In [1]:
```
print("Welcome to python program")
```

```
Welcome to python program
```

1. Understand the topic definition what is <topicName> (ex: what is variable ?) 2. Understand the topic syntax - rules 3. Refer an existing examples | 4. Activity

In [ ]:
```
Python native types
======================
int float complex
bool
NoneType
Collection
-------------
    |-> Sequential - str bytes list tuple
    |-> Mapping    - dict set

python operators
python conditional statements
python looping statements
|
FileHandling => Function
```

In [2]:
```
# single line comment
print("test code1") # display the result to monitor
print('') # empty line
# print("OK-1")
print(10+20) # simple arithmetic operation
```

```
test code1

30
```

In [3]:
```
print(10)
print(10.0)
print(10+20)
print(10+20.0)
```

```
10
10.0
30
30.0
```

In [4]:
```
print(10 >5)
```

```
True
```

In [5]:
```
print(10<5)
```

```
False
```

In [6]:
```
'''Write a python program
display productName
display pid
display productcost
test productcost above 1000'''
```

```
print('productName is:pA')
print('product cost is:1500')
print('product pA id is:101')
print(1500 >1000)
```

```
productName is:pA
product cost is:1500
product pA id is:101
True
```

In [7]:
```
# variable - placeholder (or) label - holding (or) mapping an existing value

# variableName = value  <== initialization
# |__ starts with a-zA-Z_  not starts with digit ( 5var=10 - Error )
# |__ not allows space ; specialchars  (ex: $var=10 - Error  my dept='sales' - Erro

mydept = 'sales'
v5 = 120 # 5v=120 - Error
MYDEPT = 'production'
First_last_name='Mr.Raj Kumar paul' # OK
```

In [10]:
```
print(mydept)
# Vs
print("mydept")
```

```
sales
mydept
```

In [9]:
```
print(MYDEPT)
```

```
production
```

In [11]:
```
mydept='sales'
print('My working dept is:mydept')
```

```
My working dept is:mydept
```

In [14]:
```
print('My working dept is:',mydept)
```

```
My working dept is: sales
```

In [15]:
```
'''Write a python program
initialize product ID,productName,productCost to individual variable
display product details,test product cost is above 1000 or not'''

pid = 101  # pid='A101'
pname = 'pA'
pcost = 1235.26   # pcost = '1235.26Rs'
print('productName is:',pname)
print(pname,'cost is:',pcost)
print(pname,'ID is:',pid)
print(pcost >1000)
```

```
productName is: pA
pA cost is: 1235.26
pA ID is: 101
True
```

In [16]:
```
n=56
print("n value is:",n)  # 1st style

print("n value is:%d"%(n)) # 2n style   %d -int  %f float %s string

print("n value is:{}".format(n)) # 3rd style

print(f"n value is:{n}") # 4th style
```

```
n value is: 56
n value is:56
n value is:56
n value is:56
```

In [18]:
```
'''Write a python program
initialize product ID,productName,productCost to individual variable
display product details,test product cost is above 1000 or not'''

pid = 102
pname = 'pB'
pcost = 590.42   # pcost = '1235.26Rs'
print(f'productName is:{pname}')
print(f'{pname} cost is:{pcost}')
print(f'{pname} ID is:{pid}')
print(pcost >1000)
```

```
productName is:pB
pB cost is:590.42
pB ID is:102
False
```

In [19]:
```
# Multiline string

print('''statement1
statement2
statement3
statement4''')
```

```
statement1
statement2
statement3
statement4
```

In [23]:
```
'''Write a python program
initialize product ID,productName,productCost to individual variable
display product details,test product cost is above 1000 or not'''

pid = 102
pname = 'pB'
pcost = 590.42   # pcost = '1235.26Rs'

print(f'''productName is:{pname}
{pname} cost is:{pcost}
{pname} ID is:{pid}''')

print(pcost > 1000)
```

```
productName is:pB
pB cost is:590.42
pB ID is:102
False
```

In [25]:
```python
# input() - interface to keyboard

#  variable = input('user defined string')

emp_name = input('Enter an emp name:')

print(f'Emp name is:{emp_name}')
```

```
Emp name is:anu
```

In [26]:
```python
emp_name = input('Enter an emp name:')
emp_dept = input(f'Enter {emp_name} working dept:')
print(f'Emp name is:{emp_name} working dept is:{emp_dept}')
```

```
Emp name is:anu working dept is:HR
```

In [28]:
```python
pname = input('Enter a product name:')
pid = input(f'Enter {pname} ID:')
pcost = input(f'Enter {pname} price:')

print(f'''productName is:{pname}
{pname} cost is:{pcost}
{pname} ID is:{pid}''')

#print(pcost > 1000)
```

```
productName is:HardDisk
HardDisk cost is:5000
HardDisk ID is:HD1
```

In [2]:
```python
pname = input('Enter a product name:')
pid = input(f'Enter {pname} ID:')
pcost = input(f'Enter {pname} price:')

print(f'''productName is:{pname}
{pname} cost is:{pcost}
{pname} ID is:{pid}''')

print(int(pcost) > 1000)
```

```
productName is:CDROM
CDROM cost is:1250
CDROM ID is:CD123
True
```

In [ ]:
```
Write a python program
read an emp name,emp ID,emp working dept,working place and emp basic pay
calculate 18% of basic pay and initialize calculated results to new variable
|
display - emp details line by line

Expected Result
-----------------
```

```
Emp name is: Arun
---------------------
Arun emp id: 1234
----------------------
Arun working department is:sales
--------------------------------
Arun working place is:Pune
--------------------------
Basic pay: 1000
----------------
18% of cost: 180
-----------------
Total Cost: 1180
-------------------------------------//use single print()
```

In [4]:

```
ename = input('Enter an Emp name:')
empid = input(f'Enter {ename} emp ID:')
empdept = input(f'Enter {ename} working dept:')
eplace = input(f'Enter {ename} working city:')
ecost = input(f'Enter {ename} basic pay:')

tax = float(ecost) * 0.18
gs = float(ecost)+tax

print(f'''Emp name is:{ename}
--------------------------------
{ename} Emp id:{empid}  Working department is:{empdept}
----------------------------------------------------------
{ename} Working city is:{eplace} Basic Pay is:{ecost}
-----------------------------------------------------------
Including tax:{gs}
-----------------------''')
```

```
Emp name is:Raj
--------------------------------
Raj Emp id:1234  Working department is:Sales
----------------------------------------------------------
Raj Working city is:Bangalore Basic Pay is:1200
-----------------------------------------------------------
Including tax:1416.0
-----------------------
```

In [ ]:

```
int float str bool(True False) ....
How to determine python type ?
 type(value)
```

In [5]:

```
print(type(10),type(10.0),type(''),type(True),type(False))
```

```
<class 'int'> <class 'float'> <class 'str'> <class 'bool'> <class 'bool'>
```

In [6]:

```
var = 120
print(type(var))
```

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

In [ ]:

```
list   -  Collection of different types of value - index based - mutable (we can ad
 []
```

```
tuple  -  Collection of different types of value - index based - immutable
 ()

dict   - -  Collection of different types of value - key:value - mutable( we can ad
 {}
```

In [7]:
```
print(type([]))
```

```
<class 'list'>
```

In [8]:
```
# ListName = [<list of items>]

emp = ['Raj',1234,True,4589.23]
     #   0   |  1 |  2 |  3  <== index
     #  -4     -3   -2   -1  <== index

# To get/fetch nth index value
# ListName[index] -> Value /IndexError

print(emp)
print(emp[0])
print(emp[1],emp[3])
print(f'Emp name is:{emp[0]}')
```

```
['Raj', 1234, True, 4589.23]
Raj
1234 4589.23
Emp name is:Raj
```

In [9]:
```
print(emp[3])
print(emp[-1])
```

```
4589.23
4589.23
```

In [10]:
```
L=['D1','D2',10,2.45,True]
print(L)

# len(inputList) ->output_int

print(len(L))
```

```
['D1', 'D2', 10, 2.45, True]
5
```

