
Data and Storage

Computers just love data. They like nothing better than to store a whole load of lovely data for you. They'll store it for you until you need it again, then they'll present it on screen for you, just as it was when you gave it to them. But hang on a minute, just what is data? And why is a computer so good at storing it?

You are creating data all the time when using a computer. If you type something using a computer, you create data. I'm creating data right now as I write this book on my computer. If you use a computer to draw something, that's data too, and if you use a computer to take a picture or make a movie, guess what? That's data too, and your computer is quite happy to store it all for you.

Believe it or not, there was once a time when all these tasks required a separate piece of expensive equipment. Older people, like me, maybe like you, will remember using devices like the typewriter to write letters, a camera to take photographs, another different camera to make

movies, and possibly another separate device (a microphone) to record sound. Now we can accomplish all these tasks with a single computer, and the amazing thing is, that computer doesn't have to be a big machine sitting on a desk with a separate screen, mouse and keyboard. It can be small enough for you to carry around in your bag. It may even be small enough to fit in your pocket. Tablets and smart phones are computers too, and they store our data in exactly the same way as a big powerful desktop computer.

But how do computers store our data I hear you ask. Good question. The amazing, mind blowing answer is, that a computer stores all our data as zeros and ones. Just zeros and ones, nothing else. It doesn't matter if it's text, like the text in this book, a drawing, a photograph, or even a movie, when a computer stores something as data it's just storing zeros and ones. Not just random zeros and ones mind, the sequence is very important, but it is only storing lots and lots of zeros and ones. Told you it was mind blowing.

Computers are able to store all our data like this because they use a different number system to us. It's called the binary number system. Here's how it works.

The binary number system

You're already very familiar with one number system: the one you use every day. It's called denary. Consider the denary number 1111 for a moment. We interpret this as one thousand, one hundred and eleven. That's because we've been taught that each digit in a denary number (starting from the right hand side) is worth ten times the previous digit. This is the digit's 'place value', and because each digit's place value is worth ten times the previous digit, denary is often referred to as a 'base 10' number system.

1111 in denary means one 1, one 10, one 100 and one 1,000. Add those numbers together and you get one thousand, one hundred and eleven.

1,000's	100's	10's	1's
1	1	1	1
= one thousand, one hundred and eleven			

But to a computer 1111 represents the number fifteen! That's because in the binary number system each digit (starting from the right hand side) is worth two times the previous digit. This is why the binary number system is often referred to as a 'base two' number system.

1111 in binary means one 1, one 2, one 4 and one 8. Add those numbers together and you get fifteen.

8's	4's	2's	1's
1	1	1	1
= fifteen			

The other key difference between the denary and binary number systems is that binary only allows you to write a zero or one for each place value. If you want to write the number two for example, you wouldn't just write 2 for the first place value, like you would in denary, you would write 1 0, which means zero 1's and one 2.

Which brings to mind the classic binary joke...

"There are 10 types of people in this world: those who understand binary, and those who don't."

So why do computers use the binary number system to store data? To answer that question we briefly need to consider how a computer's memory (where it stores our data when we are working on it) is constructed.

A computer's memory consists of millions and millions of tiny electrical switches, which just like a light switch, can be off or on. Because each switch has an off or on state, it can be used to represent a zero, when it's switched off, or a one, when it's switched on.

Computers work with these switches in blocks of eight to store our data. Each individual switch is called a bit (short for **binary digit**) and a block of eight bits is called a byte. By using the binary number system computers can store any number from zero to 255 in a single byte of storage, just by switching the individual switches (bits) off or on. Here's some examples.

A computer would store the number zero like this.

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

All 8 switches (bits) are switched off (set to zero). Kind of makes sense doesn't it. All the switches are off, so how could it be any number other than zero.

A computer would store the number 255 like this.

128	64	32	16	8	4	2	1
1	1	1	1	1	1	1	1
= 255							

All eight switches (bits) are switched on (set to one). Add all the numbers in the top row together: $1 + 2 + 4 + 8 + 16 + 32 + 64 + 128$. It adds up to 255 doesn't it?

A computer would store the number one like this.

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

Note how it has just switched on the first bit in the byte. The rest are off (set to zero).

It would store the number two like this.

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

Note that this time it has just switched on the second bit in the byte. The bit that represents two.

It would store the number three like this.

128	64	32	16	8	4	2	1
0	0	0	0	0	0	1	1
= 3							

Note that this time it has just switched on the first and second bits in the byte: the bits that represent one and two. Add one and two together and you get three.

Hopefully, you're starting to get the idea now. By switching on the appropriate bits in a byte, then adding together the numbers they represent, you can make any number from zero to 255. Still don't believe me? Try it yourself overleaf.

Storing big numbers

To store numbers bigger than 255 a computer simply uses more bytes of storage. A second byte of storage can be used to represent the binary place values 256, 512, 1024, 2048, 4096, 8192, 16384 and 32768. So, by using two bytes of storage, a computer can store any number from zero to 65,535. It would store the number 65,535 by switching on all bits in two bytes of storage (sixteen bits).

By using three bytes of storage to store a number, a computer can store any number from zero to 16,777,215. And using four bytes it's any number from zero to 4,294,967,295. That's over four billion different numbers!



Activity 1: Storing numbers with bytes

1. 'Store' the following numbers by inserting zeros and ones to represent 'off' or 'on' switches.

128	64	32	16	8	4	2	1
= 10							

128	64	32	16	8	4	2	1
= 25							

128	64	32	16	8	4	2	1
= 50							

128	64	32	16	8	4	2	1
= 100							

128	64	32	16	8	4	2	1
= 200							

2. Try some more if you like.

128	64	32	16	8	4	2	1
= 15							

128	64	32	16	8	4	2	1
= 30							

128	64	32	16	8	4	2	1
= 60							

128	64	32	16	8	4	2	1
= 120							

128	64	32	16	8	4	2	1
= 240							