Day 4 Strings List

February 5, 2023

[2]:
```
name="Data science Masters"
```

[3]:
```
name.swapcase()
```

[3]:
```
'dATA SCIENCE mASTERS'
```

[4]:
```
name.title()
```

[4]:
```
'Data Science Masters'
```

[6]:
```
name.capitalize()
```

[6]:
```
'Data science masters'
```

0.1 Reversing a String

[7]:
```
name[::-1]
```

[7]:
```
'sretsaM ecneics ataD'
```

[10]:
```
' '.join(reversed(name))
```

[10]:
```
's r e t s a M   e c n e i c s   a t a D'
```

[14]:
```
" ".join("abcd")
```

[14]:
```
'a b c d'
```

[17]:
```
' '.join(reversed("ant"))
```

[17]:
```
'tna'
```

[18]:
```
' Pwskills '.join(reversed("ant"))
```

[18]:
```
't Pwskills n Pwskills a'
```

[22]:
```
## typecasting
list(reversed("ant"))
```

[22]:
```
['t', 'n', 'a']
```

[23]:
```
str1="PW is a good company"
```

[ ]:
```
reversed()
```

[25]:
```
## typecasting
list(reversed("ant"))
```

[25]:
```
['t', 'n', 'a']
```

[27]:
```
for i in list(reversed(name)):
    print(i,end='')
```

```
sretsaM ecneics ataD
```

[28]:
```
## Removing character from the end of the string
```

[29]:
```
string_a=" pwskills "
```

[30]:
```
string_a.strip(" ")
```

[30]:
```
'pwskills'
```

[31]:
```
string_a.lstrip(" ")
```

[31]:
```
'pwskills '
```

[32]:
```
string_a.rstrip(" ")
```

[32]:
```
' pwskills'
```

[37]:
```
string_n="Greeting to Pwskills"
string_n.replace("Greeting","Welcome")
```

[37]:
```
'Welcome to Pwskills'
```

[34]:
```
name="Krish"
```

[35]:
```
name="pwskills"
```

[36]:
```
name[0]="k"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[36], line 1
----> 1 name[0]="k"

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

[39]:
```
string_n="Greeting to Pwskills"
var_a=string_n.replace("G","T")
```

[40]:
```
print(id(string_n))
print(id(var_a))
```

```
140496513531328
140496507478592
```

[41]:
```
'hello world'
```

[41]: 'hello world'

[42]:
```
string_n="test@gmail.com"
var_a=string_n.replace("@"," ")
var_a
```

[42]: 'test gmail.com'

[46]:
```
'hello \tworld'.expandtabs()
```

[46]: 'hello   world'

[47]:
```
str1="Welcome to pwskills.Welcome to Dat cience Masters"
```

[49]:
```
str1.replace("Dat","Data").replace("cience","Science")
```

[49]: 'Welcome to pwskills.Welcome to Data Science Masters'

[51]:
```
str1
```

[51]: 'Welcome to pwskills.Welcome to Dat cience Masters'

[52]:
```
str1.isupper()
```

[52]: False

[53]:
```
str1="KRISH"
```

[54]:
```
str1.isupper()
```

[54]: True

[55]:
```
str1.islower()
```

[55]: False

[57]:
```
' '.isspace()
```

[57]: True

[58]:
```
if " ".isspace():
    print("Hello")
```

Hello

[59]:
```
str2="pwskills"
```

[60]:
```
str2.endswith('s')
```

[60]: True

[61]:
```
str2.startswith('p')
```

[61]: True

[62]:
```
## check if all the char in string are alphanumeric
a="abcd1234"
a.isalnum()
```

[62]: True

[63]:
```
## to count the number of character in the strings
```

[64]:
```
count=0
for i in str2:
    count=count+1

print(count)
```

8

[65]:
```
len(str2)
```

[65]: 8

[66]:
```
str2
```

[66]:
```
'pwskills'
```

[67]:
```
for i in str2:
    print(i)
```

```
p
w
s
k
i
l
l
s
```

[69]:
```
range(len(str2))
```

[69]:
```
range(0, 8)
```

[70]:
```
for i in range(len(str2)):
    print(i,"=",str2[i])
```

```
0 = p
1 = w
2 = s
3 = k
4 = i
5 = l
6 = l
7 = s
```

[71]:
```
# We can use index to iterate string reverse direction
```

[73]:
```
len(str2)-1
```

[73]:
```
7
```

[72]:
```
for i in range(len(str2)-1,-1,-1):
    print(str2[i])
```

```
s
l
l
i
k
s
w
p
```

[77]:
```
string="pwskills"
ch=len(string)-1
ch
```

[77]: 7

[80]:
```
string[7],string[6],string[5]
```

[80]: ('s', 'l', 'l')

[75]:
```
while ch>=0:
    print(string[ch])
    ch=ch-1
```

[75]: 7

[83]:
```
#Best solution
for i in range(len(string)):
     print(string[len(string) - (i+1)],end="")
```

