

In [36]:

```
1 Agenda = '''
2 1. What is Python?
3 2. Data Types
4 2.5. Operators / Arithmetic Operation
5 3. Data Structure
6     a) Mutable : list, dictionary, set
7     b) Immutable : tuple, string, numbers
8 4. Control Structure
9 5. Function
10    1) User Define Function
11 6. Group Discussion and Summary
12 '''
```

In [4]:

```
1 print(Agenda)
```

```
1. What is Python?
2. Data Types
3. Data Structure
   a) Mutable : list, dictionary, set
   b) Immutable : tuple, string, numbers
4. Control Structure
5. Function
   1) Built in Function
   2) User Define Function
6. Group Discussion and Summary
```

In [7]:

```
1 data = 30.5
```

In [8]:

```
1 type(data)
```

Out[8]:

float

In [9]:

```
1 cdata = 5 + 10j
```

In [10]:

```
1 type(cdata)
```

Out[10]:

complex

In [11]:

```
1 cdata1 = complex(3,9)
```

In [12]:

```
1 cdata1
```

Out[12]:

(3+9j)

In [13]:

```
1 cdata.real
```

Out[13]:

5.0

In [14]:

```
1 cdata.imag
```

Out[14]:

10.0

In [15]:

```
1 cdata.conjugate()
```

Out[15]:

(5-10j)

In [16]:

```
1 bdata = True
```

In [17]:

```
1 type(bdata)
```

Out[17]:

bool

In [18]:

```
1 bdata = False
2 type(bdata)
```

Out[18]:

bool

In [19]:

```
1 True
```

Out[19]:

True

In [20]:

```
1 True + True + True - False
```

Out[20]:

3

In [21]:

```
1 sdata = 'Welcome'
```

In [22]:

```
1 type(sdata)
```

Out[22]:

str

In [27]:

```
1 #help(str)
```

In [28]:

```
1 sdata.upper()
```

Out[28]:

'WELCOME'

In [32]:

```
1 dir(sdata)
```

Out[32]:

```
['_add__',  
'_class__',  
'_contains__',  
'_delattr__',  
'_dir__',  
'_doc__',  
'_eq__',  
'_format__',  
'_ge__',  
'_getattr__',  
'_getitem__',  
'_getnewargs__',  
'_gt__',  
'_hash__',  
'_init__',  
'_init_subclass__',  
'_iter__',
```

'\_\_le\_\_',  
'\_\_len\_\_',  
'\_\_lt\_\_',  
'\_\_mod\_\_',  
'\_\_mul\_\_',  
'\_\_ne\_\_',  
'\_\_new\_\_',  
'\_\_reduce\_\_',  
'\_\_reduce\_ex\_\_',  
'\_\_repr\_\_',  
'\_\_rmod\_\_',  
'\_\_rmul\_\_',  
'\_\_setattr\_\_',  
'\_\_sizeof\_\_',  
'\_\_str\_\_',  
'\_\_subclasshook\_\_',  
'capitalize',  
'casefold',  
'center',  
'count',  
'encode',  
'endswith',  
'expandtabs',  
'find',  
'format',  
'format\_map',  
'index',  
'isalnum',  
'isalpha',  
'isascii',  
'isdecimal',  
'isdigit',  
'isidentifier',  
'islower',  
'isnumeric',  
'isprintable',  
'isspace',  
'istitle',  
'isupper',  
'join',  
'ljust',  
'lower',  
'lstrip',  
'maketrans',  
'partition',  
'replace',  
'rfind',  
'rindex',  
'rjust',  
'rpartition',  
'rsplit',  
'rstrip',  
'split',  
'splitlines',  
'startswith',  
'strip',  
'swapcase',

```
'title',  
'translate',  
'upper',  
'zfill']
```

In [33]:

```
1 sdata.lower()
```

Out[33]:

```
'welcome'
```

In [34]:

```
1 sdata
```

Out[34]:

```
'Welcome'
```

In [35]:

```
1 sdata1 = sdata.upper()  
2 print(sdata1)
```

```
WELCOME
```

In [37]:

```
1 50 % 3
```

Out[37]:

```
2
```

In [38]:

```
1 43 + 43
```

Out[38]:

```
86
```

In [40]:

```
1 print('Welcome to Python \n' * 10)
```

```
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
Welcome to Python
```

In [41]:

```
1 List1 = []
2 type(List1)
```

Out[41]:

```
list
```

In [42]:

```
1 List2 = list()
2 type(List2)
```

Out[42]:

```
list
```

In [44]:

```
1 print(''' List is mutable ( Modify) sequence ( index ) of
2 data ( int, float, str, complex, list, tuple etc) ''')
```

```
List is mutable ( Modify) sequence ( index ) of
data ( int, float, str, complex, list, tuple etc)
```

In [45]:

```
1 dir(List1)
```

Out[45]:

```
['_add__',  
 '_class__',  
 '_contains__',  
 '_delattr__',  
 '_delitem__',  
 '_dir__',  
 '_doc__',  
 '_eq__',  
 '_format__',  
 '_ge__',  
 '_getattr__',  
 '_getitem__',  
 '_gt__',  
 '_hash__',  
 '_iadd__',  
 '_imul__',  
 '_init__',  
 '_init_subclass__',  
 '_iter__',  
 '_le__',  
 '_len__',  
 '_lt__',  
 '_mul__',  
 '_ne__',  
 '_new__',  
 '_reduce__',  
 '_reduce_ex__',  
 '_repr__',  
 '_reversed__',  
 '_rmul__',  
 '_setattr__',  
 '_setitem__',  
 '_sizeof__',  
 '_str__',  
 '_subclasshook__',  
 'append',  
 'clear',  
 'copy',  
 'count',  
 'extend',  
 'index',  
 'insert',  
 'pop',  
 'remove',  
 'reverse',  
 'sort']
```



In [48]:

```
1 # List methods for stack operation
2 List1 = []
3 List1.append('Cbook')
4 List1.append('Cppbook')
5 List1.append('Javabook')
6 List1.append('Pythonbook')
7 List1.append('Perlbook')
8 List1.append('Jythonbook')
9 List1.append('Sparkbook')
```

In [49]:

```
1 List1
```

Out[49]:

```
['Cbook',
 'Cppbook',
 'Javabook',
 'Pythonbook',
 'Perlbook',
 'Jythonbook',
 'Sparkbook']
```

In [50]:

```
1 List1.pop()
```

Out[50]:

```
'Sparkbook'
```

In [51]:

```
1 List1
```

Out[51]:

```
['Cbook', 'Cppbook', 'Javabook', 'Pythonbook', 'Perlbook', 'Jythonbo
ok']
```

In [52]:

```
1 # Queue Operation on List. First in First Out
```

In [53]:

```
1 List1.pop(0)
```

Out[53]:

'Cbook'

In [54]:

```
1 List1
```

Out[54]:

```
['Cppbook', 'Javabook', 'Pythonbook', 'Perlbook', 'Jythonbook']
```

In [55]:

```
1 List1[0]
```

Out[55]:

'Cppbook'

In [56]:

```
1 List1[1]
```

Out[56]:

'Javabook'

In [58]:

```
1 List1[0]
```

Out[58]:

'Cppbook'

In [59]:

```
1 List1[0] = 'Scalabook'
```

In [60]:

```
1 List1
```

Out[60]:

```
['Scalabook', 'Javabook', 'Pythonbook', 'Perlbook', 'Jythonbook']
```

In [62]:

```
1 List1[0:2] = ['CPP', 'Cbook']
```

In [63]:

```
1 List1
```

Out[63]:

```
['CPP', 'Cbook', 'Pythonbook', 'Perlbook', 'Jythonbook']
```

In [64]:

```
1 List1[0:2] = ['Perl']
```

In [65]:

```
1 List1
```

Out[65]:

```
['Perl', 'Pythonbook', 'Perlbook', 'Jythonbook']
```

In [67]:

```
1 List1[0:3]
```

Out[67]:

```
['Perl', 'Pythonbook', 'Perlbook']
```

In [69]:

```
1 List2 = ['Mongobook', 'Sqlbook', 'Mysqlbook']
```

In [70]:

```
1 List1
```

Out[70]:

```
['Perl', 'Pythonbook', 'Perlbook', 'Jythonbook']
```

In [71]:

```
1 List1.extend(List2)
2 print(List1)
```

```
['Perl', 'Pythonbook', 'Perlbook', 'Jythonbook', 'Mongobook', 'Sqlbook', 'Mysqlbook']
```

In [72]:

```
1 List1.insert(2, 'Oraclebook')
2 print(List1)
```

```
['Perl', 'Pythonbook', 'Oraclebook', 'Perlbook', 'Jythonbook', 'Mongobook', 'Sqlbook', 'Mysqlbook']
```

In [73]:

```
1 List1.sort()
```

In [74]:

```
1 List1
```

Out[74]:

```
['Jythonbook',  
'Mongobook',  
'Mysqlbook',  
'Oraclebook',  
'Perl',  
'Perlbook',  
'Pythonbook',  
'Sqlbook']
```

In [75]:

```
1 List1.reverse()
```

In [76]:

```
1 List1
```

Out[76]:

```
['Sqlbook',  
'Pythonbook',  
'Perlbook',  
'Perl',  
'Oraclebook',  
'Mysqlbook',  
'Mongobook',  
'Jythonbook']
```

In [77]:

```
1 print('' Tuple is immutable ( Modify) sequence  
2 ( index ) of data ( int, float, str, complex,  
3 list, tuple etc) '' )
```

```
Tuple is immutable ( Modify) sequence  
( index ) of data ( int, float, str, complex,  
list, tuple etc)
```

In [78]:

```
1 Tuple1 = ()
2 print(type(Tuple1))
3 Tuple2 = tuple()
4 print(type(Tuple2))
5 Tuple3 = 2,3,5
6 print(type(Tuple3))
```

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

In [81]:

```
1 Tuple4 = 5,
2 print(type(Tuple4))
```

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

In [84]:

```
1 Tuple3[0] = 89 # Tuples are immutable
```

```
-----
-----
TypeError                                Traceback (most recent call
l last)
<ipython-input-84-a066cba9869f> in <module>()
----> 1 Tuple3[0] = 89
```

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

In [85]:

```
1 dir(Tuple3)
```

Out[85]:

```
['_add__',  
'__class__',  
'__contains__',  
'__delattr__',  
'__dir__',  
'__doc__',  
'__eq__',  
'__format__',  
'__ge__',  
'__getattr__',  
'__getitem__',  
'__getnewargs__',  
'__gt__',  
'__hash__',  
'__init__',  
'__init_subclass__',  
'__iter__',  
'__le__',  
'__len__',  
'__lt__',  
'__mul__',  
'__ne__',  
'__new__',  
'__reduce__',  
'__reduce_ex__',  
'__repr__',  
'__rmul__',  
'__setattr__',  
'__sizeof__',  
'__str__',  
'__subclasshook__',  
'count',  
'index']
```

In [86]:

```
1 import time
```

In [87]:

```
1 dir(time)
```

Out[87]:

```
['_STRUCT_TM_ITEMS',  
'__doc__',  
'__loader__',  
'__name__',  
'__package__',  
'__spec__',  
'altzone',  
'asctime',  
'clock',  
'ctime',  
'daylight',  
'get_clock_info',  
'gmtime',  
'localtime',  
'mktime',  
'monotonic',  
'monotonic_ns',  
'perf_counter',  
'perf_counter_ns',  
'process_time',  
'process_time_ns',  
'sleep',  
'strftime',  
'strptime',  
'struct_time',  
'time',  
'time_ns',  
'timezone',  
'tzname',  
'tzset']
```

In [91]:

```
1 TS = time.localtime()  
2 print(TS)  
3 type(TS)
```

```
time.struct_time(tm_year=2019, tm_mon=5, tm_mday=27, tm_hour=12, tm_  
min=55, tm_sec=2, tm_wday=0, tm_yday=147, tm_isdst=0)
```

Out[91]:

```
time.struct_time
```



In [92]:

```
1 Year, Mon, Day, Hour, Min, Sec, Wday, Yday, isdst = TS
```

In [93]:

```
1 Year, Mon, Day, Hour
```

Out[93]:

```
(2019, 5, 27, 12)
```

In [94]:

```
1 TS1 = Year, Mon, Day, Hour, Min, Sec, Wday, Yday, isdst
```

In [95]:

```
1 TS1
```

Out[95]:

```
(2019, 5, 27, 12, 55, 2, 0, 147, 0)
```

In [96]:

```
1 type(TS1)
```

Out[96]:

```
tuple
```

In [97]:

```
1 Var1, Var2, Var3, Var4 = 45, 63, 60, 23 # parallel assignment
```

In [98]:

```
1 Var1
```

Out[98]:

45

In [99]:

```
1 Temp = Var1
2 Var1 = Var2
3 Var2 = Temp
4 print(Var1, Var2)
```

63 45

In [100]:

```
1 Var1, Var2, Var3, Var4 = Var4, Var3, Var2, Var1
```

In [103]:

```
1 Tvar1 = Var1, Var2, Var3, Var4
2 type(Tvar1)
```

Out[103]:

tuple

In [104]:

```
1 Tvar2 = Tvar1 + ( 100, 200 )
```

In [105]:

```
1 Tvar2
```

Out[105]:

(23, 60, 45, 63, 100, 200)

In [106]:

```
1 Tvar1
```

Out[106]:

```
(23, 60, 45, 63)
```

In [107]:

```
1 Var1 = 90
```

In [108]:

```
1 Tvar1
```

Out[108]:

```
(23, 60, 45, 63)
```

In [109]:

```
1 '''Indentation'''
```

Out[109]:

```
'Indentation'
```

In [110]:

```
1 var4 = 90
2 print(var4) # Indentation Error
```

```
File "<ipython-input-110-6f09ccf768f4>", line 2
```

```
print(var4)
```

```
^
```

```
IndentationError: unexpected indent
```

In [115]:

```
1 var4 = 90; print(var4)
```

90

In [128]:

```
1  ###writefile great_two.py
2
3  Num1 = int(input("Enter Value for Num1:"))
4  Num2 = int(input("Enter Value for Num2:"))
5
6  if Num1 > Num2 :
7      print("Num1 is greatest")
8      print("If block continued")
9  else: # Else Block started if ended
10     print("Num2 is greatest")
11     print("Else block continued")
12 print("else block ended")
```

```
Enter Value for Num1:45
Enter Value for Num2:20
Num1 is greatest
If block continued
else block ended
```

In [118]:

```
1 run great_two.py
```

```
Enter Value for Num1:34
Enter Value for Num2:54
Num2 is greatest
```

In [129]:

```
1 %%writefile great_three.py
2
3 Num1 = int(input("Enter Value for Num1:"))
4 Num2 = int(input("Enter Value for Num2:"))
5 Num3 = int(input("Enter Value for Num3:"))
6
7 if (Num1 > Num2) and (Num1 > Num3):
8     print("Num1 is greatest")
9 elif Num2 > Num3:
10    print("Num2 is greatest")
11 else:
12    print("Num3 is greatest")
```

Writing great\_three.py

In [132]:

```
1 while True:
2     Num1 = int(input("Enter Value for Num1:"))
3     Num2 = int(input("Enter Value for Num2:"))
4     Num3 = int(input("Enter Value for Num3:"))
5
6     if (Num1 > Num2) and (Num1 > Num3):
7         print("Num1 is greatest")
8     elif Num2 > Num3:
9         print("Num2 is greatest")
10    else:
11        print("Num3 is greatest")
12        break
13
```

