search, make the process of searching for data much easier.

We often need to find one particular item of **data** amongst many hundreds, thousands, millions or more. This is why searching **algorithms** are important. Without them you would have to look at each item of data, individually. In a large set of data, it will take a long time to do this. Instead, a searching algorithm can be used to help find the item of data you are looking for.

Key Vocabulary

Serial Search	A simple method of searching, in which the search moves from one item to the next in sequence, until either a match is found or end of the data is reached with no match found.
Binary Search	A method of searching in which the data being searched is halved with every step.
Criteria	Singular - criterion. A set of rules or conditions that must be met. Often used in searches.
Flowchart	A diagram that shows a process, made up of boxes representing steps, decision, inputs and outputs.
Keyword	An important word that will help return accurate search results.
Sorted	When things are put into order, eg from largest to smallest.

Serial Search

Searching for a keyword or value is the foundation of many computer programs. The most basic kind of search is a serial search.

Criteria are set up before the search begins. The search then starts with the first item and then moves to each item in turn, until either a match is found or it reaches the end of the data with no match found.

Binary Search

Binary search is a faster method for searching for an item that is in an ordered list.

An ordered list is one where the sequence of items in the list is important. An ordered list does not necessarily contain a sequence of numbers (e.g. 1, 2, 3, 4) or characters (eg A, B, C, D). It might also contain, e.g. a list of names in alphabetical order, a list of files from smallest to largest or a list of records from earliest to most recent.

Positives and Negatives of a Serial Search

One of the main advantages of a serial search is that it is a very simple algorithm, which makes it very easy to write a computer program to carry it out. It can also be used on any set of data regardless of type and whether or not it is sorted.

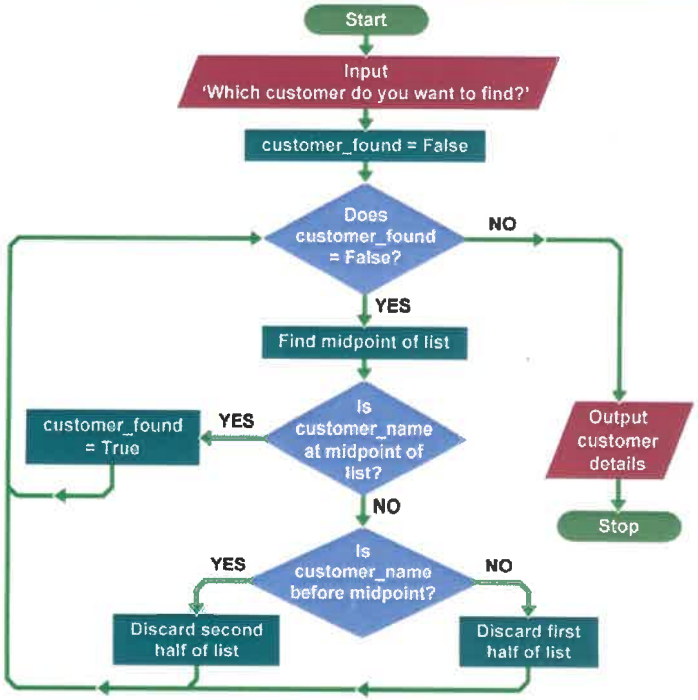
The biggest problem with a serial search is that it is very slow. For example, when searching through a database of everyone in the UK to find a particular name, it might be necessary to search through 60 million names before you found the one you wanted.

Positives and Negatives of a Binary Search

One of the main advantages of a binary search is that it is much quicker than a serial search because the data that needs to be searched halves with each step.

For example, it is possible to search through 1024 values and find the one you want within 10 steps, every time.

The biggest problem with a binary search is that you can only use this if the data is sorted into an order.



Knowledge Organiser: Sorting Algorithms

Why do we need Sorting Algorithms

A sorting algorithm will put items in a list into an order, such as alphabetical or numerical order. For example, a list of customer names could be sorted into alphabetical order by surname, or a list of people could be put into numerical order by age.

Sorting a list of items can take a long time, especially if it is a large list. A computer program can be created to do this, making sorting a list of data much easier.

