Assignment on python Basic

Name : Lovenish Gaur, Date : 18, May,2020

1. For loops & While Loops With two examples each
2. If Statement, If else, If elif..
3. Creating user defined functions
4. Escape Sequence in python
5. Python operators
6. Working on Lambda functions
7. Python modules

Loops ¶

In [1]:

```
# For loops : A for loop is used for iterating over a sequence (that is either a lis
#             a tuple, a dictionary, a set, or a string).

Cars = ["Ford", "Corvette", "Hundayi"]
for x in Cars:
  print(x)
```

```
Ford
Corvette
Hundayi
```

In [2]:

```
Rate = [2, 4, 2, 3, 4]
output = (2*n * 2
          for n in Rate)
tuple(output)
```

Out[2]:

```
(8, 16, 8, 12, 16)
```

In [3]:

```
# While loops : execute a set of statements as long as a condition is true.


a = 3 # Assigning value to a variable a
while a < 9: # Checking the condition till the time a is less than 9
    print(a) # print the value of a
    a = a+1 # increment a till a is less than 9
```

```
3
4
5
6
7
8
```

In [4]:

```
a = 0 # Assigning value to a variable a
while a < 9: # Checking the condition
  print(a) # Print the value of a
  if a == 5: # if a is equal to 3 come out of the loop
    break # for breaking the loop using the statement break
  a += 1 # Increment
```

```
0
1
2
3
4
5
```

If Statement, If else, If elif..

In [36]:

```


a = 10
b = 9
if b < a:
  print("b is less than a by:", a-b)
```

```
b is less than a by: 1
```

In [37]:

```
a = "green"
b = "red"
if b == a:
  print("a is green")
else:
    print("b is red")
```

```
b is red
```

In [38]:

```
a = "green"
b = "red"
c = "Yellow"
if b == a:
  print("a is green")
elif b==c:
    print("b is red")
else:
    print("c is Yellow")
```

```
c is Yellow
```

In [39]:

```
# Creating user defined functions

# Here we are creating a function which as per the input color from the user will ma

def color(x):
    a = "Green"
    b = "Red"
    c = "Yellow"
    d = "Blue"
    if x ==a:
        print ("Let's plant a tree")
    elif x == b:
        print ("There will be strom")
    elif x == c:
        print ("This is autumn leaves are shedding")
    else:
        print("The sky is clear and Blue")
    return


z = input("Please choose a color from Red, Yellow, Blue and Green:")
```

```
Please choose a color from Red, Yellow, Blue and Green:
```

In [40]:

```
color(z)
```

```
The sky is clear and Blue
```

Escape Sequence in python

1. ' : Single Quote
2. \ : Backslash
3. \n : New Line
4. \r : Carriage Return
5. \t : Tab
6. \b : Backspace

In [41]:

```
X = "This is the new \"India\" developing and growing."
print(X)
```

```
This is the new "India" developing and growing.
```

In [42]:

```
msg = 'It\'s alright.'  # we need the apostrophe in it's
print(msg)
```

```
It's alright.
```

In [43]:

```
a = "This will insert one \\ (backslash)."
print(a)
```

```
This will insert one \ (backslash).
```

In [44]:

```
a = "Morning\nAfternoon\nEvening!"
print(a)
```

```
Morning
Afternoon
Evening!
```

In [45]:

```
a = "Morning\r Afternoon\r  Evening!"
print(a)
```

```
  Evening!
```

In [46]:

```
a = "Morning\t Afternoon"
print(a)
```

```
Morning  Afternoon
```

In [47]:

```
a = "Morning \bAfternoon"
print(a)
```

```
MorningAfternoon
```

Python Operators

Operators are used to perform operations on variables and values.

Python divides the operators in the following groups:

1. Arithmetic operators
2. Assignment operators
3. Comparison operators
4. Logical operators
5. Identity operators
6. Membership operators

In [82]:

```
#Arithmetic operators
x = 3
y = 4
x + y #Addition
```

Out[82]:

```
7
```

In [83]:

```
x-y #Subtraction
```

Out[83]:

```
-1
```

In [84]:

```
x*y #Multiplication
```

Out[84]:

```
12
```

In [85]:

```
x/y #Division
```

Out[85]:

```
0.75
```

In [86]:

```
x%y #Modulus
```

Out[86]:

```
3
```

In [87]:

```
x**y #Exponentiation
```

Out[87]:

```
81
```

In [88]:

```
x//y #Floor division
```

Out[88]:

```
0
```

In [89]:

```
#Comparison Operators

x = 5
y = 9
```

In [90]:

```
x > y
```

Out[90]:

```
False
```

In [91]:

```
x < y
```

Out[91]:

```
True
```

In [92]:

```
x == y
```

Out[92]:

```
False
```

In [93]:

```
x != y
```

Out[93]:

```
True
```

In [94]:

```
x >= y
```

Out[94]:

```
False
```

In [95]:

```
x <= y
```

Out[95]:

```
True
```

In [96]:

```
#Logical Operators

x = 6
y = 10
```

In [97]:

```
x >4 and y <11
```

Out[97]:

```
True
```

In [98]:

```
x >4 or y <9
```

Out[98]:

```
True
```

In [99]:

```
not(x >4 and y <11)
```

Out[99]:

```
False
```

In [100]:

```
#Identity Operators

x  = "Green"
Y = "Red"
```

In [101]:

```
x is y
```

Out[101]:

```
False
```

In [102]:

```
X is not y
```

Out[102]:

```
True
```

In [103]:

```
#Membership Operators
x = [1,3,5,6,7,8]
y = 3
```

In [104]:

```
y in x
```

Out[104]:

```
True
```

In [105]:

```
y not in x
```

Out[105]:

```
False
```

Lambda functions

A lambda function is a small anonymous function.

A lambda function can take any number of arguments, but can only have one expression.

In [114]:

```
x = lambda a,b,c : a + b * c  + 10
print(x(2,3,4))
```

```
24
```

In [ ]:

In [ ]: