Advance-Indexing

In [1]:

```
a=20
b='abhi' # a string cannot be added to a number, but it can be multiplied by a numb
c=a+b
print(c)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[1], line 3
      1 a=20
      2 b='abhi'
----> 3 c=a+b
      4 print(c)

TypeError: unsupported operand type(s) for +: 'int' and 'str'
```

In [2]:

```
a=4
b="abhi"
c=a*b
print(c)
```

```
abhiabhiabhiabhi
```

In [3]:

```
5 * 'naman'
```

Out[3]:

```
'namannamannamannamannaman'
```

In [5]:

```
print("C:\nit")# \n it means print new line
```

```
C:
it
```

In [6]:

```
print("r:\nit") #r-raw string we read the path location in future
```

```
r:
it
```

Numbers calculator

In [7]:

```
a=3
b=3
c=a+b
print(a+b)
```

```
6
```

In [8]:

```
a=30
b=5
c=20
d=(a-b*c)/4
```

In [9]:
```
print(d)
```
```
-17.5
```

In [10]:
```
a=10
b=5
c=a/b
print(c)# division always returns a floating point of number
```
```
2.0
```

In [13]:
```
18/7
```
Out[13]:
```
2.5714285714285716
```

In [17]:
```
30//5
```
Out[17]:
```
6
```

In [21]:
```
78%3 #the % opeartor returns the remainder of the division
```
Out[21]:
```
0
```

In [22]:
```
x=30
y=49
x1=x*y+10
print(x1)
```
```
1480
```

In [23]:
```
x=30
y=49
x1=x*y/10
print(x1)
```
```
147.0
```

In [29]:
```
4 ** 2 #4 squared
```
Out[29]:
```
16
```

In [28]:
```
 5 ** 4 # 5 to the power of 4
```
Out[28]:
```
625
```

In [27]:
```
height=20
width=30
width*height
```
Out[27]:
```
600
```

In [31]:
```
abc='abhishek'
print(dfc) # we are try to acess undefined variable
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[31], line 2
      1 abc='abhishek'
----> 2 print(dfc) # we are try to acess un

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

In [32]:
```
10 * 2.19-10 # here we are used floating point and operators mixed type operands
```

Out[32]:
```
11.899999999999999
```

In [35]:
```
tax=50.0/100
```

In [36]:
```
tax
```

Out[36]:
```
0.5
```

In [37]:
```
price=100.05
price*tax
```

Out[37]:
```
50.025
```

In [41]:
```
price + _
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[41], line 1
----> 1 price + _

TypeError: unsupported operand type(s) for +: 'float' and 'str'
```

In [46]:
```
round('pen') # it sis a one type of function
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[46], line 1
----> 1 round(      )

TypeError: type str doesn't define __round__ method
```

In [47]:
```
round(_,2)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[47], line 1
----> 1 round(_,2)

TypeError: type str doesn't define __round__ method
```

Text/Concatinated

In [48]:
```
'Syntax error'
```

```
Out[48]: 'Syntax error'
```

```
In [49]: 'hey finally you are achived ur goal:)!yesss!' # using single quote
```

```
Out[49]: 'hey finally you are achived ur goal:)!yesss!'
```

```
In [54]: " can i bring  some water for you ->no thanks its ok!$hey  how much price of water
```

```
Out[54]: ' can i bring  some water for you ->no thanks its ok!$hey  how much price of water
         bottle'
```

```
In [55]: '2026'# digits and numerals enclosed in qoutes are also strings
```

```
Out[55]: '2026'
```

```
In [56]: 'doesn\'t'#use \' to escape the simgle qoute....
```

```
Out[56]: "doesn't"
```

```
In [58]: "doesn't'"
```

```
Out[58]: "doesn't'"
```

```
In [59]: '"yes,"they said.'
```

```
Out[59]: '"yes,"they said.'
```

```
In [62]: "'what hapend'\'t'! not anything." #...use double quotes and inside use single qout
```

```
Out[62]: "'what hapend''t'! not anything."
```

```
In [6]: "\"yes,\"they said."
```

```
Out[6]: '"yes,"they said.'
```

```
In [7]: '"isn\'t," they said.'
```

```
Out[7]: '"isn\'t," they said.'
```

```
In [8]: s='first name .\second name' #\n means new line
        print(s)
```

```
first name .\second name
```

```
In [9]: print("C:\some\n ame") # here n means new line
```

```
C:\some
 ame
```

```
In [16]: print(r"C: whats\ happen") #note the r before the quote aldo it is not start with a
```

