

Fundamental Data Type

Integer

Video 06

1 byte = 8 bit

2 byte = 16 bit

4 byte = 32 bit

`sizeof ()` operator shows us the size of a particular datatype.

```
printf ("%d", sizeof(int));
```

→ 4 byte (depending on machine)
(32 bit)

Range → Upper and lower limit of some set of data

$\{0, 1, 2, 3, 4\}$

Decimal Number System → 0 to 9.

Range of 4 bit data → 0000 to 1111
(0) (15) $2^n - 1 = 2^4 - 1 = 16 - 1 = 15$

0000 → 0	0100 → 4	1000 → 8	1100 → 12
0001 → 1	0101 → 5	1001 → 9	1101 → 13
0010 → 2	0110 → 6	1010 → 10	1110 → 14
0011 → 3	0111 → 7	1011 → 11	1111 → 15

2 byte
(≈ 16 bit)

Unsigned Range: 0 to 65535 $(2^{16}-1)$
Signed Range: -32768 to +32767

$\rightarrow 0 \rightarrow$

0000 0000 0000 0000 $\rightarrow 0$

to

1111 1111 1111 1111 $\rightarrow 65535$

4 byte
(≈ 32 bit)

Unsigned Range: 0 to 4294967295 $(2^{32}-1)$
Signed Range: -2147483648 to

+2147483647

0000 0000 0000 0000 0000 0000 0000 0000 $\rightarrow 0$

to

1111 1111 1111 1111 1111 1111 1111 1111

$\rightarrow 4294967295$

$-2^{31} = -1 \times 2^{31} = -1 \times 2^{31}$ [1111 of 0000 \leftarrow detach bit 0 to ignore
(21) \leftarrow (21) (0)

8 \leftarrow 0011 7 \leftarrow 0001 6 \leftarrow 0010 5 \leftarrow 0000

4 \leftarrow 1011 3 \leftarrow 1001 2 \leftarrow 1010 1 \leftarrow 1000

0 \leftarrow 0111 15 \leftarrow 0101 14 \leftarrow 0110 13 \leftarrow 0100

12 \leftarrow 1111 11 \leftarrow 1101 10 \leftarrow 1110 9 \leftarrow 1100