**Data Types**

a variable in C must be a specified **data type**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Type** | **Size** | **Description** | **Example** |
| int | 2 or 4 bytes | Stores whole numbers, without decimals | 1 |
| float | 4 bytes | Stores fractional numbers, containing one or more decimals.  Sufficient for storing 6-7 decimal digits | 1.99 |
| double | 8 bytes | Stores fractional numbers, containing one or more decimals.  Sufficient for storing 15 decimal digits | 1.99 |
| char | 1 byte | Stores a single character/letter/number, or ASCII values | 'A' |

**Basic Format Specifiers**

|  |  |  |
| --- | --- | --- |
| **Format Specifier** | **Data Type** | |
| **%d or %i** | **int** | |
| **%f or %F** | **float** |  |
| **%lf** | **double** |  |
| **%c** | **char** |  |
| **%s** | **Used for**[**strings**](https://www.w3schools.com/c/c_strings.php)**(text), which you will learn more about in a later chapter** | |

| Type | Size (bytes) | Format Specifier |
| --- | --- | --- |
| int | at least 2, usually 4 | %d, %i |
| char | 1 | %c |
| float | 4 | %f |
| double | 8 | %lf |
| short int | 2 usually | %hd |
| unsigned int | at least 2, usually 4 | %u |
| long int | at least 4, usually 8 | %ld, %li |
| long long int | at least 8 | %lld, %lli |
| unsigned long int | at least 4 | %lu |
| unsigned long long int | at least 8 | %llu |
| signed char | 1 | %c |
| unsigned char | 1 | %c |
| long double | at least 10, usually 12 or 16 | %Lf |

**Programs**

#include <stdio.h>

int main() {

// Basic data types

int num = 10; // Integer

float pi = 3.14f; // Float

double e = 2.718281828; // Double

char letter = 'A'; // Char

// Displaying values

printf("Integer: %d\n", num);

printf("Float: %.2f\n", pi);

printf("Double: %.9f\n", e);

printf("Character: %c\n", letter);

return 0;

}

**With all basic Datatypes**

#include <stdio.h>

int main() {

// Basic data types and their variations

// Integer types

int num = 10; // Regular integer

unsigned int u\_num = 20; // Unsigned integer (only non-negative values)

short int s\_num = 5; // Short integer (smaller range)

long int l\_num = 100000; // Long integer (larger range)

long long int ll\_num = 1000000000; // Long long integer (even larger range)

// Floating-point types

float pi = 3.14f; // Single-precision floating-point

double e = 2.718281828; // Double-precision floating-point

long double ld = 3.141592653589793; // Long double (higher precision)

// Character type

char letter = 'A'; // Single character

// Displaying values

printf("Integer: %d\n", num);

printf("Unsigned Integer: %u\n", u\_num);

printf("Short Integer: %d\n", s\_num);

printf("Long Integer: %ld\n", l\_num);

printf("Long Long Integer: %lld\n", ll\_num);

printf("Float: %.2f\n", pi);

printf("Double: %.9f\n", e);

printf("Long Double: %.15Lf\n", ld);

printf("Character: %c\n", letter);

return 0;

}

**1. Integer Types**

* **int**:
  + Size: 4 bytes
  + Range: Typically from -2,147,483,648 to 2,147,483,647
* **unsigned int**:
  + Size: 4 bytes
  + Range: Typically from 0 to 4,294,967,295
* **short int** (or short):
  + Size: 2 bytes
  + Range: Typically from -32,768 to 32,767
* **unsigned short int** (or unsigned short):
  + Size: 2 bytes
  + Range: Typically from 0 to 65,535
* **long int** (or long):
  + Size: 4 bytes (on 32-bit systems), 8 bytes (on 64-bit systems)
  + Range: On 32-bit systems, from -2,147,483,648 to 2,147,483,647; on 64-bit systems, a much larger range.
* **unsigned long int** (or unsigned long):
  + Size: 4 bytes (on 32-bit systems), 8 bytes (on 64-bit systems)
  + Range: On 32-bit systems, from 0 to 4,294,967,295; on 64-bit systems, much larger.
* **long long int**:
  + Size: 8 bytes
  + Range: Typically from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
* **unsigned long long int**:
  + Size: 8 bytes
  + Range: Typically from 0 to 18,446,744,073,709,551,615

**2. Floating-Point Types**

* **float**:
  + Size: 4 bytes
  + Range: Typically from 1.5 × 10^−45 to 3.4 × 10^38 (6-7 decimal digits precision)
* **double**:
  + Size: 8 bytes
  + Range: Typically from 5.0 × 10^−324 to 1.7 × 10^308 (15-16 decimal digits precision)
* **long double**:
  + Size: 8 bytes or 12 bytes or 16 bytes (depending on the platform and compiler)
  + Range: Similar to double, but with higher precision (often 18-19 decimal digits)

**3. Character Types**

* **char**:
  + Size: 1 byte
  + Range: Typically from -128 to 127 (for signed char), or 0 to 255 (for unsigned char)
* **unsigned char**:
  + Size: 1 byte
  + Range: 0 to 255
* **signed char**:
  + Size: 1 byte
  + Range: -128 to 127

**4. Void Type**

* **void**:

when void is used as a function return type, it indicates that the function does not return a value.