Welcome,

PROGRAMMERS

01

First program

Let’s discover our first C language program

01.

Basic Structure

Basic structure of C Lang?

BAsic structure of C Language

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

main()
{
  clrscr();

  printf("Hello World");

  getch();
}
```

Let’s break down All parts of program…

Pre-Processor

Library/Header Files

Built-in Function

Built-in Functions

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

main()
{

  clrscr();

  printf("Hello World");

  getch();
}
```

Library Files

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

main()
{
  clrscr();

  printf("Hello World");

  getch();
}
```

#include<stdio.h>

Standard input output header file

From this library, **printf()** function is imported.

Means **compiler finds meaning** of printf() function from this **stdio.h** file.

Library Files

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

main()
{
  clrscr();

  printf("Hello World");

  getch();
}
```

#include<conio.h>

Console input output header file

From this library, **clrscr() & getch()** functions are imported.

Means **compiler finds meaning** of clrscr() & getch() functions from this **conio.h** file.

But, did you notice # symbol?

pre-processor

is called as **Pre-processor**.

This **tells the compiler** that first pre-process (executes) the **process of importing a library/header file**.

For example,

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

functions

1

main()

Entry-point of our program.

clrscr()

Clears the output screen.

2

3

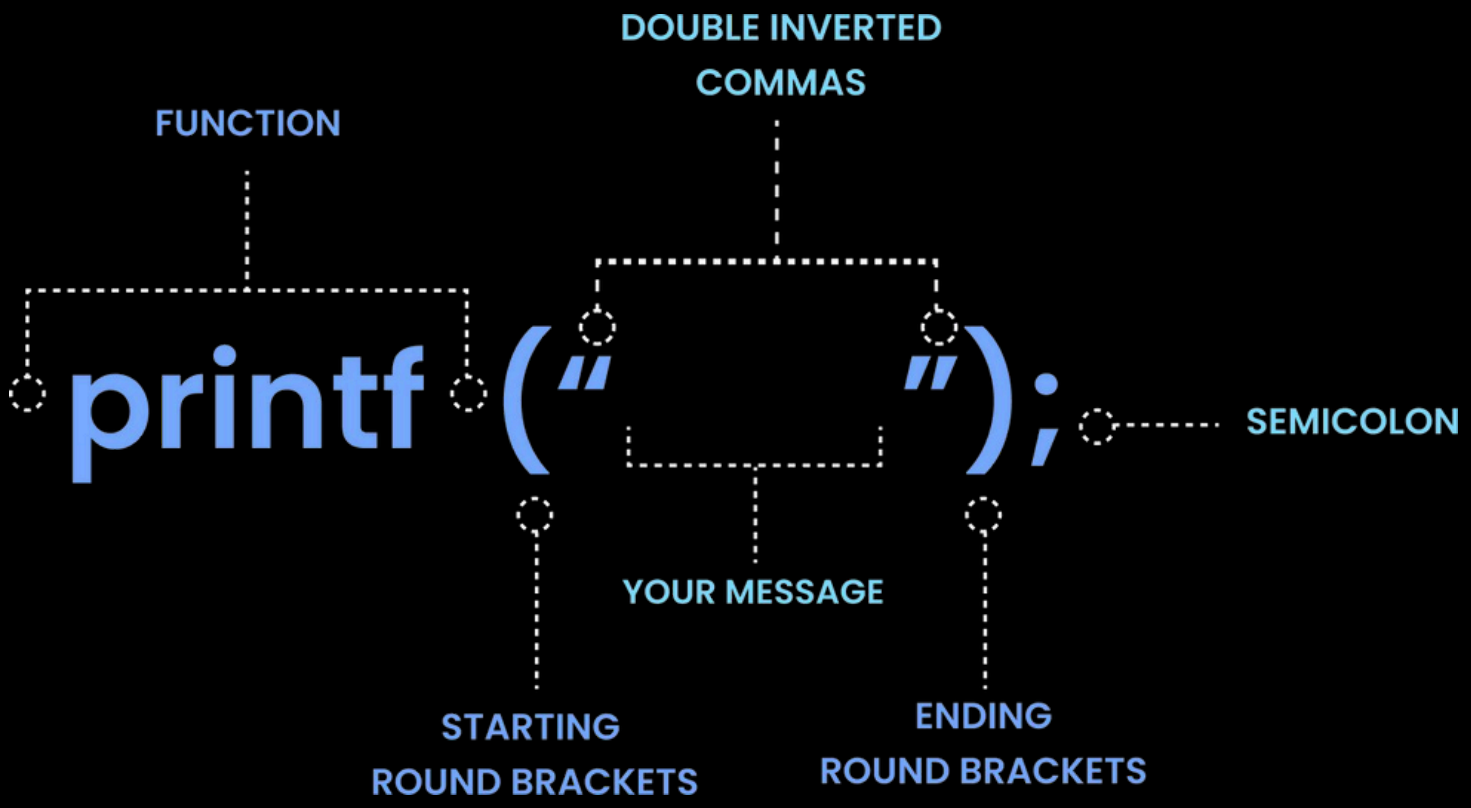
printf()

Prints any message in output screen.

getch()

Takes user input and then close the output screen.

4


DOUBLE INVERTED
COMMAS
FUNCTION
printf (" ");
SEMICOLON
YOUR MESSAGE
STARTING
ROUND BRACKETS
ENDING
ROUND BRACKETS

02

Escape sequence characters

Escape Sequence Characters

INTRODUCTION

An escape sequence contains a **backslash** (\) symbol followed by one of the escape sequence characters.

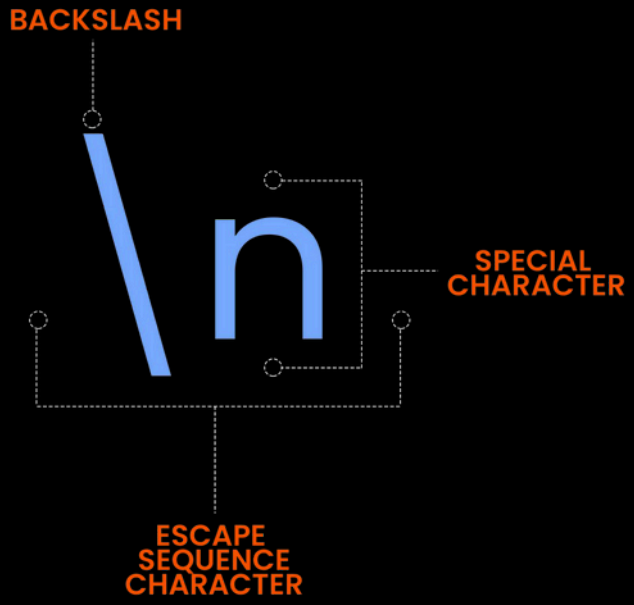
Escape sequence characters

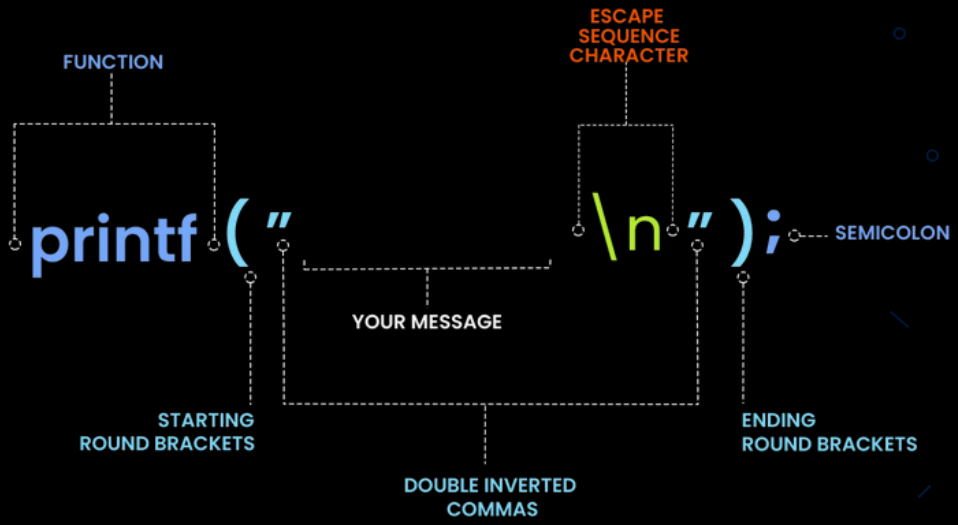
new-line

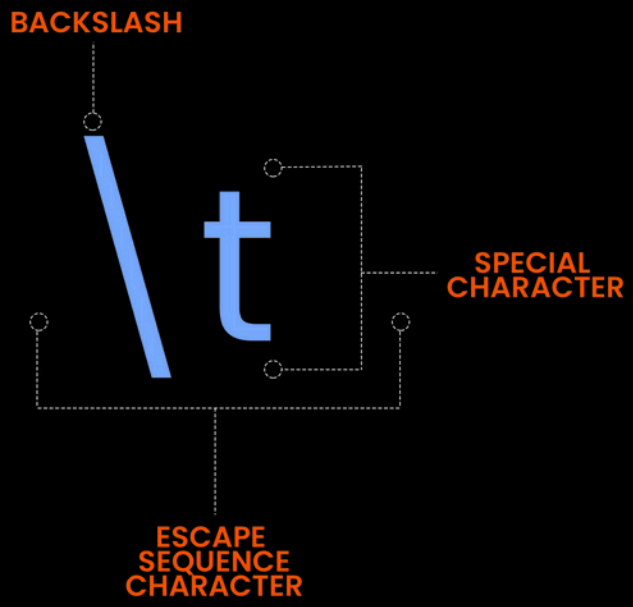