Key Vocabulary

Bubble sort	A sorting algorithm that repeatedly passes through a list to be sorted, comparing and swapping items that are in the wrong order.
Algorithm	A sequence of logical instructions for carrying out a task. In computing, algorithms are needed to design computer programs.
Bucket sort	A sorting algorithm that separates data into different collections, called buckets, which are filled with data and then sorted.
Data	Units of information. In computing there can be different data types, including integers, characters and Boolean. Data is often acted on by instructions.
Program	Sequences of instructions for a computer.

Bucket Sort

A **bucket sort algorithm** separates a list of data into different collections of data, called 'buckets'. Empty buckets are set up at the start of the sort and are filled with the relevant data. Each bucket is then sorted, and the data is finally gathered back into a list.

Bubble Sort

A bubble sort **algorithm** goes through a list of **data** a number of times, comparing two items that are side by side to see which is out of order. It will keep going through the list of data until all the data is sorted into order. Each time the algorithm goes through the list it is called a 'pass'.

If the data being sorted is a large set of data, it may take



Positives and Negatives of a Bucket Sort

One of the main **advantages** of a bucket sort is that is quicker to run than a bubble sort. Putting data into small buckets that can be sorted individually reduces the number of comparisons that need to be carried out.

The **biggest problem** with a bucket sort is that the algorithm is a bit more complicated than the bubble sort to describe for a computer. It might be easy to decide to sort numbers in groups of 10, but what about a list of names? Should there be a bucket for each letter of the alphabet, or perhaps one bucket for all of the X, Y and Z

Positives and Negatives of a Bubble Sort

One of the main **advantages** of a bubble sort is that it is a very simple algorithm to describe to a computer. There is only really one task to perform (compare two values and, if needed, swap them). This makes for a very small and simple computer program.

The **biggest problem** with a bubble sort is that it takes a very long time to run. For example, if there are 100 values to sort, each pass through the list will take 99 comparisons – and you might have to repeat it 99 times.

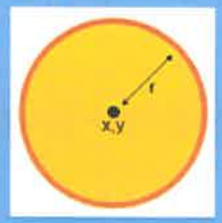




Knowledge Organiser: Bitmap and Vector Graphics

Properties of a Vector Graphic

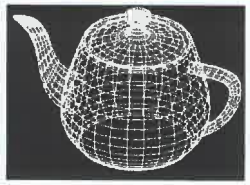
- **Vector** images comprise geometric shapes
- The properties of each shape is recorded as:
- **Shape type** (e.g. Circle)
- **Radius**



Features of a Vector Graphic

- Vector graphics **consist of shapes called objects.**
- You **can edit each object separately** to change the shape, colour, size and position.
- Recording the properties of vector shapes doesn't take up much memory. This usually makes the **file**

Uses of Vector Graphics



- Geometric logos
- Gradients, Fills, Shapes
- Grouping Objects
- Logos
- Animation
- Sign making
- Used for cartoon imagery
- General graphic design
- Computer Aided Design
- Architectural Plans



Some images is copyright © 2016 BBC.

Properties of a Bitmap Graphic

- A **Bitmap** graphic is also known as a **Raster** graphic
- A bitmap graphic is composed of tiny squares, called **pixels**
- It is possible to **edit each individual pixel.**
- The computer stores information about the **colour value of every pixel** in the image.
- This can mean a **large file size**
- Resizing a Bitmap graphic causes it to **pixelate** and **lose quality**
- Images are made up of **Pixels** (***Picture Elements***)
- Each pixel is set to one colour
- Together they look like an image

Bitmap and Vector Comparison

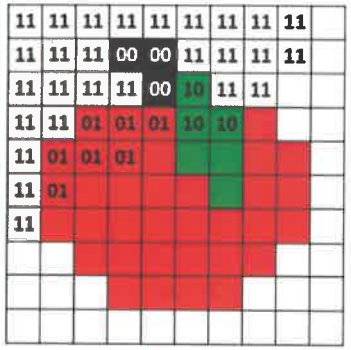
	Bitmap Graphic	Vector Graphic
Made up of	**Coloured pixels**	**Objects**
File size	**Large**	**Small**
Resizing	**Lose quality**	**No loss of quality**
Appearance	**Real**	**Cartoon like**
File Formats	**.bmp .jpg .gif**	**.svg .wmf**

