Indentation

- Python relies on indentation (whitespace at the beginning of a line) to define scope in the code. Other programming languages often use curly-brackets for this purpose.

In [4]:
```python
var1=55 # True or False
if var1>40:
    print('50 is greater than 40')
```
```
50 is greater than 40
```

In [6]:
```python
a=-2
if a<0:
    print('its positive number')
    print('another statement')
    print('3rd statement')
else:
    print('its Negative number')
```
```
its positive number
another statement
3rd statement
```

In [7]:
```python
x=55
if x%2 == 0:
    print('x is even')
else :
    print('x is odd')
```
```
x is odd
```

In [8]:
```python
x=10
y=66
if x <=y:
    print('x is less than y')
elif x >y:
    print('x is greater than y')
else:
    print('x and y are equal')
```
```
x is less than y
```

In [10]:
```python
A=900
a=500
b=800
if A<b:
    print("A is greater than B : ",A)
    print("this is another statement")
    print("this 3rd statement")
else:
    print("this is else statement")
```
```
this is else statement
```

In [12]:
```python
a=33
b=66
if a>b:
    print('A value is Greater than B value : ',a)
else:
    print("its Else Blob: A is Less Than B : ",b)
```
```
its Else Blob: A is Less Than B :  66
```

In [13]:
```python
a = 99
b = 88
if b > a:
    print("b is greater than a and B Value is : " ,b)
else:
    print(' A is Greater than B And A value is : ',a)
```
```
 A is Greater than B And A value is :  99
```

In [14]:
```python
#If statement, without indentation (will raise an error):
a = 33
b = 200
if b > a:
    print("b is greater than a") # you will get an error}
else:
    print('this is else')
```
```
b is greater than a
```

Elif

- The elif keyword is pythons way of saying "if the previous conditions were not true, then try this condition"

In [16]:
```python
a = 100

if a==200:
    print("a value is 200")
elif a>200:
    print("a value is more than 200")
elif a<200:
    print('a value is less than 200')
else:
    print('last else statements')
```
```
a value is less than 200
```

Else

- The else keyword catches anything which isn't caught by the preceding conditions.

In [17]:
```python
a = 200
b = 33
if b > a:
    print("b is greater than a")
elif a == b:
    print("a and b are equal")
else:
    print("a is greater than b",a)
```
```
a is greater than b 200
```

In [18]:
```python
a = 200
b = 33
if b > a:
    print("b is greater than a")
else:
    print("b is not greater than a")
```
```
b is not greater than a
```

- **Short Hand If**
- If you have only one statement to execute, you can put it on the same line as the if statement.

In [0]:
```python
a=99
b=88
if a < b: print("a is greater than b")
```

- **Short Hand If ... Else**
- If you have only one statement to execute, one for if, and one for else, you can put it all on the same line:

In [19]:
```python
a = 700
b = 550
print('A is greater than b Statement',a) if a > b else print('a is less than b Statement',b)
```
```
A is greater than b Statement 700
```

- One line if else statement, with 3 conditions:
- This technique is known as Ternary Operators, or Conditional Expressions.
- Ternary operators also known as conditional expressions are operators that evaluate something based on a condition being true or false. It was added to Python in version 2.5. It simply allows to test a condition in a single line replacing the multiline if-else making the code compact.

In [20]:
```python
a = 400
b = 400
print("A Greater Than B : ",b) if a > b else print("its a equal to B =") if a == b else print("B is less Than A",a)
```
```
its a equal to B =
```

In [21]:
```python
#Test if a is greater than b, AND if c is greater than a
a = 600
b = 563
c = 670
if a > b and b > c:
    print("Both conditions are True")
else:
    print('Both conditions are not True...')
```
```
Both conditions are not True...
```

In [22]:
```python
a = 200
b = 33
c = 500
if (a > b and  a > c):
    print("At least one of the conditions is True")
else:
    print("else statement ")
```
```
else statement
```

In [24]:
```python
a = 200
b = 33
c = 500
if a > b or a > c:
    print("One of the conditions is True")
```
```
One of the conditions is True
```

In [25]:
```python
a=55

if a<10:
    pass
else:
    print('this is sample pass statement')
```
```
this is sample pass statement
```

In [26]:
```python
a=55
if a > 10:
    pass
else:
    pass
```

- **The pass Statement**
- if statements cannot be empty, but if you for some reason have an if statement with no content, put in the pass statement to avoid getting an error.

In [27]:
```python
a = 33
b = 200

if b > a:
    pass
```