In [1]:
```
print("Hello, I am Learning Python")
```
```
Hello, I am Learning Python
```

In [2]:
```
#Arithmetic Operators

a = 10
b = 3

print(f"The Addition of {a} and {b} is", a+b)
print(f"The Subtraction of {a} and {b} is", a-b)
print(f"The Multiplication of {a} and {b} is",a*b)
print(f"The Divison of {a} and {b} is", a/b)
print(f"The Modulus of {a} and {b} is", a%b)
print(f"The Exponetation of {a} and {b} is", a**b)
```
```
The Addition of 10 and 3 is 13
The Subtraction of 10 and 3 is 7
The Multiplication of 10 and 3 is 30
The Divison of 10 and 3 is 3.3333333333333335
The Modulus of 10 and 3 is 1
The Exponetation of 10 and 3 is 1000
```

In [3]:
```
#Assigment Operator
x=9
x+=5 #x=x+5
print(x)
x-=3 #x=x-3
print(x)
x*=3 #x=x*3
print(x)
x/=3 #x=x/3
print(x)
x%=3 #x=x%3
print(x)
x**=3  #x=x**5
print(x)
x//=3  #x=x//3
print(x)
```
```
14
11
33
11.0
2.0
8.0
2.0
```

In [4]:
```
#Comparison Operator
x=33
y=41
print(print('x==y is',  x==y))
print('x<y is',  x<y)
print('x>y is',  x>y)
print('x>=y is',  x>=y)
print('x<=y is', x<=y)
```
```
x==y is False
None
x<y is True
x>y is False
x>=y is False
x<=y is True
```

In [5]:
```
x=7
print(x>3 and x<10) #returns true if both statements are true
```
```
True
```

In [6]:
```
#OR (Logical Operator)
x=15
print(x>5 or x<4)  #returns true if one of statements is true
```
```
True
```

In [7]:
```
a=33
b= 3.5
c= 3+3j
print(type(a))
print(type(b))
print(type(c))
```
```
<class 'int'>
<class 'float'>
<class 'complex'>
```

In [8]:
```
#String
print("Hello World")
```
```
Hello World
```

In [10]:
```
a = """Lorem ipsum dolor sit amet
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua"""
print(a)
```
```
Lorem ipsum dolor sit amet
consectetur adipiscing elit,
sed do eiusmod tempor incididunt
ut labore et dolore magna aliqua
```

In [22]:
```
a = "Hello world"
len(a)
```
Out[22]:
```
11
```

In [23]:
```
print(a[1])
```
```
e
```

In [24]:
```
print(a[3:8])
print(a[:5])
print(a[2:])
a[1:]
```
```
lo wo
Hello
llo world
```
Out[24]:
```
'ello world'
```

In [25]:
```
print(a[:5])
```
```
Hello
```

In [26]:
```
a[:]
```
Out[26]:
```
'Hello world'
```

In [27]:
```
a[-1]
```
Out[27]:
```
'd'
```

In [28]:
```
a[:-1]
```
Out[28]:
```
'Hello worl'
```

In [29]:
```
a[:-4]
```
Out[29]:
```
'Hello w'
```

In [30]:
```
a[::3]
```
Out[30]:
```
'Hlwl'
```

In [32]:
```
a[::-1]
```
Out[32]:
```
'dlrow olleH'
```

In [36]:
```
a= a+ " I am writing a code in jupyter notebook"
```

In [38]:
```
a
```
Out[38]:
```
'Hello world I am writing a code in jupyter notebook I am writing a code in jupyter notebook'
```

In [39]:
```
a
```
Out[39]:
```
'Hello world I am writing a code in jupyter notebook I am writing a code in jupyter notebook'
```

In [41]:
```
word = "hello world"
word
```
Out[41]:
```
'hello world'
```

In [42]:
```
word.count("l") #counts th total number of occurence of a character
```
Out[42]:
```
3
```

In [43]:
```
word.capitalize() #capitalize the first character of a string
```
Out[43]:
```
'Hello world'
```

In [44]:
```
word.find("world")
```
Out[44]:
```
6
```

In [45]:
```
word.replace("world", "Everyone")
```
Out[45]:
```
'hello Everyone'
```

In [53]:
```
lang="I\'m Learning"
```

In [54]:
```
lang
```
Out[54]:
```
"I'm Learning"
```

In [55]:
```
#LIST
newList=["Vegetables", "Fruits", 27, False]
print(newList)
```
```
['Vegetables', 'Fruits', 27, False]
```

In [59]:
```
newList[0]
```
Out[59]:
```
'Vegetables'
```

In [63]:
```
list_1=["apple","brinjal","cherry"]
list_1[1]="Banana"  #changing the value of list
print(list_1)
```
```
['apple', 'Banana', 'cherry']
```

In [64]:
```
list_1[1:2] = ["blackcurrant", "watermelon"]
print(list_1)
```
```
['apple', 'blackcurrant', 'watermelon', 'cherry']
```