```
sllikswp
```

[84]:
```
Name = "pwskills"
vowels = "AaEeIiOoUu"
```

[87]:
```
for ch in Name:
    if ch in vowels:
        print("{} is a vowel".format(ch))
    else:
        print("{} is not a vowel".format(ch))
```

```
p is not a vowel
w is not a vowel
s is not a vowel
k is not a vowel
i is a vowel
l is not a vowel
l is not a vowel
s is not a vowel
```

0.2 List

[89]:
```
type([])
```

[89]: list

[90]:
```
["Krish","Naik","Pwskills",32]
```

[90]: ['Krish', 'Naik', 'Pwskills', 32]

[91]:
```
list([1,2,3,4,5])
```

[91]: [1, 2, 3, 4, 5]

[93]:
```
str2
```

[93]: 'pwskills'

[94]:
```
list(str2)
```

[94]: ['p', 'w', 's', 'k', 'i', 'l', 'l', 's']

[95]:
```
str1="PW skills Data science masters"
```

[96]:
```
list(str1.split(" "))
```

[96]: ['PW', 'skills', 'Data', 'science', 'masters']

[98]:
```
lst1=str1.split(" ")
```

[101]:
```
lst1[1:]
```

[101]: ['skills', 'Data', 'science', 'masters']

[103]:
```
lst1[2]="Datas"
```

[104]:
```
lst1
```

[104]: ['PW', 'skills', 'Datas', 'science', 'masters']

[108]:
```
lst1[::-2]
```

[108]: ['masters', 'Datas', 'PW']

[112]:
```
lst1[-5::-2]
```

[112]: ['PW']

[114]:
```
lst1
```

[114]: ['PW', 'skills', 'Datas', 'science', 'masters']

[118]:
```
## concatenation operation
lst1 + ['new element',3]
```

[118]:
```
['PW', 'skills', 'Datas', 'science', 'masters', 'new element', 3]
```

[120]:
```
## concatenation operation
lst2=lst1 + [['new element',3]]
```

[124]:
```
lst2[-1][0]
```

[124]:
```
'new element'
```

[127]:
```
lst1 * 2
```

[127]:
```
['PW',
 'skills',
 'Datas',
 'science',
 'masters',
 'PW',
 'skills',
 'Datas',
 'science',
 'masters']
```

[128]:
```
lst1
```

[128]:
```
['PW', 'skills', 'Datas', 'science', 'masters']
```

[135]:
```
if "science" in lst1:
    print("Present")
```

```
Present
```

[137]:
```
for elements in lst1:
    if elements=="science":
        print(elements)
        break
```

```
science
```

[140]:
```
## check elements inside a list
lst=[1,2,3,4]
4 in lst
```

[140]:
```
True
```

[141]:
```
2.0 in lst
```

[141]:
```
True
```

[142]:
```
lst1=["Zebra","Monkey","Donkey","Lion"]
lst2=[5,6,2,9,5,8,6]
```

[143]:
```
print(max(lst1))
```

```
Zebra
```

[144]:
```
max(lst2)
```

[144]: 9

[145]:
```
min(lst1)
```

[145]: 'Donkey'

[146]:
```
min(lst2)
```

[146]: 2

[149]:
```
1.9999999999999999 in [2, 4, 5]
```

[149]: True

[148]:
```
5==5.0
```

[148]: True

[ ]:
```
lst3=[5,6,2,9,5,8,6]
type(lst3[0])  ## output is integer
type(5.0)  ### output is float
5.0 in lst3 ## output is TRUE
```

[147]:
```
## Append
```

[150]:
```
lst=[1,2,3,4,5,6]
```

[151]:
```
lst
```

[151]: [1, 2, 3, 4, 5, 6]

[153]:
```
lst.append("Pwskills")
```

[154]:
```
lst
```

[154]: [1, 2, 3, 4, 5, 6, 'Pwskills']

[155]:
```
lst.append(['Data','Science','Masters'])
```

[156]:
```
lst
```

[156]:
```
[1, 2, 3, 4, 5, 6, 'Pwskills', ['Data', 'Science', 'Masters']]
```

[157]:
```
lst1=["Zebra","Monkey","Donkey","Lion"]
```

[159]:
```
lst1.pop()
```

[159]:
```
'Lion'
```

[162]:
```
lst1.pop(2)
```

[162]:
```
'Donkey'
```

[163]:
```
lst1
```

[163]:
```
['Zebra', 'Monkey']
```

[164]:
```
lst1=["Zebra","Monkey","Donkey","Lion"]
```

[165]:
```
removed_element=lst1.pop(0)
removed_element
```

[165]:
```
'Zebra'
```

[167]:
```
lst1[100]
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
Cell In[167], line 1
----> 1 lst1[100]