```
C: whats\n happen
```

In [22]:

```
print('''\
      use:thingy [option]
      -h

      -H  hostname
      ''') #Display thuse message first line
# hostname to onnect to
```

```
     use:thingy [option]
     -h

     -H  hostname
```

In [23]:

```
2 * 'un'+'ium'# 2 times 'un' followed by 'ium'
```

Out[23]:

```
'ununium'
```

In [24]:

```
10 * 'Hydrabad'+'Telngana' # string can be concatinated with the + operator,and rep
```

Out[24]:

```
'HydrabadHydrabadHydrabadHydrabadHydrabadHydrabadHydrabadHydrabadHydrabadHydrabadT
elngana'
```

In [2]:

```
print('hy' 'drabad')
print('od' 'isha')
print('bang' 'lore')
print('py' 'thon')
```

```
hydrabad
odisha
banglore
python
```

In [31]:

```
text=('Today hava a good day for everyone '
      'so i hav a excited .')
print(text)# put several strings within parenthess ti have join togather
```

```
Today hava a good day for everyone so i hav a excited .
```

In [15]:

```
perfix='py'
perfix='thon'
print(perfix)
```

```
thon
```

In [10]:

```
('un' * 3) 'ium'
```

```
  Cell In[10], line 1
    ('un' * 3) 'ium'
               ^
SyntaxError: invalid syntax
```

In [5]:

```
3*('in'+'unm')
```

Out[5]:

```
'inunminunminunm'
```

```
 In [6]: perfix

Out[6]:  'thon'

 In [8]: perfix+'power' # if you want to concantinate variables or variables and lateral,use

Out[8]:  'thonpower'

In [16]: word='welcome'
         word[0] #character in position 0

Out[16]: 'w'

In [17]: word[1]

Out[17]: 'e'

In [24]: word[-1]

Out[24]: 'e'

In [25]: word[-4]

Out[25]: 'c'

In [26]: 'print'+'validformat'

Out[26]: 'printvalidformat'

In [27]: word

Out[27]: 'welcome'

In [28]: word[0:3] #character from position 0 (include) to3 (excluded)

Out[28]: 'wel'

In [29]: word[2:5]

Out[29]: 'lco'

In [30]: word[:2]# character from the begning to position to 2(excluded)

Out[30]: 'we'

In [31]: word[4:]# character fron position 4 (included) to the end

Out[31]: 'ome'

In [32]: word[-2:]# character from the second-last (included) to the end
```

```
Out[32]: 'me'

In [33]: word[:2] +word[2:]

Out[33]: 'welcome'

In [34]: word[:4]+word[4:]

Out[34]: 'welcome'

In [35]: word[10] #the word oniy ha s a 7 character

---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[35], line 1
----> 1 word[10]

IndexError: string index out of range

In [36]: word[4:10]

Out[36]: 'ome'

In [37]: word[10:]

Out[37]: ''

In [38]: word[0]

Out[38]: 'w'

In [39]: word[:2]

Out[39]: 'we'

In [42]: 'Most'+word[0:] #concatinate to the

Out[42]: 'Mostwelcome'

In [43]: word[0:]+'to my world'

Out[43]: 'eto my world'

 In [2]: s='neccessarylyadvicepleasedonotreturnoofmyworld'
         print(s)

        neccessarylyadvicepleasedonotreturnoofmyworld

 In [3]: len(s)

 Out[3]: 45
```

List string

In [4]:

```
squares=[1,2,3,4,5,6,7,8,9]
print(squares)
```

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

In [5]:

```
len(squares)
```

Out[5]:

```
9
```

In [6]:

```
squares
```

Out[6]:

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

In [7]:

```
squares[0]
```

Out[7]:

```
1
```

In [8]:

```
squares[-4:]# slicing returns a new list
```

Out[8]:

```
[6, 7, 8, 9]
```

In [9]:

```
squares[-1]
```

Out[9]:

```
9
```

In [10]:

```
squares
```

Out[10]:

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

List Concatenation

In [11]:

```
squares+[10,11,12,13,14,15]
```

Out[11]:

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

In [17]:

```
price=[100,30,20,50]
print(price)
```

```
[100, 30, 20, 50]
```

In [18]:

```
price.append(60)
print(price)
```

```
[100, 30, 20, 50, 60]
```

In [20]:

```
60 **3
```

Out[20]:

