Binary and Hexadecimal

Background

Computers chips are made of transistors. Transistors act like microscopic switches that can turn off and one. The one difference between transistors and light switches is that they can control each other. (It's like if you flick the switch in your room, it will light up your room but it will also check if the light is on in the kitchen and if so it will turn on the light in the bathroom.) *You'll learn more about this when you study logic gates in an electronics course.*

Being switches, they can be ON or OFF, which we can also call TRUE or FALSE, 1 or 0.

We can use this to store numbers ...

In the **decimal system** we have 10 digits: 0 to 9.

In the **binary system** we have 2 digits: 0 to 1.

How many numbers can you make using the digits 0-9? Infinite! So the same applies to binary numbers.

How numbers work

Decimal: Once we get to more than 9, we have to add a number in the next column. This number is always a 1. The 9 becomes a 0.

8 -> 9 -> 10 -> 11 ... 99 -> 100 -> 101

Binary: Once we get to more than 1, we have to add a number in the next column. This number is always a 1. The original 1 becomes a 0.

0 -> 1 -> 10 -> 11 -> 100 -> 101

Anytime you want to write a 2 (which would normally come after 1, write a 0 and carry a 1 to the next column to the left).

Column Headings

In decimal the column headings are powers of 10. Consider the number 5832

10^3	10^2	10^1	10^0
1000	100	10	1
5	8	3	2

5832 = 5x1000 + 8x100 + 3x10 + 2x1

In binary the column headings are always multiples (powers) of 2.

512	256	128	64	32	16	8	4	2	1
				1	1	0	0	1	1

The binary number 110011 = 32 + 16 + 2 + 1 = 51

★ The lowest column heading is always 1, and you always start from the right and go left.
★ Everywhere where there is a 1, add that number. If there is a zero in a column then that number is not counted (e.g. 0x8 + 0x4 + 1x2 + 1x1 = 2 + 1 = 3).

This is how you change **Binary --> Decimal**

Practice:
Convert 11011 from binary to decimal:

```
16 8 4 2 1
 1 1 0 1 1
```

So we have one 16 + 8 + 2 + 1 = 27. 11011 in binary is 27 in decimal.

☆ You can check your answers with the calculator on your computer. Just select “programmer” mode. ☆

Decimal --> Binary

Let’s convert 46 to binary.

- First write down column headings from right to left, stopping at the first number that is bigger than 46.

  64 32 16 8 4 2 1

- Put a 0 in that column, and a 1 in the next (there is a 32 in 46)

  64 32 16 8 4 2 1
  0 1

- Now subtract 32 from 46: 46-32 = 14 There are no 16s in 14, so put a zero in the 16 column

  64 32 16 8 4 2 1
  0 1 0

- Continue on the same way with the rest of the columns
  There is an 8 in 14, and there is 6 left over. 6 is made of 4 + 2. So we get

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

- Read off the final answer, ignoring leading zeros.
  46 = 101110

Sometimes we use subscripts to denote decimal, binary, hexadecimal:

$46_D = 101110_B$

♦ Practice doing binary <--> decimal conversions with numbers in the hundreds.

Check this out: $\mathbf{6073_D = 1011110111001_B}$

<u>Two problems:</u>

1. binary numbers get big really really quickly
2. humans find it really hard to read and copy binary numbers. You can probably remember 6073, but not its binary version.

Hexadecimal

Hexadecimal solves these problems (and others).

But first, some more background ...

One binary digit is called **1 bit**. So 110 is 3 bits.

Eight bits is called **1 byte.** So 1111000 is one byte.

One byte goes from 0000 0000 to 1111 1111.
The smallest number is 0, the biggest is 8 ones. We can see that the subsequent number is 256,

256	128	64	32	16	8	4	2	1
0	1	1	1	1	1	1	1	1

so the biggest number with 8 bits must be 255.

This means that 1 byte (8 bits) can have 256 different numbers: 0 to 255.

► Everything that is stored in your computer is stored as a number. This means letters, numbers, colours, music, images, movies, programs, are all numbers.
The smallest unit of information that we typically use is one byte.

Characters

In terms of things we type, we have 26 upper case letters + 26 lower case + 10 digits + about 35 punctuation characters. The total is about 95. We can encode all of the letters that we type with 7 bits (0-127), so we can use a byte for each letter, and just leave the first bit to be 0.

This leads us to the ASCII way of encoding letters. See “Ascii chart”.

A is 65, B=66, ... a = 97, b = 98, ...

Colours

We encode colours with 1 byte for red, one for green, and one for blue.
The red value goes from 0-255, same for green and blue. (Check this out using “Paint” and custom colours).
That gives 256x256x256 = 16.8 million different colours that can be displayed. Not as much as the eye can see, but still enough to make really good images.

Hexadecimal

We can split a byte into two groups of 4 bits. (Each group of 4 bits might be called a nybble).

What numbers can we make using only 4 bits?