This character enters the cursor into new line.

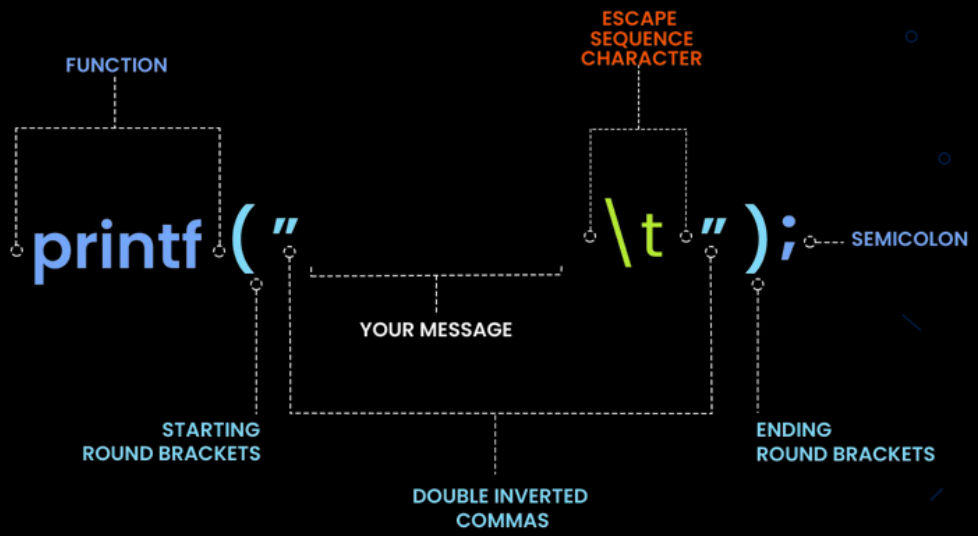
Horizontal-space / tab

This character leaves total 8 spaces from the beginning of character.


BACKSLASH
\n
SPECIAL CHARACTER
ESCAPE SEQUENCE CHARACTER


FUNCTION
ESCAPE SEQUENCE CHARACTER
printf(" \n ");
SEMICOLON
YOUR MESSAGE
STARTING ROUND BRACKETS
DOUBLE INVERTED COMMAS
ENDING ROUND BRACKETS


BACKSLASH
\t
SPECIAL CHARACTER
ESCAPE SEQUENCE CHARACTER


FUNCTION
ESCAPE SEQUENCE CHARACTER
printf(" \t ");
SEMICOLON
YOUR MESSAGE
STARTING ROUND BRACKETS
ENDING ROUND BRACKETS
DOUBLE INVERTED COMMAS

Let’s code