Number Bases

Decimal (base 10) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9

1,000,00	100,000	10,000	1,000	100	10	1
10^6	10^5	10^4	10^3	10^2	10^1	10^0

234

2 x 100 = 200
3 x 10 = 30
4 x 1 = 4

200+30+4 = 234

Count to ten
0
1
2
3
4
5
6
7
8
9
10 ten
11 eleven
..
19
20
21
..
98
99
100 hundred

Binary (base 2) 0, 1

electricity “on” 1
“off” 0

bit is 1 or 0 (binary digit)

digital – code our data only with 1’s and 0’s

8 bits is 1 byte

1 byte = 8 bits

128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Count to ten

0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010 ten

0000 0101 binary = 5 decimal

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

1 x 4 = 4
1 x 1 = 1
4+1 = 5

0100 0011 binary = 67 decimal

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

64+2+1 = 67
```

1. Convert binary number **0000 1001** to decimal

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

8 + 1 = **9**

2. Convert binary number **0000 1101** to decimal

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

8 + 4 + 1 = **13**

3. Convert binary number **1001 1010** to decimal

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

128 + 16 + 8 + 2 = **154**

4. Convert binary number **0110 1101** to decimal

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

64 + 32 + 8 + 4 + 1 = **109**

5. Convert binary number **0100 1001 0110** to decimal

We need to extend the model here … keep multiplying by 2

So the next place holders are 256, 512, 1024, and 2048

Notice how easy it is to get a 1 and 0 out of place :-(

Thus we group them by 4's and add spaces …

0	1	0	0	1	0	0	1	0	1	1	0
2048	1024	512	256	128	64	32	16	8	4	2	1

1024 + 128 + 16 + 4 + 2 = **1,174**

6. Convert decimal number **19** to binary **0001 0011**

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

19 – 16 = 3
3 – 2 = 1
1 – 1 = 0 (done)

16 + 2 + 1 = 19

7. Convert decimal number **77** to binary **0100 1101**

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

77 – 64 = 13
13 – 8 = 5
5 – 4 = 1
1 – 1 = 0 (done)

64 + 8 + 4 + 1 = 77

8. Convert decimal number **176** to binary 1011 0000

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

176 – 128 - 48
48 – 32 = 16
16 – 16 = 0 (done)

128 + 32 + 16 = 128

```
 1
128
 32
 16
---
176
```

9. Convert decimal number **275** to binary **0001 0001 0011**

```
  0    0    0   1    0   0   0   1    0   0   1   1
---- ---- --- ---  --- --- --- ---  --- --- --- ---
2048 1024 512 256  128  64  32  16   8   4   2   1
```

275 – 256 = 19
19 – 16 = 3
3 – 2 = 1
1 – 1 = 0 (done)

256 + 16 + 2 + 1 = 275

Hexadecimal "hex" (base 16) 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

65,536	4,096	256	16	1
16^4	16^3	16^2	16^1	16^0

```
--- --- ---
256  16  1
```

"Short cut" for binary

Each hex digit represents x4 binary digits

We can directly (and quickly) convert from hex to binary (and binary to hex)

For example,

hex digit 1 => 0001 as 4 digit binary number
hex digit 2 => 0010 as 4 digit binary number
…
hex digit A => 1010 as 4 digit binary number
…
hex digit F => 1 1 1 1 as 4 digit binary number
84 2 1

"Hex" Base 16

2	A
16	1

"Binary" Base 2

0	0	1	0	1	0	1	0
8	4	2	1	8	4	2	1

2A hex = 0010 1010 binary

05 hex = 0000 0101 binary

10. Convert hex number **2B** to decimal and also to binary

```
 0   2   B
--- --- ---
256  16  1
```

2 x 16 = 32
B(11) x 1 = 11
32 + 11 = **43 decimal**

2B hex = **0010 1011 binary**

2 = 0010
B = 1011

11. Convert hex number **215 hex** to decimal and also to binary

```
 2   1   5
--- --- ---
256  16  1
```

2 x 256 = 512
1 x 16 = 16
5 x 1 = 5
512 + 16 + 5 = **533 decimal**

```
            2    1    5
215 hex = 0010 0001 0101 binary
```

2 = 0010
1 = 0001
5 = 0101

12. Convert hex number **ABC hex** to decimal and also to binary

```
 A   B   C
--- --- ---
256  16  1

A(10) x 256 = 2560
B(11) x 16 = 176
C(12) x 1 =  12

 1
2560
 176
+ 12
----
```
2748 decimal

```
        A    B   C
ABC = 1010 1011 1100 binary
A(10) = 1010
B(11) = 1011
C(12) = 1100
```

0110 0111 binary = 67 hex

```
8421 8421

  6   7
0110 0111
```