TOPIC : Artificial Intelligence with Python - INMOVIDU - Jahnavi N

GETTING STARTED WITH PYTHON PROGRAMMING

LOGICAL OPERATORS

Logical operators in Python are used for conditional statements are true or false. For AND operator – It returns TRUE if both the operands (right side and left side) are true For OR operator- It returns TRUE if either of the operand (right side or left side) is true For NOT operator- returns TRUE if operand is false

In [1]:

```
a=12
b=23
print(a<b and b>a)
```

```
True
```

In [2]:

```
a=12
b=23
print(a<b or b<a)
```

```
True
```

In [3]:

```
print(not a>b)
```

```
True
```

MEMBERSHIP OPERATORS

"in" TRUE if a value/sequence is present in an object A in B "not in" TRUE if a value/sequence is NOT present in an object A not in B

In [4]:

```
print( 't' in 'aspirants')
```

```
True
```

In [5]:

```
print( "0" in "aspirants")
```

```
False
```

In [6]:

```
print("k" not in "Yup")
```

```
True
```

IDENTITY OPERATORS

The identity operators in Python are used to determine whether a value is of a certain class or typeis TRUE if both variables are actually the same object A is B is not TRUE if both variables are actually NOT the same object A is notB

In [7]:

```
x = 5.2323232
print(type(x) is float)
```

```
True
```

In [8]:
```
x=12
y="op"
print(type(x) is type(y))
```

```
False
```

In [9]:
```
print(type(89) is not type("ha"))
# int is  str
```

```
True
```

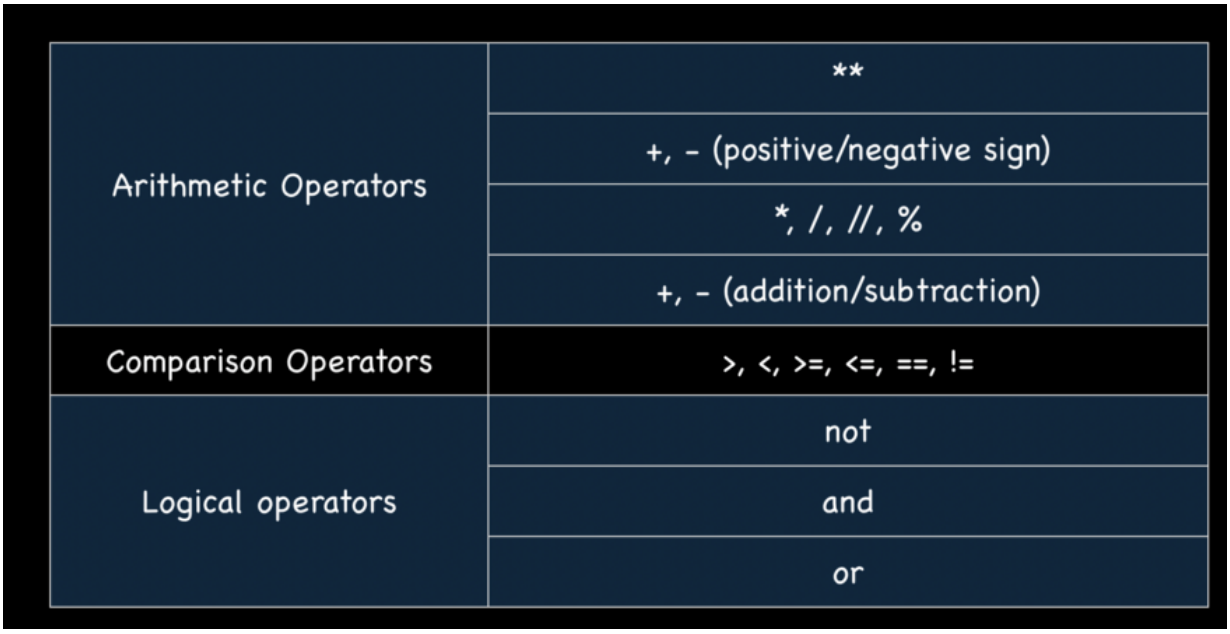
In [10]:
```
print(type("hai") is type('a'))
#(str is str)
```

```
True
```

