Data Types in C

As C is a statically typed language, before we create a variable we need to decide what type of information this variable is going to store. C programming provides you with different data types to store like character, integer, floating, etc.

Datatypes are mostly categorized into two categories

Primitive Types

These data types are predefined by the C programming language to store simple values such as numbers, characters, and booleans.

Category	Data Types
Integer	short, int, long, long long, unsigned short, unsigned int, unsigned long, unsigned long long
Character	char, unsigned char
Floating	float, double, long double
Others	bool, void

Integer

C provides several integer data types for storing whole numbers, including short, int, long, and long long, along with their unsigned variants that only allow non-negative values and offer extended ranges.

Character

C provides character data types for storing individual characters, as well as their unsigned variants.

Floating

C provides you to storing decimal numbers, including float, double, and long double for larger decimal values.

Others

C provides a boolean data type for true and false values, and the void type is primarily used as a return type for functions that do not return a value.

Example

C

```c
#include <stdbool.h>
#include <stdio.h>

int main()
{
    int age = 48;
    char name[] = "Ladoblanco";
    float weight = 261.5;
    bool isMarried = true;
    char gender = 'M';
    printf("%d \n", age);
    printf("%s \n", name);
    printf("%f \n", weight);
    printf("%d \n", isMarried);
    printf("%c \n", gender);
    return 0;
}
```

```
48
Ladoblanco
261.500000
1
M
```

Output

```
39
Sandeep
68.500000
1
M
```

Non Primitive Types

Arrays

Arrays is sequence of these primitive data types.

Pointers

C language include pointer to store the address value of a variable.

User Defined

User-defined data types in C allow programmers to create custom types for better code organization and readability.

> **Notes:**
>
> - All above data types have fixed range to store data if you store data above the range data should overflow.
> - Number of bytes allocated to the data type is not fixed. it's dependent on the compiler.