```
216000
```

```
In [21]: price
```

```
Out[21]: [100, 30, 20, 50, 60]
```

```
In [22]: cubes=[1,8,27,65,125]
         print(cubes)
```

```
[1, 8, 27, 65, 125]
```

```
In [31]: cubes[3]=64
```

```
In [32]: cubes
```

```
Out[32]: [1, 8, 27, 64, 125]
```

```
In [34]: price=[1,2,3,4,5]
         print(price)
```

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

```
In [38]: price[2]**3
```

```
Out[38]: 27
```

```
In [41]: price
```

```
Out[41]: [1, 2, 3, 4, 5]
```

```
In [42]: price.append(6)
```

```
In [43]: price
```

```
Out[43]: [1, 2, 3, 4, 5, 6]
```

```
In [48]: price.append(6** 3)
```

```
In [50]: price
```

```
Out[50]: [1, 2, 3, 4, 5, 6, 216, 343, 216]
```

```
In [6]: nums=[1,2,44,5,6]
```

```
In [7]: nums
```

```
Out[7]: [1, 2, 44, 5, 6]
```

```
In [8]: nums.append(6 ** 3)
```

```
In [9]: nums
```

```
Out[9]: [1, 2, 44, 5, 6, 216]
```

```
In [12]: 5**4
```

```
Out[12]: 625
```

```
In [13]: nums[3]
```

```
Out[13]: 5
```

```
In [15]: nums +[217,218,219,220]
```

```
Out[15]: [1, 2, 44, 5, 6, 216, 217, 218, 219, 220]
```

```
In [21]: rgb=['Red','Green','Blue']
         rgba =rgb
          # they reference the same object
```

```
In [22]: id(rgb)
```

```
Out[22]: 2034299252160
```

```
In [23]: id(rgba)
```

```
Out[23]: 2034299252160
```

```
In [24]: id(rgb) == id(rgba)
```

```
Out[24]: True
```

```
In [25]: print(rgb)
         print(rgba)
```

```
['Red', 'Green', 'Blue']
['Red', 'Green', 'Blue']
```

```
In [2]: index="welcome"
        print(index)
```

```
welcome
```

```
In [3]: len(index)
```

```
Out[3]: 7
```

Package-Module-Function

package-collection of module

module-collectoin of function

function-it is mainly 2 types - inbuilt-function -userdefine-function

```
In [4]: 'welcome'
```

```
Out[4]: 'welcome'
```

```
In [6]: index[1:5]
```

```
Out[6]: 'elco'
```

```
In [8]: index[5:]
```

```
Out[8]: 'welco'
```

```
In [9]: index[0:5]
```

```
Out[9]: 'welco'
```

```
In [10]: index[:5]
```

```
Out[10]: 'welco'
```

```
In [11]: index[5:]
```

```
Out[11]: 'me'
```

```
In [12]: index[:]
```

```
Out[12]: 'welcome'
```

```
In [13]: index
```

```
Out[13]: 'welcome'
```

```
In [14]: index[4:10]
```

```
Out[14]: 'ome'
```

```
In [15]: index[:10]
```

```
Out[15]: 'welcome'
```

```
In [17]: index[10:]
```

```
Out[17]: ''
```

```
In [18]: index[10]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[18], line 1
----> 1 index[10]

IndexError: string index out of range
```

```
In [29]: s3="hello python"
         print(s3)
```

```
hello python
```

```
In [30]: len(s3)
```

```
Out[30]: 12
```

```
In [31]: s3
```

```
Out[31]: 'hello python'
```

```
In [28]:
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[28], line 1
----> 1 s3.index(4)

TypeError: index() argument 1 must be str, not int
```

```
In [32]: s3[:4]
```

```
Out[32]: 'hell'
```

```
In [33]: s3[-10]
```

```
Out[33]: 'l'
```

```
In [35]: i5='nareshit'
         print(i5)
```

```
nareshit
```

```
In [36]: len(i5)
```

```
Out[36]: 8
```

```
In [38]: i5[0:9]
```

```
Out[38]: 'nareshit'
```

```
In [39]: i5[:]
```

```
Out[39]: 'nareshit'
```

```
In [40]: i5[4:]

Out[40]: 'shit'

In [41]: i5[3:]

Out[41]: 'eshit'

In [42]: i5[3:7]

Out[42]: 'eshi'

In [44]: i5[:4]

Out[44]: 'nare'

In [45]: i5[0:7:3]

Out[45]: 'nei'

In [46]: i5[2:-1]

Out[46]: 'reshi'