Binary	Decimal equivalent	
0000	0	
0001	1	
0010	2	
0011	3	
0100	4	
0101	5	
0110	6	
0111	7	
1000	8	
1001	9	
1010	10	**A**
1011	11	**B**
1100	12	**C**
1101	13	**D**
1110	14	**E**
1111	15	**F**

This is handy!
We can go from 0-15,
that's 16 numbers.

Let's call this hexadecimal.

But we need a single digit to represent each number, so we'll use the letters A-F

Binary <---> Hexadecimal conversions are easy

Hexadecimal --> Binary

(1) Convert 7C4D to binary

- ♦ take each hex digit and convert it to the correct 4 bit number
  7 = 0111 C is 12 and 12 is 1100 4 is 0100 D is 13 and 13 is 1101
- ♦ put them all together
  0111110001001101
- ♦ you can remove leading zeros and put in groups of 4 bits if you wish (for ease of reading)

Answer: 111 1100 0100 1101

and **Binary --> Hexadecimal**

(2) Convert 11011110110101 to hexadecimal

- ♦ Just reverse the process that we did above
- ♦ Start from the right side and group in 4 bits

  11 0111 1011 0101

- ♦ Pad with zeros to get 4 bits in left most group

  0011 0111 1011 0101

- ♦ Add column headers if you need to

```
8421 8421 8421 8421
0011 0111 1011 0101
```

- ♦ Write down what each group is individually

```
8421 8421 8421 8421
0011 0111 1011 0101
  3    7   11    5
```

Replace 11 with hexadecimal letter B,   and the answer is 37B5

♥ Each byte is represented by two hex characters. ♥

Hexadecimal Colours

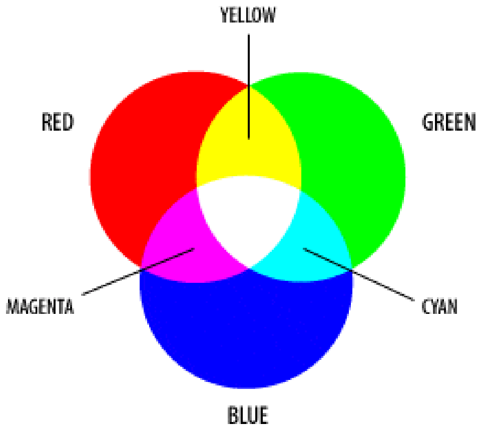
Colours are encoded by specifying the amount of primary colours red, green, and blue. Each colour gets 1 byte of data. So there are 256 shades of each of red, green, and blue, which makes about 16 million colours.

The range for the red byte goes from 0-255, or 00-FF. 00 is the darkest that the colour can be and FF (255) is the brightest. *You can think of FF as 100% if you want to*. It's the same for G and B.

A colour of #00 00 00 is black. #FF FF FF is white.

What is this colour? #9016CC   (for RR GG BB)

Which means quite a bit of red (90), hardly any green at all (16), and a lot of blue (CC). This colour will be a nice rich purple colour.



Hexadecimal --> Decimal

> This is a little difficult <

★ You have to know that the columns of hexadecimal numbers are powers of 16, just like decimal ones are powers of 10.

What is $35FC8_H$ in decimal?

16^4	16^3	16^2	16^1	16^0
65536	4096	256	16	1
3	5	F	C	8

<-- column headings

Answer: 3x65536 + 5x4096 + Fx256 + Cx16 + 8x1

= 3x65536 + 5x4096 + 15x256 + 12x16 + 8x1

= 221128

Decimal --> Hexadecimal

>>> This is a quite difficult <<<

(At least everything else was relatively easy until now.)

★ You have to know that the columns of hexadecimal numbers are powers of 16, just like decimal ones are powers of 10.

Convert 178553 to hexadecimal.

Solution:

1. first look at the columns. Find a hex column that is almost equal to the total number:
   a) 4096 is not big enough
   b) 65536 is smaller than 178553, so we'll start with that.
   c) 178553 ÷ 65536 = 2.7254 The first digit is 2!
   This 2 is in the 65536 column which is the 5th column.

   2 ? ? ? ?

2. Find the remainder after we do 2 x 65536:
   a) 178553 – 65536*2 = 47481 (check: this number should be less than 65536)
3. Divide this remainder by the next hexadecimal column: 4096
   a) 47481 ÷ 4096 = 11.592 The next digit is 11 which is a B

   2 B ? ? ?

4. Find the remainder:
   a) 47481 – 4096*11 = 2425 (check: this number is less than 4096, yep.)
5. Divide this by the next hexadecimal column: 256
   a) 2425 ÷ 256 = 9.472 The next digit is 9

   2 B 9 ? ?

6. Find the remainder from the above operation:
   a) 2425 – 256 * 9 = 121
7. How many 16s go into this?
   a) 121/16 = 7.562 The next digit is 7

   2 B 9 7 ?

8. The remainder is 121 – 16*7 = 9

   The final digit is 7

   2 B 9 7 9

So 178553 is 2B979 in hexadecimal.