Tiverton High School Year 9 Computing
Autumn Term Knowledge Organiser Part 1
Number Bases and Data Representation

Number Bases and Units of Data Storage (Unit 9-1)

Binary means base-2

Denary means base-10

Hexadecimal means base-16

Humans traditionally use **denary (base 10)** when dealing with numbers.
Computers always use **binary (base 2)** to store and process digital data.

Electronic computers contain millions of tiny **transistor** components.
A transistor behaves like a **switch**, that can only be turned **on** or **off**.
Because binary only uses two possible digits, these closely match the on/off states of the transistors that computers are made of.
The **on** or **off states** of transistors can be used to represent the two different number symbols that binary uses:

off means **0**
on means **1**

A **bit** is the **smallest** amount that a computer can store - one **binary digit**.

8-bit binary means a **pattern** of exactly **8 binary-digits**.

8-bits allow **256 possible combinations** between **00000000** and **11111111**.

This is why 8 bits can represent between **0** and **255** in base ten.

1 byte	=	**8 bits** (an ASCII character takes 1 byte)
1 kilobyte	=	**1000 bytes**
1 megabyte	=	**1000 kilobytes** (or 1000 x 1000 bytes)
1 gigabyte	=	**1000 megabytes** (or 1000 x 1000 x 1000 bytes)
1 terabyte	=	**1000 gigabytes** (or 1000 x 1000 x 1000 x 1000 bytes)

Converting Binary (base two) to Denary (base ten)

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

(**1** x 64) + (**1** x 8) + (**1** x 2) = **74** in base ten

Converting Hexadecimal (base sixteen) to Denary (base ten)

Hexadecimal is a more **compact** and **convenient** way to represent **large** numbers than binary.

Large numbers can be represented using **fewer** hexadecimal digits.

Hexadecimal numbers can only uses the symbols **0123456789ABCDEF**

A means **10**
B means **11**
C means **12**
D means **13**
E means **14**
F means **15**

16	1
2	**D**

2 groups of 16, plus **D** units.
(**2** x 16) + (**D** x 1)
(**2** x 16) + (**13** x 1)
32 + 13
45 in base 10.

How Computers Store Text (Unit 9-1)

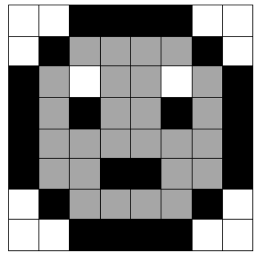
A **character** is a **symbol** that can be represented and stored by the computer system.
The full collection of ALL of the characters that a computer can represent/store is called a **character set**.
Each character symbol is represented using a **special number** called a **character code**.

ASCII is the **American Standard Code for Information Interchange**. It can be used for writing in the English language.
Plain **ASCII text** is often stored using **7 bits** per character.

A better version of ASCII is **Extended ASCII**. This can be used for writing in **English**, **French**, **German**, **Spanish** or **Italian**.
Extended ASCII allows more characters than original ASCII, but uses **8 bits (1 byte)** to store each different character code.
Lächeln **bonjour à tous** **¡Rápidamente!**

Unicode is a better character set. It can represent **any** language in the world, including Russian and Chinese, not just English.
Unicode uses up to **32 bits (4 bytes)** to store each character code.
Emoji pictures are character symbols from the Unicode character set. ASCII and Extended-ASCII do not contain any emojis.
中文 **р у с с к и й** **日本語** ☺

How Computers Store Images (Unit 9-1)



Bitmap images are **pictures** that are made up of **pixels** (picture elements).

A **pixel** is a small coloured dot in a picture.

All of the pixels are arranged in a grid, a little bit like a mosaic.

The **colour** of each pixel is stored in the memory of the computer using binary digits... **1**s and **0**s.

The **bit-depth** of an image means **how many binary digits are used to store each pixel**.

A **1-bit image** uses exactly **1 bit to store each pixel** in the picture. This allows **2 possible colours**.
A **2-bit image** uses exactly **2 bits to store each pixel** in the picture. This allows **4 possible colours**.
An **8-bit image** uses exactly **8 bits to store each pixel** in the picture. This allows **256 possible colours**.

Photoshop uses **24-bit images**. It uses **24 bits to store each pixel**. This allows **16,777,216 possible colours** for **realistic** pictures.

Resolution means the **density** of the pixels in an image: **how many pixels will fit into a certain area**.
The **resolution** of an image defines how large the individual pixels are drawn.
The higher the resolution, the more life-like the image/better quality, but the more data will be included in the bitmap file.

Most computer **screens** use **72 dots per inch** - large pixels.
Many **printers** use **150 dots per inch** or **300 dots per inch** - the smaller pixels produce a more **detailed** picture on paper.

The **colour-model** used by a program controls **how colours are mixed together** to make pictures.

Most computer programs use the **RGB (Red-Green-Blue) colour model** to display images on the screen.
The colour of any pixel can be made by mixing **red light**, **green light** and **blue light** togther in varying amounts.

Many printers use the **CMYK model**. They combine the colours **Cyan**, **Magenta**, **Yellow** and **Black** in different amounts.

Many image files contain **extra data**, as well as the pixel data. The extra data is called **meta-data**.

Meta-data can be used by programs to **reconstruct** and display images from a file of binary data:
It includes the **width**, the **height**, the **resolution** and the **bit-depth** of the image.

Extra meta-data can also be included in an image file, such as the **file format**, the **date/time it was created**, **who owns the copyright** and the **GPS coordinates of where a photo was taken**.

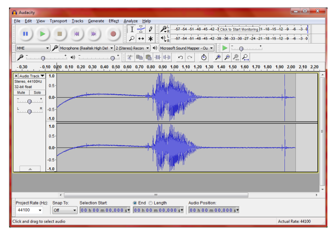
How Computer Store Audio - Sounds and Music (Unit 9-1)

To represent audio/sound inside a computer, soundwaves are converted to digital data.

First of all, a sound wave must be captured by a microphone as electrical signals.
The **height of the sound wave** can then be **measured** at regular intervals. We call each measurement a **sample**.
The **number of sample measurements that are generated each second** is called the **sample-rate**. This is **measured** in **Hertz (Hz)**.
Each sample/measurement is stored in the computer using a binary number.
The **number of binary digits used in each sample** is called the **sample-size**.

A higher sample rate and sample size, leads to a larger audio-file, but a better-quality recording.
A realistic audio file will need to use thousands of samples a second. Common sample rates include **22050 Hz** or **44100 Hz**.

To reconstruct a sound from binary data, a audio file needs to contain extra meta-data that describes how the binary data is structured and how to play it back:



- **Duration** of the sound (how many **seconds** the recording lasts).
- **Sample-Rate** (**how many samples were used each second** e.g. **8000 Hz**).
- **Sample-Size** (**how many bits each sample contains** e.g. **32 bits**).
- **Channels** (**how many speakers are needed** e.g. **1 for mono**, **2 for stereo**).
- **Date & time** that the audio file was **created** or **last changed**.
- **Author**, **genre** or **copyright information** about **who** created the recording.