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**.

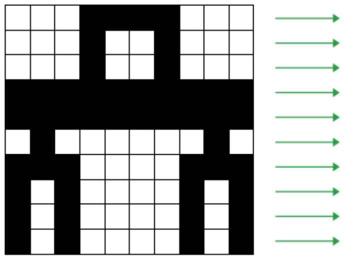
ASCII code can only store 128 characters, which is enough for most words in English but not enough for other languages.
If you want to use accents in European languages or larger alphabets such as Cyrillic (the Russian alphabet) and Chinese Mandarin then more characters are needed.

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

Representing Images—Black and White

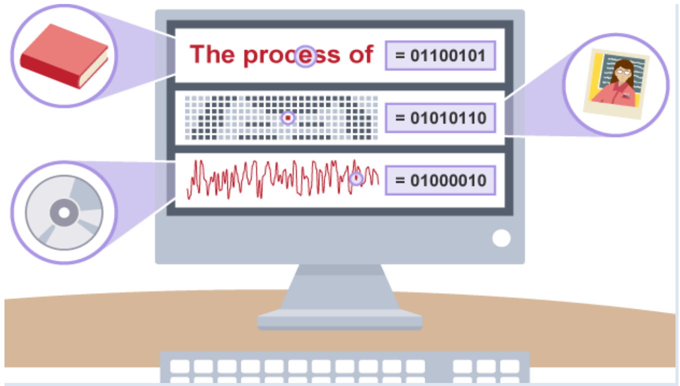
Images also need to be converted into **binary** in order for a computer to process them so that they can be seen on our screen. Digital images are made up of **pixels**. Each pixel in an image is made up of binary numbers.

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.



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

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

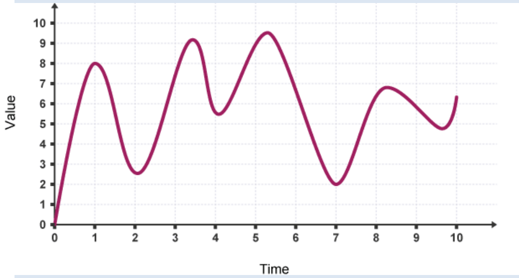
Image quality is affected by the **resolution** of the image. The resolution of an image is a way of describing how tightly packed the pixels are.

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 size will be bigger.

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.

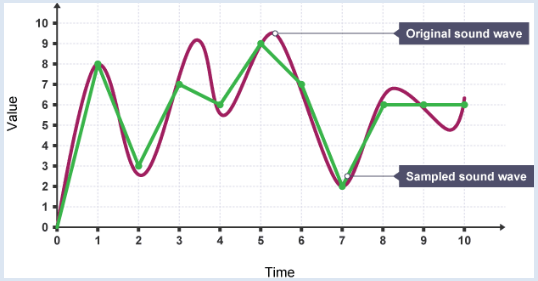
An **analogue** to digital converter will sample a sound wave at regular time intervals. For example, a sound wave like this can be sampled at each time sample point:



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.