# PYTHON

## From Simple to Complex
With
## Examples

AYESHA NOREEN

Bachelor's in Software Engineering,

Master's in Computer Science

from COMSATS University, Islamabad

# NOTE!!!

In these notes Screenshots of practice examples and coding are added. The code files are also available in code folder that contain .ipynb files that are created on Jupyter notebook.

# Chapter6
# Python Operators

Some python operators are:

- Arithmetic operators
- Assignment operators
- Comparison operators
- Logical operators
- Identity operators
- Membership operators
- Bitwise operators

# Arithmetic Operators

Arithmetic operators are:

- Addition (+)
- Subtraction (-)
- Multiplication (*)
- Exponential (**)
- Division (/)
- Floor division (//)
- Modulus (%)

```
#Arithmetic operators
x=5
y=10
print('Addition: ',x+y)
print('Subtraction: ',x-y)
print('Multiplication: ',x*y)
print('Exponential: ',x**y)
print('Division: ',x/y)
print('Floor division: ',x//y)
print('Modulus: ',x%y)
```

```
Addition:  15
Subtraction:  -5
Multiplication:  50
Exponential:  9765625
Division:  0.5
Floor division:  0
Modulus:  5
```

# Assignment Operators

Assignment operators are:

- = (assignment)
- += (add and assign)
- -= (subtract and assign)
- *= (multiply and assign)
- **= (exponent and assign)
- /= (divide and assign)
- //= (floor division and assign)

```
#Assignment operator
x=5
print(x)
x+=5      #x=x+5
print(x)
x-=5      #x=x-5
print(x)
x*=5      #x=x*5
print(x)
x**=5     #x=x**5
print(x)
x/=5      #x=x/5
print(x)
x//=5     #x=x//5
print(x)
```

```
5
10
5
25
9765625
1953125.0
390625.0
```

# Assignment Operators

Assignment operators are:

- %= (modulus and assign)
- &= (AND and assign)
- |= (OR and assign)
- ^= (XOR and assign)
- >>= (logical shift left)
- <<= (logical shift right)

```
x=51      #51/5= 10 and 1 remainder
x%=5      #x=x%5
print(x) #1
```

```
1
```

```
#it return and of corresponding binary numbers
x = 5    #x=101
x &= 3   #x=x&3 ==> 101&011 ==>001 (1)
print(x)
```

```
1
```

```
#it return or of corresponding binary numbers
x = 5    #x=101
x |= 3   #x=x|3 ==> 101|011 ==>111 (7)
print(x)
```

```
7
```

```
#it return xor of corresponding binary numbers
#(for same values 0 for different values 1)
x = 7    #x=111
x ^= 4   #x=x^4 ==> 111^100 ==>011 (3)
print(x)
```

```
3
```

```
#logical shift left and fill zeros at right
x=5    #101
x<<=5  #10100000 (=160)
print(x)
```

```
160
```

```
#logical shift right
x=127 #1111111
x>>=5 #11 (3)
print(x)
```

```
3
```

# comparison Operators

Comparison operators are:

- Equal (==)
- Not equal (!=)
- Greater than (>)
- Less than (<)
- Greater than equal (>=)
- Less than equal (<=)

```
#Comparision operators
x=5
y=6
print(x==y)
print(x!=y)
print(x>=y)
print(x<=y)
print(x>y)
print(x<y)
```

```
False
True
False
True
False
True
```

## logical Operators

Logical operators are:

- AND operator
- OR operator
- NOT operator

```
x=2
print(x > 3 and x < 10) #True if both are True
print(x > 3 or x < 10)  #True if any one is True
print(not(x>1 and x<3)) #invert the output
```

```
False
True
False
```

## identity Operators

- Is operator
- IS NOT operator

```
x = 'a'
y = 5
z = x
print(x is z) # returns True because z is the same object as x
print(x is y) # returns False because x is not the same object as y
print(x is not z) # returns False because z is the same object as x
print(x is not y) # returns True because x is not the same object as y
```

```
True
False
False
True
```

## Membership Operators

Membership operators are:

- In
- Not in

## Bitwise Operators

Bitwise operators are:

- & (AND)
- | (OR)
- ^ (XOR)
- ~ (NOT)
- << (Shift left)
- >> (shift right)

```
x = "banana"
print("a" in x)
print("a" not in x)
```

```
True
False
```

```
x=5  #101
y=3  #011
print(x&y) #101&011=1
print(x|y) #101|011=111(7)
print(x^y) #101|011=110(6)
print(~y)  #100 =-(4)
print(x<<y) #logical shift left 101 by 3==>101000(40)
print(x>>y) #logical shift right 101 by 3==>0
```

```
1
7
6
-4
40
0
```