Creating an Image

- Each Pixel is given a binary value
- Each value represents a different colour
- Using one bit per pixel allows only 2 values, 0 and 1 -1 = White, 0 = Black
- More bits per pixel = more colour combinations
- 1 bit = 2 Colours
- **2 bits = 4 Colours**
- 3 bits = 8 Colours
- 4 bits = 16 Colour

Bits and Bytes

1	1	1	1	1	1	1	1	1	1
1	1	1	1	0	0	1	1	1	1
1	1	1	0	0	0	0	1	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	0	0	0	0	0	1	1
1	1	0	0	1	1	0	0	1	1
1	1	0	0	1	1	0	0	1	1
1	1	1	1	1	1	1	1	1	1



Knowledge Organiser: Data Representation

Summary—Data Representation

Computers use electrical signals that are on or off, so they have to see everything as a series of **binary** numbers. This data is represented as a sequence of 1s and 0s (on and off). All data that we want a computer to process needs to be converted into this binary format.

Computers use binary to process data. There are simple techniques to convert between binary and denary and to add two binary numbers together.

How do Computers see the world

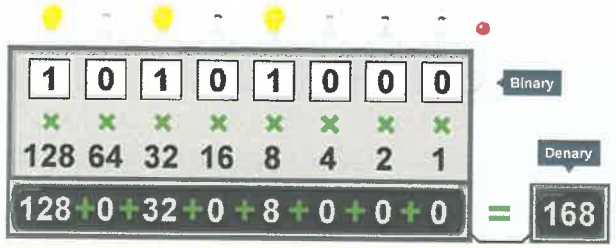
Computers use electrical signals that are on or off, so they have to see everything as a series of **binary** numbers. This data is represented as a sequence of 1s and 0s (on and off). **All data that we want a computer to process needs to be converted into this binary format.**

What is Binary

Binary is a number system that only uses two digits: 1 and 0. All information that is processed by a computer is in the form of a sequence of 1s and 0s. Therefore, all data that we want a computer to process needs to be converted into binary.

The binary system is known as a 'base 2' system. This is because:

- there are only two digits to select from (1 and 0) when using the binary system, data is converted using the power of two.



Alan Turing



Turing was quite a brilliant mathematician, most famous for his work on breaking the German Enigma codes... Without his outstanding contribution the history of the Second World War could have been very different.

Turing was highly influential in the development of theoretical computer science, providing a formalisation of the concepts of algorithm and computation with the Turing machine, which can be considered a model of a general-purpose computer. Turing is widely considered to be the father of theoretical computer science and artificial intelligence.

What is Denary

Thousands	Hundreds	Tens	Units
(1000s)	(100s)	(10s)	(1s)

People use the **denary** (or decimal) number system in their day-to-day lives. This system has 10 digits that we can use: 0, 1, 2, 3, 4, 5, 6, 7, 8 and 9.

Binary to Denary Conversion

To convert a **binary** number to denary, start by writing out the binary place values.

128	64	32	16	8	4	2	1
1	0	1	0	1	0	0	0

Example the binary value of **10101000 = 168 in Denary**

1 x 128
0 x 64
1 x 32
0 x 16
1 x 8

Binary to Denary Conversion

Example 199 **Denary**

199 – 128 = 71. Because 128 could be taken off, put a 1 in the '128' place value column:

128	64	32	16	8	4	2	1
1							

Now repeat for 64: **71 – 64 = 7**

128	64	32	16	8	4	2	1
1	1						

Again for 32: **7 – 32** won't work, so put a 0 in that place value column.

128	64	32	16	8	4	2	1
1	1	0					

Try again for 16: **7 – 16** won't work, so add a 0 to that place value column

128	64	32	16	8	4	2	1
1	1	0	0				

Next is 8: **7 – 8** won't work. Add a 0 to the '8' place value column

128	64	32	16	8	4	2	1
1	1	0	0	0			

Try again for 4: **7 – 4 = 3**, so add a 1 to the '4' place value column.