In [65]:
```
list_1.insert(2,"strawberry")
list_1
```
Out[65]:
```
['apple', 'blackcurrant', 'strawberry', 'watermelon', 'cherry']
```

In [66]:
```
l1=[2,4,6,0,7]
```

In [68]:
```
l1
```
Out[68]:
```
[2, 4, 6, 0, 7]
```

In [69]:
```
l1.sort()
```

In [70]:
```
l1
```
Out[70]:
```
[0, 2, 4, 6, 7]
```

In [73]:
```
l1.reverse()
print(l1)
```
```
[7, 6, 4, 2, 0]
```

In [74]:
```
l1.pop(4)
print(l1)
```
```
[7, 6, 4, 2]
```

In [75]:
```
l1.remove(2)
l1
```
Out[75]:
```
[7, 6, 4]
```

In [76]:
```
#TUPLES
t=(1,2, "hello",3,"bye")
t
```
Out[76]:
```
(1, 2, 'hello', 3, 'bye')
```

In [77]:
```
print(t[-1])
```
```
bye
```

In [78]:
```
len(t)
```
Out[78]:
```
5
```

In [79]:
```
t[0]="change"
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In [79], line 1
----> 1 t[0]="change"

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

In [80]:
```
#DICTIONARY
my_dict={"key1":"Value1", "key2":"Value2"}
```

In [81]:
```
my_dict
```
Out[81]:
```
{'key1': 'Value1', 'key2': 'Value2'}
```

In [85]:
```
my_dict.values()
```
Out[85]:
```
dict_values(['Value1', 'Value2'])
```

In [86]:
```
my_dict.keys()
```
Out[86]:
```
dict_keys(['key1', 'key2'])
```

In [89]:
```
my_dict={"key1":1,"key2":[12,99,"Newvalue"],"key3":['item1','item2','item3']}
my_dict
```
Out[89]:
```
{'key1': 1, 'key2': [12, 99, 'Newvalue'], 'key3': ['item1', 'item2', 'item3']}
```

In [90]:
```
my_dict["key3"]
```
Out[90]:
```
['item1', 'item2', 'item3']
```

In [91]:
```
my_dict["key2"][0]
```
Out[91]:
```
12
```

In [93]:
```
my_dict["key2"][2].upper()
```
Out[93]:
```
'NEWVALUE'
```

In [97]:
```
new_d={}
new_d["val"]=1
new_d["marks"]=[22,33,44]
```

In [98]:
```
new_d
```
Out[98]:
```
{'val': 1, 'marks': [22, 33, 44]}
```

In [99]:
```
new_d.items()
```
Out[99]:
```
dict_items([('val', 1), ('marks', [22, 33, 44])])
```

In [100…
```
#SETS
thisset=set()
```

In [101…
```
thisset.add(1)
```

In [103…
```
thisset
```
Out[103]:
```
{1}
```

In [105…
```
thisset.add((4,5,6))   #adding tuple in set
thisset
```
Out[105]:
```
{(4, 5, 6), 1}
```

In [106…
```
thisset.add(["apple","banana"])   #we can't add list in set
thisset
```
```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In [106], line 1
----> 1 thisset.add(["apple","banana"])
      2 thisset

TypeError: unhashable type: 'list'
```

In [108…
```
thisset.add(1) #sets cannot contain duplicate values
thisset
```
Out[108]:
```
{(4, 5, 6), 1}
```

In [109…
```
len(thisset)
```
Out[109]:
```
2
```

In [110…
```
thisset.clear()
```

In [111…
```
thisset
```
Out[111]:
```
set()
```

In [113…
```
newset=("banana","apple",33,77,"name","age")
newset
```
Out[113]:
```
('banana', 'apple', 33, 77, 'name', 'age')
```

In [117…
```
#CONDITIONAL EXPRESSIONS
a=22
if(a>9):
    print("a is greater")
else:
    print("a is lesser")
```
```
a is greater
```

In [118…
```
str = "testing for loops"
count = 0

for x in str:
  if(x == 't'):
    count += 1

print(count)
```
```
2
```

In [119…
```
a = 35
b = 35
if b > a:
  print("b is greater than a")
elif a == b:
  print("a and b are equal")
```
```
a and b are equal
```

In [120…
```
i = 1
while i < 6:
  print(i)
  i += 1
```
```
1
2
3
4
5
```

In [121…
```
i = 0
while i < 6:
  print(i)
  if (i == 3):
    break
  i += 1
```
```
0
1
2
3
```

In [123…
```
#print content of the list using while loop
fruits=["Banana","Mango","Grapes","Kiwi"]
i=0
while i<len(fruits):
    print(fruits[i])
    i=i+1
```
```
Banana
Mango
Grapes
Kiwi
```

In [ ]:

In [ ]:

In [ ]: