# Lecture-11

**String Formatting Operator:**

One of Python's coolest features is the string format operator %. This operator is unique to strings and makes up for the pack of having functions from C's printf() family. Following is a simple example –

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

When the above code is executed, it produces the following result –

```
My name is Zara and weight is 21 kg!
```

Here is the list of complete set of symbols which can be used along with % –

| Format Symbol | Conversion |
|---|---|
| %c | character |
| %s | string conversion via str() prior to formatting |
| %i | signed decimal integer |
| %d | signed decimal integer |
| %u | unsigned decimal integer |
| %o | octal integer |
| %x | hexadecimal integer (lowercase letters) |
| %X | hexadecimal integer (UPPERcase letters) |

| Format Symbol | Conversion |
|---|---|
| %e | exponential notation (with lowercase 'e') |
| %E | exponential notation (with UPPERcase 'E') |
| %f | floating point real number |
| %g | the shorter of %f and %e |
| %G | the shorter of %f and %E |

**The below example describes the use of formatting using % in Python**

```
# Python program to demonstrate the use of formatting using %
# Initialize variable as a string
variable = '15'
string = "Variable as string = %s" %(variable)
print string
# Printing as raw data
# Thanks to Himanshu Pant for this
print "Variable as raw data = %r" %(variable)
# Convert the variable to integer
# And perform check other formatting options
 variable = int(variable) # Without this the below statement  will give error.
string = "Variable as integer = %d" %(variable)
print string
print "Variable as float = %f" %(variable)
```

```
# printing as any string or char after a mark
# here i use mayank as a string
print "Variable as printing with special char = %cmayank" %(variable)
  print "Variable as hexadecimal = %x" %(variable)
print "Variable as octal = %o" %(variable)
```

**Output :**

Variable as string = 15V

ariable as raw data = '15‘

Variable as integer = 15

Variable as float = 15.000000

Variable as printing with special char = mayank

Variable as hexadecimal = f

Variable as octal = 17

# Triple Quotes:

Python's triple quotes comes to the rescue by allowing strings to span multiple lines, including verbatim NEWLINEs, TABs, and any other special characters.

The syntax for triple quotes consists of three consecutive **single or double** quotes.

```
para_str = """this is a long string that is made up of several lines
  and non-printable characters such as TAB (\t ) and they will show
  up that way when displayed. NEWLINEs within the string, whether
  explicitly given like this within the brackets [ \n ], or just a
  NEWLINE within the variable assignment will also show up.""“
```

print para_str

When the above code is executed, it produces the following result. Note how every single special character has been converted to its printed form, right down to the last NEWLINE at the end of the string between the "up." and closing triple quotes. Also note that NEWLINEs occur either with an explicit carriage return at the end of a line or its escape code (\n) −

```
this is a long string that is made up ofseveral lines and non-printable characters
  such asTAB (    ) and they will show up that way when displayed.NEWLINEs within
  the string, whether explicitly given likethis within the brackets [

 ], or just a NEWLINE withinthe variable assignment will also show up.
```

## References:

1. Fundamentals of Python first Programmes by Kenneth A Lambert, Copyrighted material Course Technology Inc. 1 st edition (6th February 2009)
2. https://www.tutorialspoint.com/python/index.htm
3. https://www.geeksforgeeks.org/python-programming-language

# ****END OF THE LECTURE***

# ***THANK YOU***