This is python tutorial

Jupyter Shortcuts

- Esc --> To make cell blue,
- y --> To make cell codeable
- m --> To make cell markdown
- $a + b = c$ --> To write arithmetic formulas (must start and end with $ sign)
- use ipython as a calculator, 1) open anaconda editor 2) type "ipython"
- Enter few characters of a variable, and press '*tab*' to autoshow options
- Click on any function, then press '*shift+tab*' to show the main functionality of a function

This is out first program in python . Just starts Here

print ("Hellow World")

$a = b + c$

Data Types

Mutable Data Types

Objects whose value can change are said to be mutable

- List
- Dictionary
- Set

Immuable Data Types

objects whose value is unchangeable once they are created are called immutable

- Numbers (integer, bool, float, complex)
- String
- Tuple

Variables

Three properties are associated with a variable

- **Identity** --> It can be considered as an address of memory where an object is stored and cannot be changed, *id()* function return identity
- **Type** --> It defines possible values and operations. *Type()* function return the type of a variable
- **Value** --> Actual Data stored in a variable

In [17]:
```
 a = 5
 b = 23.6
c = a>b                    # Bool Variable c
print (c)                     # Print value of bool variable c
print (type(c))               #print the type of variable c
d = (-1 == True)
print ("the value of d is", d)
e = 1+ True
f = 1+ False
print ("The value of e is",e)
print("The value of e is", f)
# g = (0 == (False+False))
# print ("The Value of g is",g)
# g = (0.0 == (False+False))
# print ("The Value of g is",g)
g = ((None) == (False+False))
print ("The Value of g is",g)

abcdefgh = 233
print (a)
print (b)
# %whos
print (type(b))
x,y,z = 2, 3.9, "My Name is Muhammad Iqbal"
%whos
print("The value of a is", a, ", Type of a is", type(a),"and id of a is", id(a))
print("The value of b is", b,", Type of b is", type(b), "and id of b is",   id(b))
print(type(abcdefgh))
del z
id(x)
print(z)
```

```
False
<class 'bool'>
the value of d is False
The value of e is 2
The value of e is 1
The Value of g is False
5
23.6
<class 'float'>
Variable   Type     Data/Info
----------------------------
a          int      5
abcdefgh   int      233
b          float    23.6
bool       list     n=0
c          bool     False
d          bool     False
e          int      2
f          int      1
g          bool     False
x          int      2
y          float    3.9
z          str      My Name is Muhammad Iqbal
The value of a is 5 , Type of a is <class 'int'> and id of a is 2334218676656
The value of b is 23.6 , Type of b is <class 'float'> and id of b is 2334301998704
<class 'int'>
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14864\1417904334.py in <module>
     29 del z
     30 id(x)
---> 31 print(z)

NameError: name 'z' is not defined
```

Variables Advanced Concepts

- Note that the string object "Subject" has become orphaned object, as no variable is refering to it now. This is because the reference variable *ver1* is now pointing/refering to new object "Data Science". All orphan objects are reaped by Python Garbage Collector

In [ ]:
```
var1 = "Subject"
print (var1)
# var1[1] = 'b'
var1 = "Data Science"
print (var1)
print (id(var1))
y=complex(2,5)
print(y)
```

Operators

Jupyter Notebook

git reset text1.txt --> To unstage the changes
git checkout text1.txt --> To remove changes from file
git checkout . --> To remove changes from all files
// --> Floor Division (Divide and remove decimal from answer)
** --> To the power ofs

Type Conversions

In [19]:
```
i = '21'
print(type(i), i)
```

```
<class 'str'> 21
```

In [20]:
```
j = int(i)
print(type(j), j)
```

```
<class 'int'> 21
```

In [24]:
```
z = bool(-5)
type(z), z
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
~\AppData\Local\Temp\ipykernel_14864\2152333174.py in <module>
----> 1 z = bool(-5)
      2 type(z), z

TypeError: 'list' object is not callable
```

Comparision Operators

- **Always return Bool Value, True or False**

In [26]:
```
x= 10
y= 30
print ("x>yis:", x>y)
print ("x<yis:", x<y)
print ("x==yis:", x==y)
print ("x!=yis:", x!=y)
print ("x>=yis:", x>=y)
print ("x<=yis:", x<=y)
```

```
x>yis: False
x<yis: True
x==yis: False
x!=yis: True
x>=yis: False
x<=yis: True
```

Logical Operators

- **Always return Boolean Values, True or False**

In [27]:
```
x= True
y= False
```

```
print ("x and y is:", x and y)
print ("x or y is:", x or y)
print ("x not is:", not x)
```

```
x and y is: False
x or y is: True
x not is: False
```

In [33]:
```
numb = 3
(2 <3 and 3 <=5 ) and not (numb<0 or True)
# (2 <3 and 3 <=5 ) or not (numb<0 or True)
```

Out[33]:
```
False
```

How to handle zero in Division and Short Circut Concept

In [38]:
```
x = 10
y = 0
(x > 10) and (y==0)                    # if first condition is false then kernal te
                                       #it will not execute the second condition
# (x == 10) and (x/y)                  # To evide this type of error please use gua
(x > 10) and (y != 0) and (y/x)       # (y != 0) act as a guard, and evide to execu
```

Out[38]:
```
False
```

Identity Operators

- **Identity operators are used to compare the memory address of two variables**
- **Return true if both variables refer to same memory location**
- **'*is*','*is not*'**

In [41]:
```
x = 10
y = 8
z = 10
print (x is y)
print (x is z)
print (x is not y)
```

```
False
True
True
```

Membership Operator

- **To show the membership in sequences such as Strings, List, and Tuples**
- **'*in*', '*not in*'**

In [46]:
```
a = [1,2,3,5,6,7,'ali']
print(10 in a)
print(5 in a)
x=7
print (x in a)
rv = x in a
print(rv)
```

```
False
True
True
True
```