

C Programming

Data Types

Examples of Data Types

Type	Example	Meaning
Integer	8, 100, -25	Whole numbers, both positive and negative
Float (floating-point number)	0.44, -12.34	Numbers with fractional/decimal parts
Character	'x', 'A', '4'	Any character inside single quotes
String	"abc", "A100"	One or more characters inside double quotes

Note:

Indentation - use the tab key and NOT the spacebar when you indent any code inside its own set of curly-brackets.

Variables

A variable is a piece of memory set aside in a program to store data. Let's take a look at the following code:

Type	Data Type	Example	Meaning
Integer	int	8, 100, -25	Whole numbers, both positive and negative
Float (floating-point number)	float	0.44, -12.34	Numbers with fractional/decimal parts
Character	char	'x', 'A', '4'	Any character inside single quotes
String	Find out later !	"abc", "A100"	One or more characters inside double quotes

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

```
int main()
{
    int var1;
    float var2;
    char var3;

    var1 = 10;
    var2 = 8.21;
    var3 = 'q';

    return 0;
}
```

Repl: <https://replit.com/@michaelTUDublin/Assignment-Operation#main.c>

Assignment operation

When you assign some data into a variable
e.g.,

```
int number;
number = 10;
```

Rules for naming variables

1. A variable name can only be constructed using the letters a - z, A - Z, numerals 0 - 9 or an underscore _
2. A variable MUST start with a letter or an underscore
3. A variable name cannot be a reserved word in C, e.g., main
4. A variable name can contain any number of characters. However, it is advisable to keep variable names as short as possible.

e.g.,

month_1_sales	Valid
month1sales	Valid

1st_month_sales	Invalid
char	Invalid

Keyboard output: Simple output to the screen

The screen is technically called "Standard Output"

To output to Standard Output, C uses the **printf()** statement

```
printf(".....");
```

A Delimiter. A Delimiter is represented by a % followed by a character. Each data type has its own delimiter to display the contents of a variable

Delimiter	Type
%d	Displays an integer
%f	Displays a float
%c	Displays a character

The Escape character

The Escape character is represented by a backslash, i.e., \

It has a number of different functions, e.g.,

```
\n    print a new line to standard output
\t    display a tab space to standard output
```

(See code below)

Code

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

```
int main()
```

```
{
```

```
    int var1;
```

```
    float var2;
```

```
    char var3;
```

```
    var1 = 10;
```

```
    var2 = 8.21;
```

```
    var3 = 'q';
```

```
    printf("var1 contains %d\nvar2 contains %f\nvar3 contains  
%c", var1, var2, var3);
```

```
    return 0;
```

```
}
```

Repl: <https://replit.com/@michaelTUDublin/Standard-Output-printf#main.c>