C Library - printf() function

The C library **`printf()`** function is a fundamental tool for outputting formatted text to the standard output stream. It allows for versatile printing of variables, strings, and other data types.

Syntax

Following is the C library syntax of the **`printf()`** function −

```
int printf(const char *format, ...)
```

Parameters

Following is the list of parameters −

- **format :** A string that may contain format specifiers like %d, %s, etc., which control the formatting of subsequent arguments.
- **... :** A variable number of arguments to be formatted and printed according to the format string.

Return value

Returns the number of characters printed (excluding the null byte used to end the output to strings) if successful.On error, it returns a negative value.

Example 1: Printing Integer and String

Here, printf() prints an integer and a string using format specifiers %d and %s respectively.

Below is the illustration of the C library **`printf()`** function.

Open Compiler

```c
#include <stdio.h>

int main() {
   int num = 10;
   char str[] = "Hello";

   printf("Integer: %d, String: %s\n", num, str);

   return 0;
}
```

Output

The above code produces following result −

```
Integer: 10, String: Hello
```

Example 2: Printing Octal and Hexadecimal Numbers

Here, the printf() prints an octal number (octal_num) using %o format specifier and a hexadecimal number (hex_num) using %X format specifier.

Open Compiler

```c
#include <stdio.h>

int main() {
   // Octal representation of 61
   int octal_num = 075;

   // Hexadecimal representation of 31
   int hex_num = 0x1F;

   printf("Octal: %o, Hexadecimal: %X\n", octal_num, hex_num);
```

```
    return 0;
}
```

Output

After execution of above code, we get the following result −

```
Octal: 75, Hexadecimal: 1F
```