In [11]:
```
L=[]
print(len(L))
```

```
0
```

In [12]:
```
L=['D1','D2',10,2.45,True]
# How to modify an existing value from list
#  listName[oldIndex] = value
print(L)
print(L[1])
L[1] = 4590.23 # we can modify an exising value - list is mutable
print('') # empty line
print(L)
print(L[1])
```

```
['D1', 'D2', 10, 2.45, True]
D2

['D1', 4590.23, 10, 2.45, True]
4590.23
```

In [ ]:
```
# To add new data to an existing list
#
# Listname.append()  =>   Listname.append(Value) ->None
#   Vs
# Listname.insert(index,Value) ->None
```

In [13]:
```
dept = ['sales','DBA','crm']
print(f'No.of dept:{len(dept)}')
print(dept)
print('')

dept.append('QA') # adding new value - at the end of the list
dept.append('Devops') # adding new value - at the end of the list
print(f'No.of dept:{len(dept)}')
print(dept)
```

```
No.of dept:3
['sales', 'DBA', 'crm']

No.of dept:5
['sales', 'DBA', 'crm', 'QA', 'Devops']
```

In [14]:
```
print(dept)

dept.insert(1,'AI') # adding new value at the 1st index
print(dept)
```

```
['sales', 'DBA', 'crm', 'QA', 'Devops']
['sales', 'AI', 'DBA', 'crm', 'QA', 'Devops']
```

In [ ]:
```
# create an empty list name is called products
# use len() - display number of products - 0
# use append() - add few products into an existing list
# use len() - display number of products
|
# modify 1st index product value => prodAB
# display updated product list
```

In [16]:
```
products = []
print(f'No.of products:{len(products)}')

products.append('prodX')
products.append('prodY')
products.append('prodZ')
products.append('prodA')
print(f'No.of products:{len(products)}')
print(products)
```

```
No.of products:0
No.of products:4
['prodX', 'prodY', 'prodZ', 'prodA']
```

In [17]:
```
products[1] = 'prodAB' # modification
```

In [18]:
```
print(products)
```

```
['prodX', 'prodAB', 'prodZ', 'prodA']
```

In [ ]:
```
# To remove nth index item from given list

Listname.pop() # default - remove last index value  -> return removedValue
(or)
Listname.pop(index) - remove nth index value ->return removedValue
```

In [19]:
```
L=['D1','D2','D3','D4','D5',10,20,30,40,50]
print(len(L))
L.pop()
```

```
10
```

Out[19]:
```
50
```

In [20]:
```
L=['D1','D2','D3','D4','D5',10,20,30,40,50]
print(len(L))
removed_value = L.pop()
print(len(L))
```

```
10
9
```

In [21]:
```
print(removed_value)
```

```
50
```

In [22]:
```
print(L)
```

```
['D1', 'D2', 'D3', 'D4', 'D5', 10, 20, 30, 40]
```

In [23]:
```
rv = L.pop(2)
print(f"removed value:{rv}")
print(L)
```

```
removed value:D3
['D1', 'D2', 'D4', 'D5', 10, 20, 30, 40]
```

In [24]:
```
print(type([]),type(()))
```

```
<class 'list'> <class 'tuple'>
```

In [25]:
```
# emp = ['Raj',1234,True,4589.23] - list

emp = ('Raj',1234,True,4589.23) # tuple
print(type(emp),len(emp))
```

```
<class 'tuple'> 4
```

In [26]:
```
print(emp[0])
print(emp[1])
print(emp[-1])
```

```
Raj
1234
4589.23
```

In [28]:
```
emp[1] = 4567 # Error - tuple is immutable - we can't modify the nth item
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 emp[1] = 4567

TypeError: 'tuple' object does not support item assignment
```

In [29]:
```
# str also immutable - we can't add/delete/modify like tuple

s='abcdefg'
print(type(s),len(s))
print(s[0])
print(s[1])
print(s[-1])
```

```
<class 'str'> 7
a
b
g
```

In [30]:
```
s[1]='X' # Error - str is immutable - we can't modify
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[30], line 1
----> 1 s[1]='X'

TypeError: 'str' object does not support item assignment
```

########## END OF THE DAY1 SESSION ################