128	64	32	16	8	4	2	1
1	1	0	0	0	1		

Next try 2: **3 – 2 = 1**, so add a 1 to the '2' place value column.

128	64	32	16	8	4	2	1
1	1	0	0	0	1	1	

And finally, 1: **1 – 1 = 0** – add a 1 to the '1' place value column.

128	64	32	16	8	4	2	1
1	1	0	0	0	1	1	1

This means that 199 as a binary number is 1100 0111.

Knowledge Organiser: Representing Text Images and Sound

Representing Data

All data inside a computer is transmitted as a series of electrical signals that are either **on** or **off**. Therefore, in order for a computer to be able to process any kind of data, including text, images and sound, they must be converted into **binary** form. If the data is not converted into binary – a series of 1s and 0s – the computer will simply not understand it or be able to process it.

Representing Text

A code where each number represents a character can be used to convert text into binary. One code we can use for this is called **ASCII**.

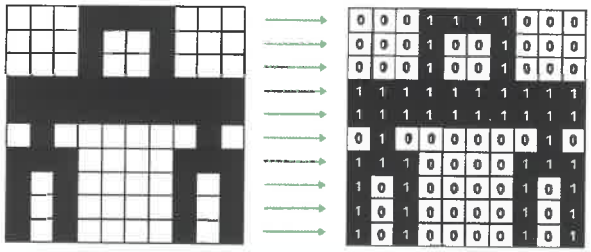
ASCII code can only store 128 characters, which is enough for most words in English but not enough for other languages.
If you want to use accents in European languages or larger alphabets such as Cyrillic (the Russian alphabet) and Chinese Mandarin then more characters are needed.

Therefore another code, called **Unicode**, was created. This meant that computers could be used by people using

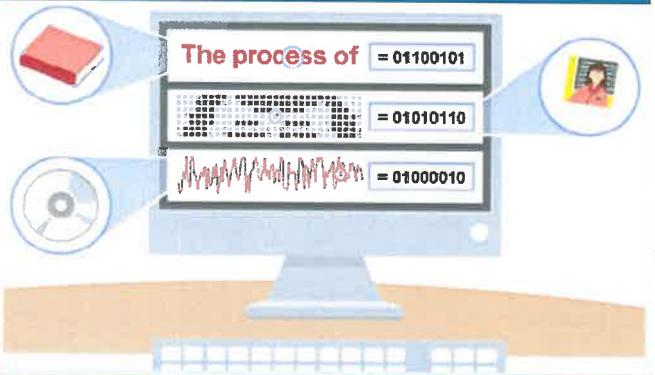
Representing Images—Black and White

Images also need to be converted into **binary** in order for a computer to process them so that they can be seen on our screen. Digital images are made up of **pixels**. Each pixel in an image is made up of binary numbers.

If we say that 1 is black (or on) and 0 is white (or off), then a simple black and white picture can be created using binary.



Text Images and Sound



Representing Images—Colour

Adding another binary digit will double the number of colours that are available:

- 1 bit per pixel (0 or 1): two possible colours
- 2 bits per pixel (00 to 11): four possible colours
- 3 bits per pixel (000 to 111): eight possible colours
- 4 bits per pixel (0000 – 1111): 16 possible colours
- ...

16 bits per pixel (0000 0000 0000 0000 – 1111 1111 1111 1111): over 65 000 possible colours

The number of bits used to store each pixel is called the **colour depth**. Images with more colours need more pixels to store each available colour. This means that images that use lots of colours are stored in larger files.

Image Quality

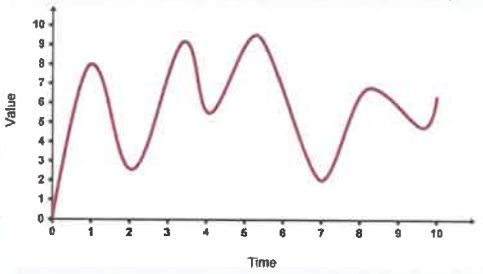
Image quality is affected by the **resolution** of the image. The resolution of an image is a way of describing how tightly packed the pixels are.

