Data Representation

Year 8 | Theme 1

Knowledge Organiser

This is the **Binary Number Line** or the **Binary Place Values**. As you can see we start with 1 from the right and double as we go along!

128	64	32	16	8	4	2	1

We can use this as a tool for many conversions from binary to denary and vice versa! x 2 x 2 x 2

What is binary?

Computers use electrical signals that are on or off, so they have to see everything as a series of **binary** numbers.

Binary is a number system that uses only **two** digits: **0** & **1** (on and off).

All data that we want a computer to process needs to be converted into this binary format.

Binary is known as a 'base-2' system. This is because:

- There are only two digits to select from (0 and 1)
- When using the binary system, data is converted using the power of two *(on the binary number line the place values are multiplied by two each time!)*

Binary	Denary	Hexadecimal
Base-2	Base-10	Base-16

What is Denary?

People use the denary (or decimal) number system in their day-to-day lives. ***In Maths you use Denary!***

Denary is a number system that uses **ten** digits to represent numbers: 0,1,2,3,4,5,6,7,8 and 9.

Denary is known as a 'base-10' system. This is because:

- There are only ten digits to select from (0 through to 9)
- When using the denary system, data is presented using the power of 10 *(on the number line you may be familiar with since primary school the place values are multiplied by ten each time e.g. units, tens, hundreds, thousands…)*

Th	H	T	U
4	3	6	4

= *4 thousand 3 hundred and 64*

4364

How to convert from Binary to Denary

Example – Convert **10101000** into denary

1. Start by writing out the binary place values.
2. Put the binary number into the place values table from right to left.

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

on / off

3. Add up the numbers on the **place value** column where there is a binary digit of 1 (meaning it's "switched on"!)
4. 128 + 32 + 8 = 168
5. So **10101000** is equal to **168** in denary!

How to convert from Denary to Binary

Example – Convert **193** into binary

1. Start by writing out the binary place values

128	64	32	16	8	4	2	1

2. This time we start at the far left, we say "does 128 fit into 199?" Yes it does! So place a 1 in the '128' place value column and perform the subtraction: 193-128=**65**

128	64	32	16	8	4	2	1
1							

3. 71 is now our up to date number to convert. We repeat the same step on the next column. "Does 64 fit into 65?" Yes it does! So place a 1 in the '64' place value column and perform the subtraction: 65-64 = 1
4. 1 is now our most up to date number. We continue moving right. 32 does not fit into 1…. 16 does not fit into 1…. 8 does not fit into 1…. 4 does not fit into 1…. 2 does no fit into 1…. 1 fits into 1! Be sure to include 0's when a number doesn't fit.

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

5. So **193** is equal to **11000001** in binary!

Binary Addition Rules

0 + 0 = **0**

0 + 1 = **1** *or* 1 + 0 = **1**

1 + 1 = **1 carry 1**

1 + 1 + 1 = **1 carry 1**

Note: 1 + 0 + 1 would be same as 1 + 1

Example:

```
  1 1 0 1      13
+ 0 1 1 0    +  6
1 0 0 1 1     19
1 1
```

Unit of Information

smallest

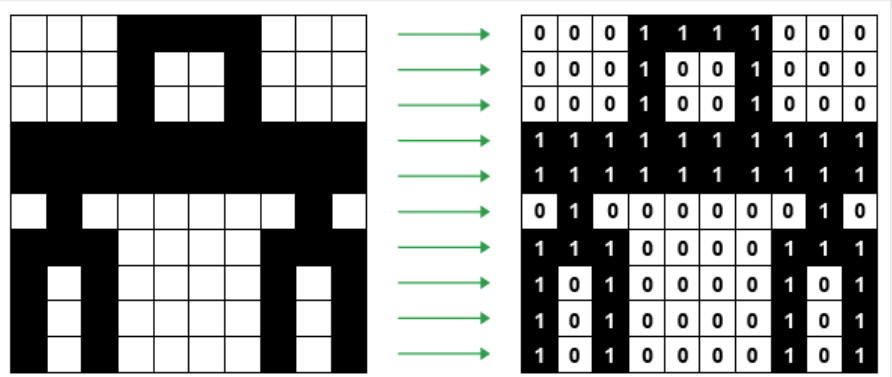
1 bit	e.g. 0 or 1
8 bits	1 byte e.g. 01101110
1,000 bytes	1 kilobyte (KB)
1,000 kilobytes	1 megabyte (MB)
1,000 megabytes	1 gigabyte (GB)
1,000 gigabytes	1 terabyte (TB)
1,000 terabytes	1 petabyte (PB)

largest

Representing Images

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 of **pixels**. Each pixel is an image is made up of binary numbers. If we say 1 is black (or on) and 0 is white (or off), then a simple black and white picture can be created using binary.



Pixel is short for picture element.

A pixel is a single point within a bitmap image.

A bitmap Image is a map of bits made up of pixels.

Computing @ turton