Data types specify how we enter data into our programs and what type of data we enter.

**Primary data types**:

These are fundamental data types in C namely integer(int), floating point(float), character(char) and void.

**Derived data types**:

Derived data types are nothing but primary datatypes but a little twisted or grouped together like **array**, **structure**, **union** and **pointer**.

|  |  |
| --- | --- |
| Basic Data Type | int, char, float, double |
| Derived Data Type | array, pointer, structure, union |
| Enumeration Data Type | enum |
| Void Data Type | void |

|  |  |  |
| --- | --- | --- |
| **Datatype** | **Memory size** | **Range** |
| **char** | 1 byte | −128 to 127 |
| **short** | 2 byte | −32,768 to 32,767 |
| **int** | 2 byte or 4 byte | −32,768 to 32,767 |
| **float** | 4 byte |  |
| **double** | 8 byte |  |

**Enum : user defined datatype in c**

**An enumeration is a user-defined data type that consists of integral constants.**

#include<stdio.h>

enum week{Mon, Tue, Wed, Thur, Fri, Sat, Sun};

int main()

{

enum week day;

day = Wed;

printf("%d",day);

return 0;

}

//output : 2

//enum constants starts with 0. We can change the default value by assigning custom values.

enum week{Mon=5, Tue, Wed, Thur, Fri, Sat, Sun};

//starts with 5, left to right

**Note ;** No boolean datatype in **stdio**, so use stdbool header to use bool for boolean datatype.

#include <stdbool.h>

bool t = true;

printf("%d\n",t);

**Format specifier :**

Format specifiers can be defined as the operators which are used in association with printf() function for printing the data that is referred by any object or any variable.

|  |  |
| --- | --- |
| **Format specifier** | **Description** |
| %d | Integer Format Specifier |
| %f | Float Format Specifier |
| %c | Character Format Specifier |
| %s | String Format Specifier |
| %x | Hexadecimal format specifier, Ex: address of variable |

Example of format specifier

printf("the number is %d \n",123);

printf("the message is %s \n","hello world");

printf("the decimal number %f is \n", 34.6);

printf("last alphabet is %c ",'z');

Output :

the number is 123

the message is hello world

the decimal number 34.6

last alphabet is z

**Variables**

Used to store data.

1. Variable name must not start with a digit.
2. Variable name can consist of alphabets, digits and special symbols like underscore \_.
3. Blank or spaces are not allowed in variable name.
4. Keywords are not allowed as variable name.
5. Upper and lower case names are treated as different, as C is case-sensitive, so it is suggested to keep the variable names in lower case.