**String Formatting**

# Escape Characters

Escape or non-printable characters that can be represented with backslash notation.

An escape character gets interpreted; in a single quoted as well as double quoted strings.

|  |  |  |
| --- | --- | --- |
| Backslash notation | Hexadecimal character | Description |
| \a | 0x07 | Bell or alert |
| \b | 0x08 | Backspace |
| \cx |  | Control-x |
| \C-x |  | Control-x |
| \e | 0x1b | Escape |
| \f | 0x0c | Form feed |
| \M-\C-x |  | Meta-Control-x |
| \n | 0x0a | Newline |
| \nnn |  | Octal notation, where n is in the range 0.7 |
| \r | 0x0d | Carriage return |
| \s | 0x20 | Space |
| \t | 0x09 | Tab |
| \v | 0x0b | Vertical tab |
| \x |  | Character x |
| \xnn |  | Hexadecimal notation, where n is in the range 0.9, a.f, or A.F |

# String Formatting Operator

Python's format operator % is unique to strings and makes up for the pack of having functions from C's printf() family.

#!/usr/bin/python3

print("My name is %s and weight is %d kg!" %('Zara', 21))

Output:

My name is Zara and weight is 21 kg!

|  |  |  |
| --- | --- | --- |
| S. No. | Format Symbol &  Conversion |  |
| 1 | **%c** | Character |
| 2 | **%s** | string conversion via str() prior to formatting |
| 3 | **%i** | signed decimal integer |
| 4 | **%d** | signed decimal integer |
| 5 | **%u** | unsigned decimal integer |
| 6 | **%o** | octal integer |
| 7 | **%x** | hexadecimal integer (lowercase letters) |
| 8 | **%X** | hexadecimal integer (UPPERcase letters) |
| 9 | **%e** | exponential notation (with lowercase 'e') |
| 10 | **%E** | exponential notation (with UPPERcase 'E') |
| 11 | **%f** | floating point real number |
| 12 | **%g** | the shorter of %f and %e |
| 13 | **%G** | the shorter of %f and %E |

# Symbols and Functionality

|  |  |  |
| --- | --- | --- |
| S. No. | Symbol | Functionality |
| 1 | **\*** | argument specifies width or precision |
| 2 | **-** | left justification |
| 3 | **+** | display the sign |
| 4 | **<sp>** | leave a blank space before a positive number |
| 5 | **#** | add the octal leading zero ( '0' ) or hexadecimal leading '0x' or '0X', depending on whether 'x' or 'X' were used. |
| 6 | **0** | pad from left with zeros (instead of spaces) |
| 7 | **%** | '%%' leaves you with a single literal '%' |
| 8 | **(var)** | mapping variable (dictionary arguments) |
| 9 | **m.n.** | m is the minimum total width and n is the number of digits to display after the decimal point (if appl.) |