IndexError: list index out of range
```

[168]:
```
## Sorting and Reverse method in list
```

[169]:
```
new_list=['q','e','f','s','t','u']
new_list
```

[169]:
```
['q', 'e', 'f', 's', 't', 'u']
```

[170]:
```
new_list[::-1]
```

[170]:
```
['u', 't', 's', 'f', 'e', 'q']
```

[172]:
```
## inplace
new_list.reverse()
```

[173]:
```
new_list
```

[173]:
```
['u', 't', 's', 'f', 'e', 'q']
```

[174]:
```
new_list.sort()
```

[175]:
```
new_list
```

[175]:
```
['e', 'f', 'q', 's', 't', 'u']
```

[184]:
```
new_list.sort(reverse=True)
```

[185]:
```
new_list
```

[185]:
```
['u', 't', 's', 'q', 'f', 'e']
```

[181]:
```
lst=[1,2,3,4,5,6]
lst.append(10)
```

[179]:
```
lst.append(['PW','Skills'])
```

[182]:
```
lst.extend(['PW','Skills'])
```

[183]:
```
lst
```

[183]:
```
[1, 2, 3, 4, 5, 6, 10, 'PW', 'Skills']
```

0.3 Nested List

[186]:
```
# Let's make three lists
lst_1=[1,2,3]
lst_2=[4,5,6]
lst_3=[7,8,9]

# Make a list of lists to form a matrix
matrix = [lst_1,lst_2,lst_3]
```

[187]:
```
matrix
```

[187]:
```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

[190]:
```
matrix[1][2]
```

[190]:
```
6
```

[191]:
```
matrix[2][1:]
```

[191]:
```
[8, 9]
```

[204]:
```
## List Comprehension
[i for i in range(20)]
```

[204]:
```
[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19]
```

[211]:
```
## List Comprehension
## Even numbers
[i if i%2==0 else "ODD" for i in range(0,20)]
```

[211]:
```
[0,
 'ODD',
 2,
 'ODD',
 4,
 'ODD',
 6,
 'ODD',
 8,
 'ODD',
 10,
 'ODD',
 12,
 'ODD',
 14,
 'ODD',
 16,
 'ODD',
 18,
 'ODD']
```

[ ]:
```
## Assignment
## Sum of even numbers and odd numbers
lst=[1,2,3,4,5,6,7,8]
```

[212]:
```
lst = [1,2,3,4,5,6,7,8]

even_sum = 0
odd_sum = 0
for i in lst:
    if i % 2 ==0:
        even_sum += i
    else:
        odd_sum += i
```

```
print(even_sum)
print(odd_sum)
```

```
20
16
```

[216]:
```
even_sum=sum([num for num in lst if num%2==0])
```

[217]:
```
even_sum
```

[217]:
```
20
```

[218]:
```
odd_sum=sum([num for num in lst if num%2!=0])
odd_sum
```

[218]:
```
16
```

[219]:
```
lst=[1,2,3,4,5,6,7,8,9,10]
[num**2 for num in lst]
```

[219]:
```
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
```

[220]:
```
# Example 2: Create a list of only the positive numbers from a given list
numbers = [-2, -1, 0, 1, 2, 3, 4]
[num for num in numbers if num > 0]
```

[220]:
```
[1, 2, 3, 4]
```

[221]:
```
# Example 3: Create a list of only the first letters of words in a list
words = ['apple', 'banana', 'cherry', 'date']
[word[0] for word in words]
```

[221]:
```
['a', 'b', 'c', 'd']
```

[ ]:
```
(9/5)*temp+32
```

[222]:
```
# Example 4: Convert a list of temperatures from Celsius to Fahrenheit uing␣
↪list comprehesnion
celsius_temperatures = [0, 10, 20, 30, 40, 50]
```

[223]:
```
[(9/5)*temp + 32 for temp in celsius_temperatures]
```

[223]:
```
[32.0, 50.0, 68.0, 86.0, 104.0, 122.0]
```

[224]:
```
# Example 5: Flatten a list of lists into a single list
lists = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

[228]:
```
[j for i in lists for j in i]
```

[228]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

[ ]:
```
## Assignment
## Using both code and list comprehesnion
# Example 2: Create a list of only the prime numbers from a given list
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

[ ]:
```
# Example 3: Create a list of all the possible combinations of 2 elements from␣
 ↪a list
numbers = [1, 2, 3, 4, 5]
```

[229]:
```
(1,2),(1,3)
```

[229]:
```
((1, 2), (1, 3))
```

[230]:
```
lst
```

[230]:
```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

[ ]:
```
lst.insert()
```

[232]:
```
c=1+2j
```

[234]:
```
c.real
```

[234]:
```
1.0
```

[ ]:
```

```