In [ ]:
```
# TASK-2 Numbers,Text String index Basic Task.
```

In [2]:
```
spam=1
spam
```

Out[2]: 1

In [4]:
```
text="#this is not a comment line because its inside the quotes."
text
```

Out[4]: '#this is not a comment line because its inside the quotes.'

In [5]:
```
2+2
```

Out[5]: 4

In [6]:
```
50-5*6
```

Out[6]: 20

In [7]:
```
(50-5*6)/4
```

Out[7]: 5.0

In [8]:
```
(50-5*6)//4
```

Out[8]: 5

In [9]:
```
8/5
```

Out[9]: 1.6

In [10]:
```
8//5
```

Out[10]: 1

In [11]:
```
17/3
```

Out[11]: 5.666666666666667

In [12]:
```
17//3
```

Out[12]: 5

In [13]:
```
17*4
```

Out[13]: 68

In [14]:
```
17%3
```

Out[14]: 2

In [15]:
```
5*2-4+2-1
```

Out[15]: 7

In [16]:
```
5*3+2
```

Out[16]: 17

In [17]:
```
5**2
```

Out[17]: 25

In [18]:
```
2**7
```

Out[18]: 128

In [19]:
```
width=20
```

In [20]:
```
height=5*9
```

In [21]:
```
width*height
```

Out[21]: 900

In [22]:
```
n
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[22], line 1
----> 1 n

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

In [23]:
```
4*3.75-1
```

Out[23]: 14.0

In [24]:
```
tax=12.5/100
```

In [25]:
```
price=100.50
```

In [26]:
```
price*tax
```

Out[26]: 12.5625

In [27]:
```
price+_  #here the value of (_) is 12.5625 taken from previous output.
```

Out[27]: 113.0625

In [28]:
```
round(_,2)
```

Out[28]: 113.06

In [29]:
```
'spam eggs' #these is the single quotes('..')
```

Out[29]: 'spam eggs'

In [30]:
```
"paris rabbit got your back:)! yah!" #these is double quotes("...")
```

Out[30]:
```
'paris rabbit got your back:)! yah!'
```

In [31]:
```
'1975' # it is a string value because  int value is inside the quotes.
```

Out[31]:
```
'1975'
```

In [33]:
```
'doesn\t' # use \ to escape the single quote.
```

Out[33]:
```
'doesn\t'
```

In [34]:
```
"dosen't" # or use double quotes instead of using \.
```

Out[34]:
```
"dosen't"
```

In [35]:
```
'"yes", they said'
```

Out[35]:
```
'"yes", they said'
```

In [38]:
```
"\"yes,\" they said "
```

Out[38]:
```
'"yes," they said '
```

In [40]:
```
'"Isn\'t," they said"'
```

Out[40]:
```
'"Isn\'t," they said"'
```

In [45]:
```
s='first Line.\n Second line.'
s
```

Out[45]:
```
'first Line.\n Second line.'
```

In [46]:
```
print(s)
```

```
first Line.
 Second line.
```

In [47]:
```
print('c:\same\name')
```

```
c:\same
ame
```

```
<>:1: SyntaxWarning: invalid escape sequence '\s'
<>:1: SyntaxWarning: invalid escape sequence '\s'
C:\Users\P HARI PRASAD\AppData\Local\Temp\ipykernel_11584\309634723.py:1: SyntaxW
arning: invalid escape sequence '\s'
  print('c:\same\name')
```

In [48]:
```
print(r'c:\same\name')
```

```
c:\same\name
```

In [49]:
```
print("""\
      Usage: thingy [Options]
      -h                Display this usage message
      -H                Hostname to connet to
      """)
```

```
Usage: thingy [Options]
-h              Display this usage message
-H              Hostname to connet to
```

In [51]:
```
4*'un'+'ium'
```

Out[51]: 'unununununium'

In [52]:
```
'py''thon'
```

Out[52]: 'python'

In [57]:
```
text=('put several strings within parenthases ''to have them joined together.')
```

In [58]:
```
text
```

Out[58]: 'put several strings within parenthases to have them joined together.'

In [59]:
```
prefix='py'
```

In [60]:
```
past='thon'
```

In [61]:
```
prefix +past
```

Out[61]: 'python'

In [62]:
```
prefix + 'thon' # these is concat of two string in one line.
```

Out[62]: 'python'

In [63]:
```
word ='python'
word
```

Out[63]: 'python'

In [66]:
```
word[0]
```

Out[66]: 'p'

In [70]:
```
word[0:6] # these call slicing
```

Out[70]: 'python'

In [71]:
```
len(word)
```

Out[71]: 6

In [72]:
```
word[-1]
```

Out[72]: 'n'

In [78]:
```
word[-4]
```

Out[78]: 't'

In [79]:
```
word[-6]
```

Out[79]: 'p'

In [80]:
```
word[0:2]
```

Out[80]: 'py'

In [82]:
```
word[2:6]
```

Out[82]: 'thon'

In [83]:
```
word[:2]
```

Out[83]: 'py'

In [84]:
```
word[4:]
```

Out[84]: 'on'

In [85]:
```
word[-2:]
```

Out[85]: 'on'

In [87]:
```
word[:2] + word[2:]
```

Out[87]: 'python'

In [88]:
```
word[42]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[88], line 1
----> 1 word[42]

