In [1]:

```
# Creating Empty String-using quotes
S1 = ''
```

In [2]:

```
type(S1)
```

Out[2]:

```
str
```

In [3]:

```
# Creating Empty String-using constructor
S2 = str()
```

In [4]:

```
type(S2)
```

Out[4]:

```
str
```

In [5]:

```
# Creating Non-Empty String-using quotes
S3 = 'Python Programming'
```

In [6]:

```
type(S3)
```

Out[6]:

```
str
```

In [7]:

```
# Creating Non-Empty String-using constructor
S4 = str('VIT')
```

In [8]:

```
type(S4)
```

Out[8]:

```
str
```

In [9]:

```
print(S1)
print(S2)
print(S3)
print(S4)
```

```


Python Programming
VIT
```

In [10]:

```
S5 = S3 + S4
print(S5)
```

```
Python ProgrammingVIT
```

In [11]:

```
print(S3[6])
```

In [12]:

```
print(S3)
print("ID of S3 : ", id(S3))
print(S4)
print("ID of S4 : ", id(S4))
```

```
Python Programming
ID of S3 :  1556499460464
VIT
ID of S4 :  1556499540016
```

In [13]:

```
S3 += S4
print(S3)
print("ID of S3 : ", id(S3))
print(S4)
print("ID of S4 : ", id(S4))
```

```
Python ProgrammingVIT
ID of S3 :  1556499570240
VIT
ID of S4 :  1556499540016
```

In [14]:

```
print(S3)
```

```
Python ProgrammingVIT
```

In [15]:

```
# Accessing individual elements - indexing (positive indexing)
S3[21]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-15-0d288034f9e4> in <module>
      1 # Accessing individual elements - indexing (positive indexing)
----> 2 S3[21]

IndexError: string index out of range
```

In [16]:

```
# Accessing individual elements - indexing (negative indexing)
S3[-5]
```

Out[16]:

```
'n'
```

In [17]:

```
# Length of the string
len(S3)
```

Out[17]:

```
21
```

In [18]:

```
S3 = "Python Programming,VIT"
for i in range(0,len(S3),3):
    print(S3[i])
```

```
P
h
 
o
a
i
,
T
```

In [19]:

```
# Accessing group of elements - slicing
S3[7:18]
```

Out[19]:
```
'Programming'
```

In [20]:
```
# Accessing group of elements - slicing with step value
S3[7:18:3]
```

Out[20]:
```
'Pgmn'
```

In [21]:
```
S3[-15:-5:3]
```

Out[21]:
```
'Pgmn'
```

In [22]:
```
# To Capitalize the first character
Name = 'achu'
M = Name.capitalize()
print(Name.capitalize())
```

```
Achu
```

In [24]:
```
# To Align the string in centre
print(Name.center(9,'*'))
```

```
***achu**
```

In [25]:
```
S3
```

Out[25]:
```
'Python Programming,VIT'
```

In [26]:
```
# To Count the number of occurrences of a character
S3.count('M')
```

Out[26]:
```
0
```

In [27]:
```
# To Check the substring in the beginning
S3.startswith('Python')
```

Out[27]:
```
True
```

In [28]:
```
# To Check the substring at the end
S3.endswith('VIT')
```

Out[28]:
```
True
```

In [29]:
```
# To Check whether the string contains all characters as upper case
S4.isupper()
```

Out[29]:
```
True
```

In [30]:
```
# To Check whether the string contains all characters as lower case
S3.islower()
```

Out[30]:
```
False
```

In [31]:
```
S = 'VIT'
```

In [ ]:
```
S.isupper()
```

In [ ]:
```
S.islower()
```

In [32]:
```
# To find the position of a particular character
S3.index('I')
```

Out[32]:
```
20
```

In [33]:
```
len(S)
```

Out[33]:
```
3
```

In [34]:
```
# To Check whether the string contains all elements as alphabets
S.isalpha()
```

Out[34]:
```
True
```

In [35]:
```
# To Check whether the string contains all elements as digits
S.isdigit()
```

Out[35]:
```
False
```

In [36]:
```
T = 'VIT12345'
```

In [37]:
```
# To Check whether the string contains all elements as alphabets & digits
T.isalnum()
```

Out[37]:
```
True
```

In [38]:
```
S
```

Out[38]:
```
'VIT'
```

In [39]:
```
# To Convert all the elements into upper case
S.lower()
```

Out[39]:
```
'vit'
```

In [40]:
```
S3
```

Out[40]:
```
'Python Programming,VIT'
```

In [41]:

```
# To Convert all the elements into lower case
S3.lower()
```

Out[41]:

```
'python programming,vit'
```

In [42]:

```
S3.upper()
```

Out[42]:

```
'PYTHON PROGRAMMING,VIT'
```

In [43]:

```
# To Convert the first character in each word into upper case
Course = "computer programming python"
Course.title()
```

Out[43]:

```
'Computer Programming Python'
```

In [44]:

```
# To swap upper to lower and lower to upper cases
S3.swapcase()
```

Out[44]:

```
'pYTHON pROGRAMMING,vit'
```

In [45]:

```
# To split the group of characters using the delimiters
print(S3)
S3.split(',')
```

```
Python Programming,VIT
```

Out[45]:

```
['Python Programming', 'VIT']
```

In [46]:

```
S4 = "Python* Java* C* C++* FORTRAN* BASIC"
S4.split('*')
```

Out[46]:

```
['Python', ' Java', ' C', ' C++', ' FORTRAN', ' BASIC']
```

In [47]:

```
# To find the element with maximum ASCII value
max(S3)
```

Out[47]:

```
'y'
```

In [48]:

```
# To find the element with minimum ASCII value
min(S3)
```

Out[48]:

```
' '
```

In [49]:

```
# To remove spaces on left side
Q = '      VIT      '
Q.lstrip()
```

Out[49]:

```
'VIT      '
```

In [50]:

```
# To remove spaces on right side
Q.rstrip()
```

```
Out[50]: '      VIT'
```

```
In [51]: # To remove spaces on both left and right side
         Q.strip()
```

```
Out[51]: 'VIT'
```

```
In [ ]: S3
```

```
In [54]: # To Replace a substring with another substring
         S3.replace('C', 'Java')
```

```
Out[54]: 'Python Programming,VIT'
```

```
In [55]: temp = 'A'
```

```
In [56]: # To find the ASCII value of a character
         ord(temp)
```

```
Out[56]: 65
```

```
In [57]: # To find the character of an ASCII value
         chr(100)
```

```
Out[57]: 'd'
```

```
In [58]: S3
```

```
Out[58]: 'Python Programming,VIT'
```

```
In [59]: S3[-5:-16:-1]
```

```
Out[59]: 'gnimmargorP'
```

```
In [ ]:
```