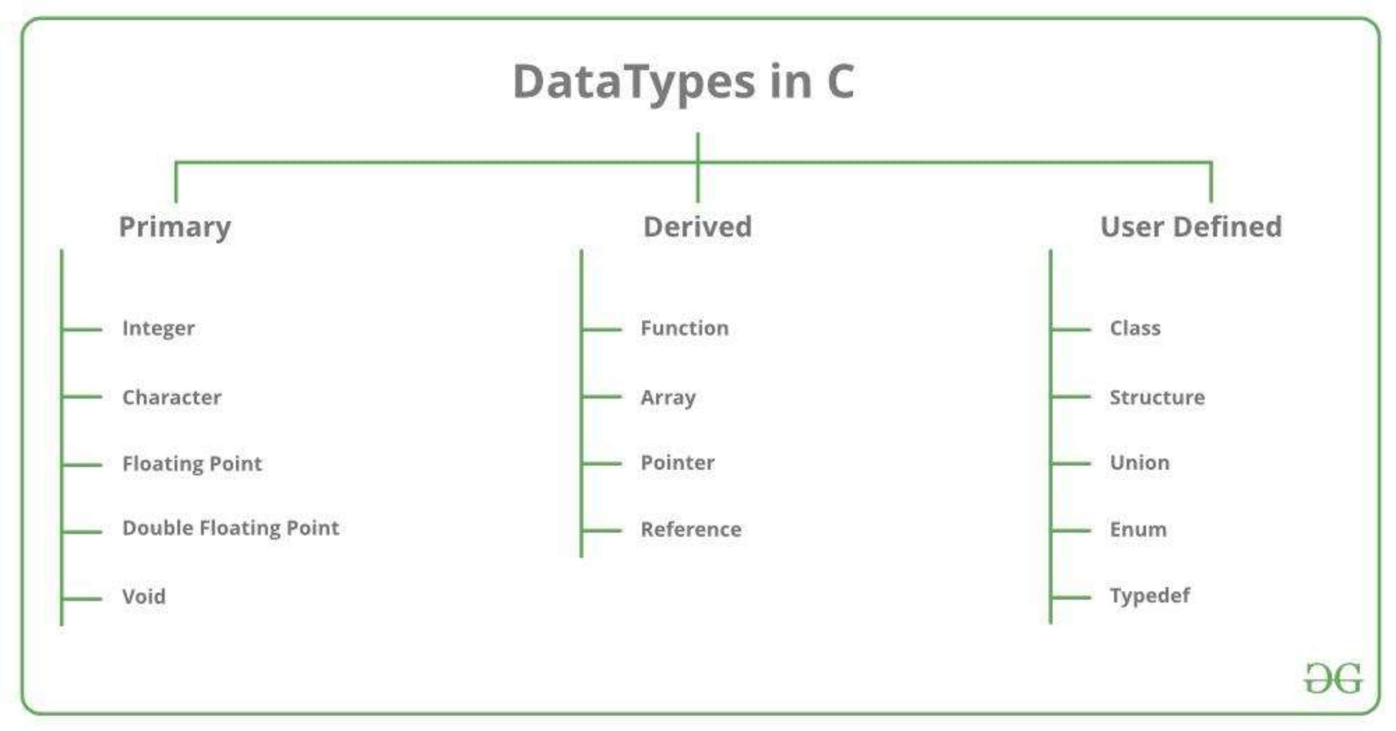
Data Types in C

Each variable in C has an associated data type. It specifies the type of data that the variable can store like integer, character, floating, double, etc. Each data type requires different amounts of memory and has some specific operations which can be performed over it. The data type is a collection of data with values having fixed values, meaning as well as its characteristics.

The data types in C can be classified as follows:

Types	Description
Primitive Data Types	Primitive data types are the most basic data types that are used for representing simple values such as integers, float, characters, etc.
User Defined Data Types	The user-defined data types are defined by the user himself.
Derived Types	The data types that are derived from the primitive or built-in datatypes are referred to as Derived Data Types.



Different data types also have different ranges up to which they can store numbers. These ranges may vary from compiler to compiler. Below is a list of ranges along with the memory requirement and format specifiers on the ***32-bit GCC compiler***.

Data Type	Size (bytes)	Range	Format Specifier
short int	2	-32,768 to 32,767	%hd
unsigned short int	2	0 to 65,535	%hu
unsigned int	4	0 to 4,294,967,295	%u
int	4	-2,147,483,648 to 2,147,483,647	%d
long int	4	-2,147,483,648 to 2,147,483,647	%ld
unsigned long int	4	0 to 4,294,967,295	%lu
long long int	8	-(2^63) to (2^63)-1	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615	%llu
signed char	1	-128 to 127	%c

Data Type	Size (bytes)	Range	Format Specifier
unsigned char	1	0 to 255	%c
float	4	1.2E-38 to 3.4E+38	%f
double	8	1.7E-308 to 1.7E+308	%lf
long double	16	3.4E-4932 to 1.1E+4932	%Lf

Note: *The **long, short, signed and unsigned** are datatype modifier that can be used with some primitive data types to change the size or length of the datatype.*

The following are some main primitive data types in C:

Integer Data Type

The integer datatype in C is used to store the integer numbers(any number including positive, negative and zero without decimal part). Octal values, hexadecimal values, and decimal values can be stored in int data type in C.