IndexError: string index out of range
```

In [89]:
```
word[4:42]
```

Out[89]: 'on'

In [90]:
```
word[42:]
```

Out[90]: ''

In [91]:
```
word[0]='J'
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[91], line 1
----> 1 word[0]='J'

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

In [93]:
```
'J'+word[1:6]
```

Out[93]: 'Jython'

In [95]:
```
word[:5] + 'pyth'
```

Out[95]: 'pythopyth'

In [96]:
```
s=' jbskbknvkjgsvlsdknldnknkghksdl'
```

In [98]:
```
len(s) # these len variable is used for finding length string.
```

Out[98]: 31

In [100…
```
_ # here the (_) will take the previous valuse as output.
```

Out[100… 31

In [103…
```
Squares=[1,4,9,24,34]
Squares
```

Out[103… [1, 4, 9, 24, 34]

In [105…
```
Squares[0]
```

Out[105… 1

In [106…
```
Squares[-1:]
```

Out[106… [34]

In [108…
```
Squares[-3:]
```

Out[108… [9, 24, 34]

In [109…
```
Squares + [5,10,15,20,25,30]
```

Out[109… [1, 4, 9, 24, 34, 5, 10, 15, 20, 25, 30]

In [110…
```
cube=[1,2,3,4,5,6,7]
```

In [111…
```
cube
```

Out[111… [1, 2, 3, 4, 5, 6, 7]

In [112…
```
5**2
```

Out[112… 25

In [114…
```
cube[4]=25  # here the value repalce the wrong value.
cube
```

Out[114… [1, 2, 3, 4, 25, 6, 7]

In [115…
```
cube.append(8)
```

In [116…
```
cube
```

Out[116… [1, 2, 3, 4, 25, 6, 7, 8]

In [117…
```
cube.append(3**2)
```

In [118…
```
cube
```

Out[118… [1, 2, 3, 4, 25, 6, 7, 8, 9]

In [119…
```
rgb =["red","Green","Blue"] #here we creating the strings
```

In [120…
```
rgba=rgb                  # here we are given the value equal with another string
```

In [121…
```
id(rgb)==id(rgba)      # id is (Address of the memorey location).
```

Out[121… True

In [122…
```
rgba.append("Alpha")    # append is used add extra string to the exsiting string
```

In [123…
```
rgb
```

Out[123… ['red', 'Green', 'Blue', 'Alpha']

In [124…
```
correct_rgba=rgba[:]
```

In [126…
```
correct_rgba[-1]="Alpha"
```

In [127…
```
correct_rgba
```

Out[127… ['red', 'Green', 'Blue', 'Alpha']

In [128…
```
rgba
```

Out[128… ['red', 'Green', 'Blue', 'Alpha']

In [129…
```
letter= ['a','b','c','d','e','f','g']
```

In [130…
```
letter
```

Out[130… ['a', 'b', 'c', 'd', 'e', 'f', 'g']

In [131…
```
letter[2:5]
```

Out[131… ['c', 'd', 'e']

In [132…
```
letter[2:5]=['C','D','E']
```

In [133…
```
letter
```

Out[133… ['a', 'b', 'C', 'D', 'E', 'f', 'g']

In [140…
```
letter[2:5]=[]
```

In [142…
```
letter
```

Out[142… ['a', 'b']

In [143…
```
letter[:]=[]
```

In [144…
```
letter
```

Out[144… []

In [145…
```
len(letter)
```

Out[145… 0

In [147…
```
letter=['a','b','c','d','e']
```

In [148…
```
len(letter)
```

Out[148… 5

In [149…
```
# create the lists containg other lists
```

In [150…
```
a=['a','b','c']
```

In [151…
```
n=[1,2,3]
```

In [152…
```
x=[a,n]
x
```

Out[152… [['a', 'b', 'c'], [1, 2, 3]]

In [153…
```
x[0]
```

Out[153… ['a', 'b', 'c']

In [154…
```
x[1]
```

Out[154… [1, 2, 3]

In [155…
```
x[0][1]
```

Out[155… 'b'

In [5]:
```
a,b=0,1
while a<10:
    print(a,end="_")
    a,b=b,a+b
```

```
0_1_1_2_3_5_8_
```

In [7]:
```
i=256*256
print('the value of i is',i)
```

```
the value of i is 65536
```