In a low-resolution image, the pixels are larger so fewer are needed to fill the space. This results in images that look blocky or **pixelated**. An image with a high resolution has more pixels, so it looks a lot better when you zoom in or stretch it. The downside of having more pixels is that the file

Representing Sound

Sound needs to be converted into **binary** for computers to be able to process it. To do this, sound is captured - usually by a microphone - and then converted into a **digital** signal.

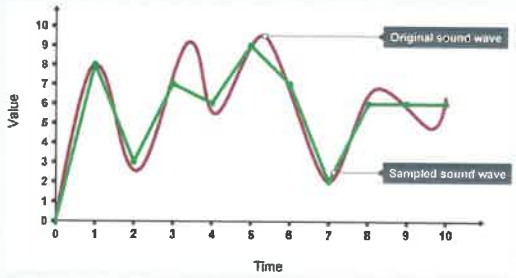
An **analogue** to digital converter will sample a sound wave at regular time intervals. For example, a sound wave like this can be sampled at each time sample point:



The samples can then be converted to binary. They will be recorded to the nearest whole number.

If the time samples are then plotted back onto the same graph, it can be seen that the sound wave now looks different. This is because sampling does not take into account what the sound wave is doing in between each time sample.

Sound loses quality as data has been lost between the time samples



The way to increase the quality is to have more time samples that are closer together.
The frequency at which samples are taken is called the **sample rate**, and is measured in Hertz (Hz).

Compression

Lossy compression
Lossy compression removes some of a file's original data in order to reduce the file size.

Lossless compression
Lossless compression doesn't reduce the quality of the file at all. There are various algorithms for doing this. The space savings of lossless compression are not as good as they are with **lossy** compression.

Knowledge Organiser: Memory and Storage

Primary Memory

Primary memory holds data and programs that are currently in use. **Memory** is the component of the computer that holds **data**, **programs** and **instructions** that are currently in use. Primary memory is built inside the computer. This gives the **processor** fast access to the data and

ROM—Read Only Memory

Read only memory (ROM) is **non-volatile** primary **memory**. Its contents are **not lost** when the computer is turned off. ROM can be read from, but not written to, hence the term 'read only'. This makes ROM ideal for storing **instructions** and **data** that are needed for the computer to run. The **BIOS (Basic Input Output System)** is an example of a program that is stored

RAM—Random Access Memory

Random access memory (RAM) is **volatile** primary memory. Once the computer is switched off, the **data** and **instructions** held in RAM are lost. RAM is used to hold data and instructions that are currently in use. In a modern PC, RAM is used to hold the **operating system** and any open documents and **programs** that are running.

The more RAM a computer has, the more data and programs it can hold simultaneously. RAM can also be upgraded easily, unlike other types of primary memory.

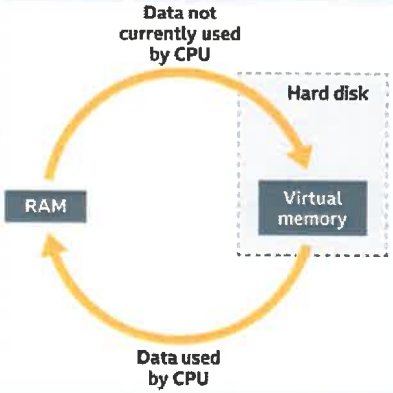
Cache Memory

Cache memory is a type of high-speed **random access memory (RAM)** which is built into the **processor**. **Data** can be transferred to and from cache memory more quickly than from RAM. This allows for faster processing as the processor does not have to wait for the data and instructions to be fetched from RAM. The more cache memory a computer has, the faster it runs.

Memory



Virtual Memory



Virtual memory is used when the computer has no more available random access memory (RAM).

When this happens, part of the computer's **secondary storage**, such as the **hard disk**, can be used to store data temporarily.

Virtual memory enables data that is in RAM and not currently being used to be transferred to the hard disk. This frees up room in RAM for other programs and data. This process is known as **swapping**.

