- capitalize/upper/lower/
- count
- replace

In [1]:
```
str1='welcome'
str1.replace('l','L')
```

Out[1]:
```
'weLcome'
```

index

In [2]:
```
str1='Python'
str1.index('y')
```

Out[2]:
```
1
```

In [3]:
```
str1='Python'
str1.index('n')
```

Out[3]:
```
5
```

In [7]:
```
#index function will return the lowest index
str1='Hai hai hai'
str1.index('i')
```

Out[7]:
```
2
```

In [26]:
```
#index function will return the lowest index
str1='Hai hai hai hia'
i2=str1.index('a')
i3=str1.index('a',i2+1)
i4=str1.index('a',i3+1)
i5=str1.index('a',i4+1)
i2,i3,i4,i5
# i4=str1.index('a',str1.index('a',str1.index('a',i5=str1.index('a',i4+1)+1)+1)
i4
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[26], line 8
      6 i5=str1.index('a',i4+1)
      7 i2,i3,i4,i5
----> 8 i4=str1.index('a',str1.index('a',str1.index('a',i5=str1.index('a',i4+
1)+1)+1))
      9 i4

TypeError: str.index() takes no keyword arguments
```

In [27]:
```
for i in range(len(str2)):
    if str2[i]=='a':
        print(i)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 for i in range(len(str2)):
      2     if str2[i]=='a':
      3         print(i)

NameError: name 'str2' is not defined
```

- if substring not found
- replace: same string
- count:0
- index:error

-find

In [32]:
```
str2='Hai hai hai hia'
# str2.index('a',3,8)
str2.find('a',5)
```

Out[32]:
```
5
```

In [34]:
```
str1='Hai hai hai hia'
i2=str1.find('a')
i3=str1.find('a',i2+1)
i4=str1.find('a',i3+1)
i5=str1.find('a',i4+1)
i6=str1.find('a',i5+1)
i2,i3,i4,i5,i6
```

Out[34]:
```
(1, 5, 9, 14, -1)
```

In [53]:
```
s1= 'omkar.nallagoni@cognizant.com'
s2= 'virat.kohli@rcb.com'
s3= 'lokesh.rahul@lsg.com'
s5= 'a.b@c'
# firstname='omkar'
# lastname='nallagoni'
# cname='cognizant'

#idea1 excat.index s1[0:<.index]
#idea2  extraxt @index s1[.:@]
#idea3 extract second .index s1[@:.]


i1=s1.index('.')
i2=s1.index('@')
i3=s1.index('.com')
# i1,i2,i3
i11=s2.index('.')
i22=s2.index('@')
i33=s2.index('.com')
#i11,i22,i33
i111=s3.index('.')
i222=s3.index('@')
i333=s3.index('.com')
# i111,i222,i333
print("firstname",s1[0:i1])
print("lastname",s1[i1+1:i2])
print("cname",s1[i2+1:i3])
print("firstname",s2[0:i11])
print("lastname",s2[i11+1:i22])
print("cname",s2[i22+1:i33])
print("firstname",s3[0:i111])
print("lastname",s3[i111+1:i222])
print("cname",s3[i222+1:i333])
```

```
firstname omkar
lastname nallagoni
cname cognizant
firstname virat
lastname kohli
cname rcb
firstname lokesh
lastname rahul
cname lsg
```

lstrip - rstrip - strip

In [59]:
```
str1=' python'
str2='python   '
str3='   python'

#strip will avoid left and right space
# str1.strip
# str2.lstrip()
# str3.rstrip()
str1.strip,str2.lstrip(),str3.rstrip()
```

Out[59]:
```
(<function str.strip(chars=None, /)>, 'python   ', '   python')
```

In [58]:
```
str2.strip()
```

Out[58]:
```
'python'
```

In [64]:
```
str2='@@@@@python@@@@@'
# str2.strip('@')
# str2.rstrip('@')
str2.lstrip('@')
```

Out[64]:
```
'python@@@@@'
```

In [68]:
```
str3='@@@@@python$$$$$'
# str2.strip('@')
# str2.rstrip('@')
str3.lstrip('@').rstrip('$')
```

Out[68]:
```
'python'
```

In [69]:
```
str3.strip('@$')
```

Out[69]:
```
'python'
```

In [73]:
```
str1='hello /r/r/r/n how are you'
#op 'hello hpw are you'
str1.strip('/rn')
```

Out[73]:
```
'hello /r/r/r/n how are you'
```

Startswith-endswith

In [76]:
```
s1='hai how are you'
s1.startswith('hai')
```

Out[76]:
```
True
```

In [77]:
```
s1.endswith('u')
```

Out[77]:
```
True
```

split

In [78]:
```
#it will split thw owrds and output will be in the list
s1='Hai how are uoi'
s1.split()
```

Out[78]:
```
['Hai', 'how', 'are', 'uoi']
```

In [79]:
```
s1='Hai how, are uoi'
s1.split(',')
```

Out[79]:
```
['Hai how', ' are uoi']
```

In [80]:
```
s1='Hai how, are uoi'
s1.split('a')
```

Out[80]:
```
['H', 'i how, ', 're uoi']
```

- capital/upper/lower/casefold
- replace
- count
- index/find
- rstrip/lstrip/strip
- startswith/endswith

In [82]:
```
dir('string')
```

Out[82]:
```
['__add__',
 '__class__',
 '__contains__',
 '__delattr__',
 '__dir__',
 '__doc__',
 '__eq__',
 '__format__',
 '__ge__',
 '__getattribute__',
 '__getitem__',
 '__getnewargs__',
 '__getstate__',
 '__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',
 'removeprefix',
 'removesuffix',
 'replace',
 'rfind',
 'rindex',
 'rjust',
 'rpartition',
 'rsplit',
 'rstrip',
 'split',
 'splitlines',
 'startswith',
 'strip',
 'swapcase',
 'title',
 'translate',
 'upper',
 'zfill']
```

In [ ]:
```
'isalnum',
'isalpha',
'isascii',
'isdecimal',
'isdigit',
'isidentifier',
'islower',
'isnumeric',
'isprintable',
'isspace',
'istitle',
'isupper',
```

In [85]:
```
s2='ishwary'
s2.isalpha()
```

Out[85]: True

In [86]:
```
s2='ishwary123'
s2.isalnum()
```

Out[86]: True

In [87]:
```
s2='ishwary123'
s2.isascii()
```

Out[87]: True

In [88]:
```
s2='ishwary123'
s2.isdecimal()
```

Out[88]:
```
False
```

In [ ]:
```
s2='ishwary123'
s2.isdecimal()
```