7) PRECEDENCE

In [11]:
```
from IPython import display
display.Image("pre.png")
```

Out[11]:



Arithmetic Operators	**
	+, - (positive/negative sign)
	*, /, //, %
	+, - (addition/subtraction)
Comparison Operators	>, <, >=, <=, ==, !=
Logical operators	not
	and
	or

In [12]:
```
print(5**(2+1))
```

```
125
```

In [13]:
```
print(5*4+2)
```

```
22
```

In [14]:
```
print(5+2*4)
```

```
13
```

8) Intendation

Indentation refers to the spaces at the beginning of a code line. Where in other programming languages the indentation in code is for readability only, the indentation in Python is very important.This intendation identifies the blocks it shows error if you skip any intendations

In [15]:

```
print("hello")
print("people")
```

```
hello
people
```

11) TAKING INPUTS

In [16]:

```
name=input("enter name ")
print("name is ",name)
```

```
enter name jahnavi
name is  jahnavi
```

In [17]:

```
x=float(input("enter number "))
```

```
enter number 23.33
```

In [18]:

```
print(type(x),x)
```

```
<class 'float'> 23.33
```

In [19]:

```
x,y=2,3
print(x)
print(y)
```

```
2
3
```

In [21]:

```
x,y=input("enter ").split(",")
```

```
enter 23,34
```

In [22]:

```
print(x,y)
```

```
23 34
```

9) CONTROL STATEMENTS

Conditional Statement in Python perform different computations or actions depending on whether a specific Boolean constraint evaluates to true or false.

we use key words if ,else and elif

In [ ]:

```
#if statement
```

In [23]:

```
v=int(input("enter "))
if(v==100):
    print("it is 100")
```

```
enter 100
it is 100
```

In [26]:

```
x=10
if(x>5):
    x=x+5
print(x)
```

```
15
```

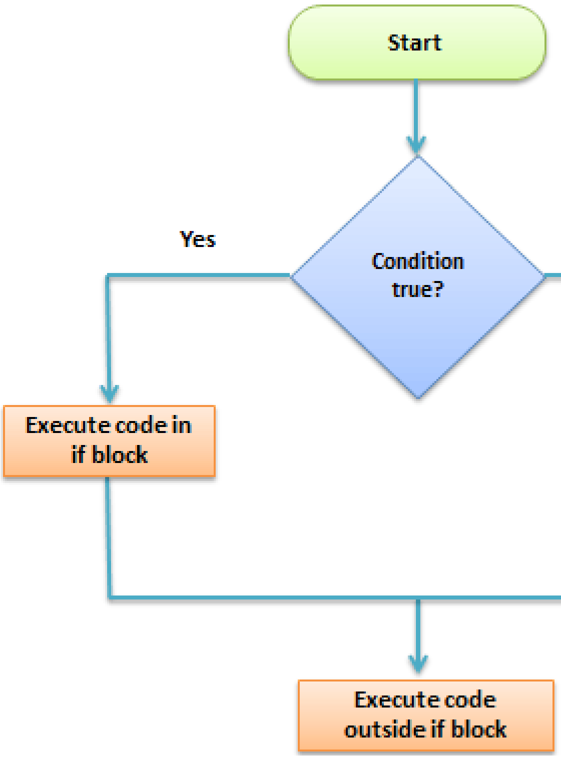
In [27]:

```
#if - else statement
```

In [24]:

```
from IPython import display
display.Image("https://www.guru99.com/images/2013/04/if_then_flowchart.png")
```

Out[24]:



In [25]:

```
a=20
b=50
if(a>b):
    print("a >b")
else:
    print("B<a")
```

```
B<a
```

In [28]:

```
v=2000
if(v>=100 or v<500):
    print("print")
else:
    print("not")
```

```
print
```