Using virtual memory makes a computer run slower, as the processor has to wait while data is swapped between hard disk and RAM.

Glossary

BIOS	Basic input output system - the basic firmware that is embedded in the computer ROM chip used to start a computer.
Cache	A piece of temporary memory. It can refer to a part of the RAM, storage disk, CPU, or an area for storing web pages.
Flash	Non-volatile memory that can be read from and written to. It is suitable for secondary storage.
Non Volatile Memory	A form of computer memory that stores data even when not powered.

Flash Memory

Flash memory is non-volatile memory than can be read from and written to. **Flash memory** is a special type of memory. It can be written to and overwritten just like **random access memory (RAM)**. However, unlike RAM, it is **non-volatile**, which means that when the computer's power is switched off, flash memory will retain its contents.

Flash memory is fast to access and write to, although not as fast as when reading from or writing to RAM. It requires little power and contains no moving parts, making it an ideal storage medium for many modern devices, such as tablets, smartphones and digital cameras.

Flash memory is also used as external **secondary storage**, for example in USB memory sticks and solid-state drives.



Knowledge Organiser: Storage

Storage—Volatile and Non– Volatile

Computers use primary **memory** such as random access memory **(RAM)** and **cache** to hold **data** that is being processed. However, this type of memory is **volatile**, which means it loses its contents when the computer is switched off. Secondary storage is needed to keep programs and data indefinitely. **Secondary storage** is **non-volatile**, long-term storage. It is used to keep programs and data indefinitely.

Magnetic Storage

Magnetic devices such as **hard disk drives** use magnetic fields to magnetise tiny individual sections of a metal spinning disk. Each tiny section represents one **bit**. A magnetised section represents a **binary** '1' and a demagnetised section represents a binary '0'. These sections are so tiny that disks can contain **terabytes (TB)** of data. As the disk is spinning, a read/write head moves across its surface.

Optical Devices

Optical devices use a **laser** to scan the surface of a spinning disc made from metal and plastic. The disc surface is divided into tracks, with each track containing many flat areas and hollows. The flat areas are known as lands and the hollows as pits.

When the laser shines on the disc surface, lands reflect the light back, whereas pits scatter the laser beam. A sensor looks for the reflected light.



Reflected light (lands) represents a binary '1', and no reflection (pits) represents a binary '0'.

Storage



Solid State Devices

Solid state devices use **non-volatile** random access memory **(RAM)** to store data indefinitely. They tend to have much faster access times than other types of device and, because they have no moving parts, are more durable. This sort of memory is expensive, solid state devices tend to be smaller in capacity than other types.



For example, a solid-state drive that holds 256 **GB** might be of a similar cost to a hard disk with several terabytes capacity. They also consume little power and durable and portable.

Factors to Consider for suitable storage media

- Cost - what is the cost per gigabyte (GB)?
- Capacity - how much data can the medium hold?
- Speed of access - how quickly can data be transferred to and from the medium?
- Portability - how portable is the medium? Does it need to be portable?

Glossary

Key Vocabulary

Binary	A number system that contains two symbols, 0 and 1. Also known as base 2.
Bit	The smallest unit of data in computing represented by a 1 in binary.
Cache	A piece of temporary memory. It can refer to a part of the RAM, storage disk, CPU, or an area for storing web pages.
Data	Units of information. In computing there can be different data types, including integers, characters and Boolean. Data is often acted on by instructions.
Embedded system	A special purpose computer built into another device.
GB	Gigabyte (GB) - a measurement of file size or storage capacity, 1,024 megabytes, or 1 billion bytes.
General Purpose Computer	A computer designed to be able to carry out a wide range of instructions.
Hard Disk Drive	A device used to store large amounts of data.
Laser	An intense beam of light that is monochromatic and coherent (in phase).
Media	Physical devices that are used to transfer data.
Memory	The part of a computer that stores data.
Non Volatile Memory	A form of computer memory that stores data even when not powered.
RAM	Random access memory. This is volatile memory that is constantly being written to and read from. It does not retain its contents without a constant supply of power. When a computer is turned off, everything stored in its RAM is lost.