

(Video 7) Fundamental Data Type. (Part 2) Integer

If ~~size~~ integer size is 4 bytes,

short int may be 2 bytes.

↳ 2 byte

long int 8 byte

[-0]

⇒ $\text{sizeof}(\text{short}) \leq \text{sizeof}(\text{int}) \leq \text{sizeof}(\text{long})$

[Change 2003 and 2011 standard] [Up to machine]

Modifiers: (short, long, signed, unsigned) and for

Integer

#include <stdio.h>

int main() {

printf("sizeof (long int) is %d", sizeof(long int));

return 0;

}

output 8 bytes.

~~(Video)~~

*** by default, 2

int variable_name;

↳ signed integer variable

Unsigned int variable_name;

↳ allows only positive values.

<stdint.h> defines

<stdint.h>

by default
for 32-bit

signed integer then
only positive
value assigned

#include <stdio.h>

#include <limits.h>

int main () {

int var1 = INT_MIN;

int var2 = INT_MAX;

printf ("range of signed integer %d to %d",
var1, var2);

return 0;

}

output: -2147483648 to 2147483647

check signed integer

sizeof (int) → 4 byte
32 bit

0000 ... 0000
1111 ... 1111

#include <stdio.h>

#include <limits.h>

32bit
4byte
int main() {

unsigned int var1 = 0;

unsigned int var2 = UINT_MAX;

printf("range of unsigned integers %u to %u",
var1, var2);

return 0;

<limits.h>

used to print
unsigned decimal value.

output → 0 to 4294967295

0123456789
FEDCBA9876543210

Same

```
int main() {
```

```
    short int var1 = SHRT_MIN;
```

```
    short int var2 = SHRT_MAX;
```

```
    printf("var1, var2",
```

```
        printf("%d %d", var1, var2);
```

```
    return 0; }
```

limits.h

→ short

2 byte → 4 bit 16 bit

0000 0000 0000 0000
1111 1111 1111 1111

$2^{16} - 1 = 65535$

As signed → -32768 to $+32767$

short unsigned int var1 = 0;

short unsigned int var2 = USHRT_MAX;

output → 0 to 65535

order not matter →

→ short unsigned.

```
int main() {
```

```
    long int var1 =
```

```
    long int var2 =
```

```
    printf("%ld - %ld")
```

```
    return 0; }
```

```
int main() {
```

```
    long unsigned int var1 = 0000 0000
```

```
    long unsigned int var2 =
```

```
    printf("%ld - %ld")
```

```
    printf("%lu - %lu")
```

```
    return 0; }
```

```
    return 0; }
```

```
    return 0; }
```


another one is (long long int)

if ~~long~~ $\text{sizeof}(\text{long int}) = 4 \text{ byte}$ (32 bit)

then $\text{sizeof}(\text{long long int}) = 8 \text{ byte}$ (64 bit)

size of (long int) = 8 byte (64 bit)

Size of (long long int) = 8 byte

change 2 byte to 4 byte

max limit 64 bit

Summary:

$\rightarrow \text{sizeof}(\text{short}) \leq \text{sizeof}(\text{int}) \leq \text{sizeof}(\text{long})$

\rightarrow signed int variable name; equivalent to writing
int variable name;

$\rightarrow \%d$ used to print signed integer

$\rightarrow \%u$ unsigned

$\rightarrow \%ld$ long integer or signed long integer

$\rightarrow \%lu$ long unsigned integer

$\rightarrow \%lld$ long long integer

$\rightarrow \%llu$ unsigned long long integer.