In [48]: i5[::-2]

Out[48]: 'thea'

In [49]: i5[::2]

Out[49]: 'nrsi'

In [50]: i5[::3]

Out[50]: 'nei'

In [52]: i5[::-2]

Out[52]: 'thea'

In [53]: i5

Out[53]: 'nareshit'

In [54]: i5[::-3]

Out[54]: 'tsa'

 In [1]: letters=['a','b','c','d']
         print(letters)

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

In [2]:

```
len(letters)
```

Out[2]:

```
4
```

In [3]:

```
letters+['e','f','g','h']
```

Out[3]:

```
['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']
```

In [7]:

```
a=['a','b','c']
n=[1,2,3]
x=[a,n]
print(x)
```

```
[['a', 'b', 'c'], [1, 2, 3]]
```

In [8]:

```
x[0]
```

Out[8]:

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

In [9]:

```
x[1]
```

Out[9]:

```
[1, 2, 3]
```

In [10]:

```
x[0][1]
```

Out[10]:

```
'b'
```

In [19]:

```
print (x[0][0])
print(x[0][1])
print(x[0][2])
```

```
a
b
c
```

In [21]:

```
print(x[1][0])
print(x[1][1])
print(x[1][2])
```

```
1
2
3
```

In [22]:

```
i=256*256
print('The value of i is',i)
```

```
The value of i is 65536
```

In [25]:

```
python='welcome'
print(python)
```

```
welcome
```

In [28]:

```
python[0:5:-2]
```

```
Out[28]: ''
```

```
In [1]: l3=[1,2,3,4,5]
        print(l3)
```

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

```
In [2]: l4=[]
        print(l4)
```

```
[]
```

```
In [4]: l4.extend(l3)
        print(l4)
```

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

```
In [5]: l4.append(6)
        print(l4)
```

```
[1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6]
```

```
In [7]: l5=[]
        print(l5)
```

```
[]
```

```
In [9]: l5=l4.copy()
        print(l5)
```

```
[]
```

```
In [10]: l4
```

```
Out[10]: []
```

```
In [11]: l4=[1,2,3,4,4]
         print(l4)
```

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

```
In [12]: l5=l4.copy()
```

```
In [13]: l5
```

```
Out[13]: [1, 2, 3, 4, 4]
```

```
In [14]: l5.append(5)
         print(l5)
```

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

```
In [15]: print(l4)
         print(l5)
```

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

In [27]:

```
l6=[1,2,3,4,5]
print(l6)
```

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

In [28]:

```
x=[1,2,3,4,5]
print(x)
```

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

In [29]:

```
y=[]
print(y)
```

```
[]
```

In [30]:

```
y.extend(x)
print(y)
```

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

In [31]:

```
print(x)
print(y)
```

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

In [34]:

```
x.append(6)
print(x)
```

```
[1, 2, 3, 4, 5, 6]
```

In [35]:

```
print(x)
print(y)
```

```
[1, 2, 3, 4, 5, 6]
[1, 2, 3, 4, 5, 6]
```

In [46]:

```
patato=30
tamato=40
ginger=100
sum=patato+tamato+ginger
print('The Price  of patato',patato,'Rupeers and tamato',tamato,'Rupees or ginger',
print('The Price of patato{} and Tamato{} or ginger{} total :-{}'.format(patato,tam
print(f'the Price of patato{patato}andtamato{tamato} or ginger{ginger} total:-',sum
```

```
The Price  of patato 30 Rupeers and tamato 40 Rupees or ginger 100 Rupees Total_vegi
table:- 170
The Price of patato30 and Tamato40 or ginger100 total :-170
the Price of patato30andtamato40 or ginger100 total:- 170
```

In [24]:

```
a=['a','b','c']
n=[1,2,3]
x=[a,n]
print(x)
```

```
[['a', 'b', 'c'], [1, 2, 3]]
```

In [25]:

```
x[0]
```

```
Out[25]:  ['a', 'b', 'c']
```

```
In [30]:  x[0][1]
```

```
Out[30]:  'b'
```

```
In [32]:  x[0][2]
```

```
Out[32]:  'c'
```

```
In [44]:  print (x[1][0])
          print(x[1][1])
          print(x[1][2])
```

```
1
2
3
```

```
In [45]:  print(x[0][0])
          print(x[0][1])
          print(x[0][2])
```

```
a
b
c
```

```
In [47]:  "hello" + " world"
```

```
Out[47]:  'hello world'
```

```
In [ ]:
```