Print command

Here we use print function with setting desire values in

```
print ("My name is {}, My Number is {} There is {}gb ram in my
laptop".format("Niko",'10',16))
```

```
My name is Niko, My Number is 10 There is 16gb ram in my laptop
```

Here we change the position of X and y

```
print("First = {x} Second = {y}".format(y='Yadav',x='Nitin'))
```

```
First = Nitin Second = Yadav
```

Here we have repeat those x and y again

```
print("{x} is a boy, {x} is preparing for data science, did {y}
too".format(x='Niko', y='MBA'))
```

```
Niko is a boy, Niko is preparing for data science, did MBA too
```

List

How to Prepare list

```
mylist = ['a','b','c','z',1,2,'my','name','is','Anthony Gonsalves']
```

Calling any item or items from list , counting in Pyhton starts from 0

```
mylist[0],mylist[-1],mylist[9]  #Here -1 means last -2 means second
last and for multipale we need to put , in beteen them
```

```
('a', 'Anthony Gonsalves', 'Anthony Gonsalves')
```

Changing item in list

```
mylist[-1] = 'Niko'
mylist
```

```
['a', 'b', 'c', 'z', 1, 2, 'my', 'name', 'Niko']
```

Here we will add new item in list we will use append function it will added at last

```
mylist.append('Yadav')
mylist
```

```
['a', 'b', 'c', 'z', 1, 2, 'my', 'name', 'Niko', 'Yadav']
```

Remove last item from list

```
last = mylist.pop() # if we dont provide any  number it will pop last
last  # last item that pop out
```

```
'Yadav'
```

```
mylist
```

```
['a', 'b', 'c', 'z', 1, 2, 'my', 'name', 'Niko']
```

```
second = mylist.pop(1) # here we are pop out second item in python counting satrt from 0
second
```

```
'b'
```

```
mylist
```

```
['a', 'c', 'z', 1, 2, 'my', 'name', 'Niko']
```

Here we will find item in list

```
'z' in mylist
```

```
True
```

```
'k' in mylist
```

```
False
```

```
'a' in ['a','b','c']
```

```
True
```

List inside list

```
mylist1=['a','b',[100,200,['my','name','is',['Ram','Shaam','Niko']]]]
mylist1
```

```
['a', 'b', [100, 200, ['my', 'name', 'is', ['Ram', 'Shaam', 'Niko']]]]
```

calling items from list inside list

```
mylist1[2][2][3][2]
```

```
'Niko'
```

Dictionary

Dictionary doesnt carry order so we cant call them by there position but Dictionary can be call by key values

```
d= {'key1': 'value1', 'key2': 'value2','key3':'value3'}
d
```

```
{'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
```

calling Dictionary values

```
 d['key1'], d['key2']  # here to call slected values by keys
```

```
('value1', 'value2')
```

```
d.keys() # Here we all calling only keys
```

```
dict_keys(['key1', 'key2', 'key3'])
```

```
d.values() # Here we calling only values
```

```
dict_values(['value1', 'value2', 'value3'])
```

```
print(d) # here we are calling all the keys and values
```

```
{'key1': 'value1', 'key2': 'value2', 'key3': 'value3'}
```

Tuple

Tuples are type of list but they are immutable and they also have ()

```
t= (1,2,3)
t
```

```
(1, 2, 3)
```

```
t[0] = 'new'
```

```
---------------------------------------------------------------------------
-----
TypeError                                 Traceback (most recent call
last)
~\AppData\Local\Temp\ipykernel_10132\1055354548.py in <module>
----> 1 t[0] = 'new'

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

Sets

set is an unordered collection of unique items

```
s={1,1,1,1,1,2,2,2,2,3,3,3,3}
s
```

```
{1, 2, 3}
```

Conditional Statement that result in True and False

```
1> 5
```

```
False
```

```
1<=5
```

```
True
```

```
1<4 and 3==(2+1) and 3>2
```

```
True
```

```
if 1==3:
    print('Worked!')
else:
    print('Nope!')
```

```
Nope!
```

```
if 1==3:
    print('Worked!')
elif 1<8:
    print('Yeah its right!')
else:
    print('Nope!')
```

```
Yeah its right!
```

For loop

```
seq= [1,2,3,4,5]

for item in seq:         #Here we can use anything in place of item
but that should be same in for and print
    print(item)
```

```
1
2
3
4
5
```

few more expriments with for loop and changing name

```
for jelly in seq:
    print (jelly)
```

```
1
2
3
4
5
```

```
for item in seq:
    print ('jelly')
```

```
jelly
jelly
jelly
jelly
jelly
```

```
for jelly in seq:
    print (jelly,(jelly+jelly)*10,'Jelly')
```

```
1 20 Jelly
2 40 Jelly
3 60 Jelly
4 80 Jelly
5 100 Jelly
```

Looping with While

```
i=1
while i<5:
    print ('i is {}'.format(i))
    i=i+1
```

```
i is 1
i is 2
i is 3
i is 4
```

Here we will create list by range

```
a=range(5)   # to know syntax press shift + tab
a
```

```
range(0, 5)
```

```
list(range(5)) # or list(a)  # range doesnt display last number
```

```
[0, 1, 2, 3, 4]
```

```
list(range(5,32,5)) # Here start at 5 end at 32 with difference of 5
```

```
[5, 10, 15, 20, 25, 30]
```

```
for x in range(5):
    print('hi')  # There are 4 hi not 5
```

```
hi
hi
hi
hi
hi
```

List comprehension

Here we are going to create list with that square of items in list

```
x=[1,2,3,4,5,6,7,8,9,10]
x
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
out=[]          # Here we create null list
for num in x:
    out.append(num**2)     # now appending values by squaring them in
list out

print(out)
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Here we are going to perform same by using list comprehension

```
[num**2 for num in x]
```

```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

Function

```
def myfunc():   # Here we have define function
    print('Hello')

myfunc()
```

```
Hello
```

Now we will give to argument in function

```
def myfunc(name):
    print('Hello ' +name)    # add + Sign here is compulsory

myfunc('Niko')
```

```
Hello Niko
```

Here it going to set default if person not provide any name

```
def myfunc(name = 'NO NAME'):
    print('Hello ' + name)

myfunc()
```

```
Hello NO NAME
```

```
def myfunc(name = 'NO NAME'): #A docstring is a string literal that
occurs as the first statement in
                              #a module, function, class, or method
definition
    '''
    This will show as docstring if you press shift + tab you will se
this line will automatically added in docstring
    '''
    print('Hello ' + name)

myfunc('Niko')
```

```
Hello Niko
```

Return

```
def square(x):
    return(x**2)

resultr = square(8)
resultr, type(resultr) #Here we can see that result type is integer
```

```
(64, int)
```

Difference between Return and Print

```
def square1(x):
    print(x**2)

result = square1(8)  # here we can see that print statement is
NoneTpye because it doesnt hold it just show value
result, type(result), type(resultr)         # but return holds values
```

```
64

(None, NoneType, int)
```

Lambda Function

```
def five_times(var):
    return(var*5)

five_times(3)
```

```
15
```

```
x=lambda  var: var*5 # Here we are use lambda function lambda then
name : and then formula
x(3)
```

```
15
```

Strings

```
st = "Hello my name is NIKO"

st.lower()
```

```
'hello my name is niko'
```

```
st.upper()
```

```
'HELLO MY NAME IS NIKO'
```

```
st.split() # Now here that split function convert string into list
```

```
['Hello', 'my', 'name', 'is', 'NIKO']
```

```
s1= "Hey#i#am#learning#Python"
s1
```

```
'Hey#i#am#learning#Python'
```

```
s1.split('#')
```

```
['Hey', 'i', 'am', 'learning', 'Python']
```

Now we will split on the bases of some specail character

```
s2= "Hello #sport" # here we have to extract hashtag word
```

```
s2.split('#')[-1] # -1 is to extract last hashtag word
```

```
'sport'
```

```
s3= "Hello #sport #Party"
```

```
s3.split('#')[-2]
```

```
'sport '
```