- **Range:** -2,147,483,648 to 2,147,483,647
- **Size:** 4 bytes
- **Format Specifier:** %d

Syntax of Integer

We use int keyword to declare the integer variable:

```
int var_name;
```

The integer data type can also be used as

1. **unsigned int:** Unsigned int data type in C is used to store the data values from zero to positive numbers but it can't store negative values like signed int.
2. **short int:** It is lesser in size than the int by 2 bytes so can only store values from -32,768 to 32,767.
3. **long int:** Larger version of the int datatype so can store values greater than int.
4. **unsigned short int:** Similar in relationship with short int as unsigned int with int.

Note: *The size of an integer data type is compiler-dependent. We can use sizeof operator to check the actual size of any data type.*

Example of int

- C

```
// C program to print Integer data types.
#include <stdio.h>

int main()
{
    // Integer value with positive data.
    int a = 9;

    // integer value with negative data.
    int b = -9;

    // U or u is Used for Unsigned int in C.
    int c = 89U;

    // L or l is used for long int in C.
    long int d = 99998L;

    printf("Integer value with positive data: %d\n", a);
    printf("Integer value with negative data: %d\n", b);
    printf("Integer value with an unsigned int data: %u\n",
           c);
    printf("Integer value with an long int data: %ld", d);

    return 0;
}
```

Output

```
Integer value with positive data: 9
Integer value with negative data: -9
Integer value with an unsigned int data: 89
Integer value with an long int data: 99998
```

Character Data Type

Character data type allows its variable to store only a single character. The size of the character is 1 byte. It is the most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.

- **Range:** (-128 to 127) or (0 to 255)
- **Size:** 1 byte
- **Format Specifier:** %c

Syntax of char

The **char keyword** is used to declare the variable of character type:

```
char var_name;
```

Example of char

- C

```
// C program to print Integer data types.
#include <stdio.h>

int main()
{
    char a = 'a';
    char c;

    printf("Value of a: %c\n", a);

    a++;
    printf("Value of a after increment is: %c\n", a);

    // c is assigned ASCII values
    // which corresponds to the
    // character 'c'
    // a-->97 b-->98 c-->99
    // here c will be printed
    c = 99;
    printf("Value of c: %c", c);


    return 0;
}
```

Output

```
Value of a: a
Value of a after increment is: b
Value of c: c
```

Float Data Type

In C programming float data type is used to store floating-point values. Float in C is used to store decimal and exponential values. It is used to store decimal numbers (numbers with floating point values) with single precision.

- **Range:** 1.2E-38 to 3.4E+38
- **Size:** 4 bytes
- **Format Specifier:** %f

Syntax of float

The **float keyword** is used to declare the variable as a floating point:

```
float var_name;
```

Example of Float

- C

```
// C Program to demonstrate use
// of Floating types
#include <stdio.h>


int main()
{
    float a = 9.0f;
    float b = 2.5f;


    // 2x10^-4
    float c = 2E-4f;
    printf("%f\n", a);
    printf("%f\n", b);
    printf("%f", c);

    return 0;
}
```

Output

```
9.000000
2.500000
0.000200
```

Double Data Type

A Double data type in C is used to store decimal numbers (numbers with floating point values) with double precision. It is used to define numeric values which hold numbers with decimal values in C.

The double data type is basically a precision sort of data type that is capable of holding 64 bits of decimal numbers or floating points. Since double has more precision as compared to that float then it is much more obvious that it occupies twice the memory occupied by the floating-point type. It can easily accommodate about 16 to 17 digits after or before a decimal point.

- **Range:** 1.7E-308 to 1.7E+308
- **Size:** 8 bytes
- **Format Specifier:** %lf

Syntax of Double

The variable can be declared as double precision floating point using the **double keyword:**

```
double var_name;
```

Example of Double

- C

```
// C Program to demonstrate
// use of double data type
#include <stdio.h>

int main()
{
    double a = 123123123.00;
    double b = 12.293123;
    double c = 2312312312.123123;

    printf("%lf\n", a);

    printf("%lf\n", b);

    printf("%lf", c);

    return 0;
}
```

Output

```
123123123.000000
12.293123
2312312312.123123
```

Void Data Type

The void data type in C is used to specify that no value is present. It does not provide a result value to its caller. It has no values and no operations. It is used to represent nothing. Void is used in multiple ways as function return type, function arguments as void, and pointers to void.

Syntax:

```
// function return type void
void exit(int check);
// Function without any parameter can accept void.
int print(void);
// memory allocation function which
// returns a pointer to void.
void *malloc (size_t size);
```

Example of Void

- C

```
// C program to demonstrate
// use of void pointers
#include <stdio.h>
```

```
int main()
{
    int val = 30;
    void* ptr = &val;
    printf("%d", *(int*)ptr);
    return 0;
}
```

Output

```
30
```

Size of Data Types in C

The size of the data types in C is dependent on the size of the architecture, so we cannot define the universal size of the data types. For that, the C language provides the sizeof() operator to check the size of the data types.

Example

- C

```
// C Program to print size of
// different data type in C
#include <stdio.h>

int main()
{
    int size_of_int = sizeof(int);
    int size_of_char = sizeof(char);
    int size_of_float = sizeof(float);
    int size_of_double = sizeof(double);

    printf("The size of int data type : %d\n", size_of_int);
    printf("The size of char data type : %d\n",
           size_of_char);
    printf("The size of float data type : %d\n",
            size_of_float);
    printf("The size of double data type : %d",
            size_of_double);

    return 0;
}
```

Output

```
The size of int data type : 4
The size of char data type : 1
The size of float data type : 4
The size of double data type : 8
```