Enter Value for Num1:3

Enter Value for Num2:3

Enter Value for Num3:3

Num3 is greatest

In [133]:

```
1 List1
```

Out[133]:

```
['Sqlbook',  
'Pythonbook',  
'Perlbook',  
'Perl',  
'Oraclebook',  
'Mysqlbook',  
'Mongobook',  
'Jythonbook']
```

In [134]:

```
1 for book in List1:  
2     print(book)
```

```
Sqlbook  
Pythonbook  
Perlbook  
Perl  
Oraclebook  
Mysqlbook  
Mongobook  
Jythonbook
```

In [136]:

```
1 for book in List1:  
2     print(List1.index(book),book)
```

```
0 Sqlbook  
1 Pythonbook  
2 Perlbook  
3 Perl  
4 Oraclebook  
5 Mysqlbook  
6 Mongobook  
7 Jythonbook
```

In [138]:

```
1 list(range(20))
```

Out[138]:

```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [139]:

```
1 list(range(1,20))
```

Out[139]:

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

In [140]:

```
1 list(range(1,20,2))
```

Out[140]:

```
[1, 3, 5, 7, 9, 11, 13, 15, 17, 19]
```

In [142]:

```
1 list(range(2,20,2))
```

Out[142]:

```
[2, 4, 6, 8, 10, 12, 14, 16, 18]
```

In [146]:

```
1 for i in range(len(List1)):  
2     print(i,List1[i])
```

```
0 Sqlbook  
1 Pythonbook  
2 Perlbook  
3 Perl  
4 Oraclebook  
5 Mysqlbook  
6 Mongobook  
7 Jythonbook
```

In [153]:

```
1 String1 = 'python scripting language'
```

In [150]:

```
1 String1[0:6]
```

Out[150]:

```
'python'
```

In [154]:

```
1 String1[7:]
```

Out[154]:

```
'scripting language'
```

In [155]:

```
1 String1[-1]
```

Out[155]:

```
'e'
```

In [163]:

```
1 String1[None:None:-1]
```

Out[163]:

```
'egaugnal gnitpircs nohtyp'
```

In [164]:

```
1 String1[::-1]
```

Out[164]:

```
'egaugnal gnitpircs nohtyp'
```



In [167]:

```
1 String1[::2]
```

Out[167]:

```
'pto citn agae'
```

In [168]:

```
1 String1.upper()
```

Out[168]:

```
'PYTHON SCRIPTING LANGUAGE'
```

In [169]:

```
1 String1.title()
```

Out[169]:

```
'Python Scripting Language'
```

In [170]:

```
1 String1[0]
```

Out[170]:

```
'p'
```

In [171]:

```
1 String1[0]='W'
```

```
-----  
-----  
TypeError                                 Traceback (most recent call  
1 last)  
<ipython-input-171-9c1db2fe5798> in <module>()  
----> 1 String1[0]='W'
```

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

In [173]:

```
1 # User Define Function
2
3 def hello():
4     '''This is a hello function'''
5     print("Hello")
6
7
```

In [174]:

```
1 hello()
```

Hello

In [178]:

```
1 h = hello # alias of hello function
2
3 h()
```

Hello

In [176]:

```
1 h.__doc__
```

Out[176]:

```
'This is a hello function'
```

In [179]:

```
1 help(hello)
```

Help on function hello in module `__main__`:

```
hello()
    This is a hello function
```

In [183]:

```
1 def greatest_two_number(Num1, Num2):
2     '''Greatest of two numbers'''
3     if Num1 > Num2 :
4         print("%d is greatest"%Num1)
5     else:
6         print("%d is greatest"%Num2)
```

In [184]:

```
1 greatest_two_number(32,53)
2 greatest_two_number(64,53)
```

53 is greatest  
64 is greatest

In [188]:

```
1 def greatest_three_number(Num1, Num2, Num3):
2     '''Greatest of three numbers'''
3     if (Num1 > Num2) and (Num1 > Num3):
4         print("%d is greatest"%Num1)
5     elif Num2 > Num3:
6         print("%d is greatest"%Num2)
7     else:
8         print("%d is greatest"%Num3)
9
10 greatest_three_number(45,63,92)
```

92 is greatest

In [189]:

```
1 greatest_three_number(423,623,92)
```

623 is greatest

In [191]:

```
1 import math
2 math.factorial(8)
```

Out[191]:

40320

In [192]:

```
1 5*4*3*2*1
```

Out[192]:

120

In [193]:

```
1 def factorial(Num): # Recursive Function
2     '''Factorial of Number'''
3     if Num <= 0:
4         return 1
5     else:
6         return Num*factorial(Num-1)
```

In [195]:

```
1 factorial(-1)
```

Out[195]:

1

In [196]:

```
1 factorial(6)
```

Out[196]:

720

In [197]:

```
1 '''Fibonacci Sequence of N Number'''
2
3 Seq = 0,1,1,2,3,5,8,13,21
```

In [202]:

```
1 def fibonacci(Num):
2     '''Fibonacci Sequence of Num'''
3
4     first, next_num = 0, 1
5     print(first)
6
7     while next_num < Num:
8         print(next_num)
9         first, next_num = next_num, first + next_num
10
11
```

In [201]:

```
1 fibonacci(50) # Calling Function
```

```
0
1
1
2
3
5
8
13
21
34
```

In [206]:

```
1 def localvar(n):
2     '''This is a demo of local function'''
3     print("Value of n is ",n)
4     n = 5
5     print("Value of n after intialisation ",n)
6
7
```

In [207]:

```
1 n = 200
2 localvar(50)
3 print("Value of n after local function", n)
```

```
Value of n is 50
Value of n after intialisation 5
Value of n after local function 200
```

In [211]:

```
1 def globalvar():
2     '''This is a demo of local function'''
3     global n # declare global variable
4     print("Value of n is ",n)
5     n = 5
6     print("Value of n after intialisation ",n)
7
8
9 n = 200
10 globalvar()
11 print("Value of n after local function", n)
12 n = 400
13 print("Value of n after local function", n)
```

```
Value of n is 200
Value of n after intialisation 5
Value of n after local function 5
Value of n after local function 400
```

In [212]:

```
1 def default_value(var1, var2 = 10, var3 = 'hello'):
2     '''Default Value argument function'''
3     print("var1 = %d var2 = %d var3 = %s "\
4           %(var1,var2,var3))
5
6
```

In [213]:

```
1 default_value(50)
```

```
var1 = 50 var2 = 10 var3 = hello
```

In [214]:

```
1 default_value(50,90)
```

```
var1 = 50 var2 = 90 var3 = hello
```

In [215]:

```
1 default_value(50,90,'Hi')
```

```
var1 = 50 var2 = 90 var3 = Hi
```

In [217]:

```
1 default_value(var3='New Position',var1=20)
```

```
var1 = 20 var2 = 10 var3 = New Position
```

In [218]:

```
1 def vararg(*arg):  
2     '''Variable Argument Function'''  
3     print(arg)
```

In [221]:

```
1 vararg(5,10,'hello','hi')
```

```
(5, 10, 'hello', 'hi')
```

In [222]:

```
1 5*2*1*8
```

Out[222]:

80

In [223]:

```
1 def multiplier(*anyarg):
2     '''Multiplier function for any number
3     of argument'''
4
5     result = 1
6
7     for var in anyarg:
8         result = result * var
9     return result
10
11
```

In [227]:

```
1 multiplier(4,5,'hello ')
```

Out[227]:

```
'hello hello hello hello hello hello hello hello hello hello h
ello hello hello hello hello hello hello hello hello '
```

In [ ]:

```
1
```