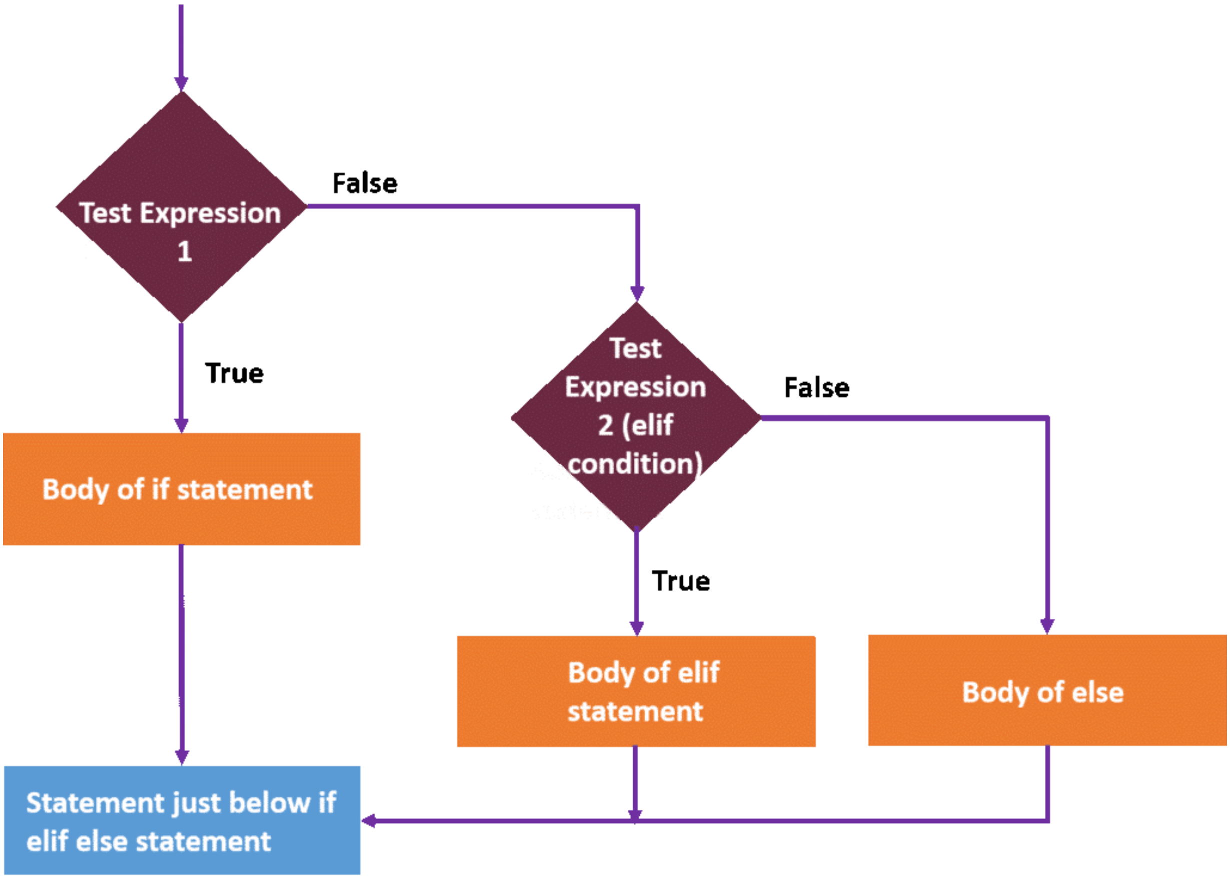
In [ ]:

```
# NESTED IF  " else -if " , "elif "
```

In [29]:

```
display.Image("iff.png")
```

Out[29]:



In [30]:

```
a=17
b=22
if(a==10):
    print("its 10")
elif(a==12):
    print("12 ")
```

elif is short name for else - if statment so above code could be written as ..

In [31]:

```
a=17
b=22
if(a==10):
    print("its 10")
else:
    if(a==12):
        print("12 ")
    else:
        print("not 10 and 12")
```

```
not 10 and 12
```

In [ ]: