Day4:

String

In [105]:

```
#Creating a String<br>


letter = 'P'                # A string could be a single character or a bunch of texts
print(letter)               # P
print(len(letter))          # 1
greeting = 'Hello, World!'  # String could be  a single or double quote,"Hello, World!"
print(greeting)             # Hello, World!
print(len(greeting))        # 13
sentence = "I hope you are enjoying 30 days of Python Challenge"
print(sentence)
```

```
P
1
Hello, World!
13
I hope you are enjoying 30 days of Python Challenge
```

Multiline string is created by using triple single (''') or double quotes ("""). See the example below.

multiline_string = '''I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python.'''
print(multiline_string)

Another way of doing the same thing

multiline_string = """I am a teacher and enjoy teaching.
I didn't find anything as rewarding as empowering people.
That is why I created 30 days of python."""
print(multiline_string)

In [107]:

```
#String Concatenation
#We can connect strings together. Merging or connecting strings is called concatenatio
n. See the #example below:

first_name = 'Asabeneh'
last_name = 'Yetayeh'
space = ' '
full_name = first_name  +  space + last_name
print(full_name) # Asabeneh Yetayeh
# Checking length of a string using len() built-in function
print(len(first_name))  # 8
print(len(last_name))   # 7
print(len(first_name) > len(last_name)) # True
print(len(full_name)) # 15
```

```
Asabeneh Yetayeh
8
7
True
16
```

In [108]:

```
#Escape Sequences in Strings
#In python and other programming languages \ followed by a character. Let's see the mos
t common #escape characters:

#\n: new line
#\t: Tab means(8 spaces)
#\\: Back slash
#\': Single quote (')
#\": Double quote (")
print('I hope everyone is enjoying the Python Challenge.\nAre you ?') # line break
print('Days\tTopics\tExercises')
print('Day 1\t3\t5')
print('Day 2\t3\t5')
print('Day 3\t3\t5')
print('Day 4\t3\t5')
print('This is a backslash  symbol (\\)') # To write a backslash
print('In every programming language it starts with \"Hello, World!\"')
```

```
I hope everyone is enjoying the Python Challenge.
Are you ?
Days    Topics  Exercises
Day 1   3       5
Day 2   3       5
Day 3   3       5
Day 4   3       5
This is a backslash  symbol (\)
In every programming language it starts with "Hello, World!"
```

In [109]:

```
first_name = 'Asabeneh'
last_name = 'Yetayeh'
language = 'Python'
formated_string = 'I am {} {}. I teach {}'.format(first_name, last_name, language)
print(formated_string)
a = 4
b = 3

print('{} + {} = {}'.format(a, b, a + b))
print('{} - {} = {}'.format(a, b, a - b))
print('{} * {} = {}'.format(a, b, a * b))
print('{} / {} = {:.2f}'.format(a, b, a / b)) # limits it to two digits after decimal
print('{} % {} = {}'.format(a, b, a % b))
print('{} // {} = {}'.format(a, b, a // b))
print('{} ** {} = {}'.format(a, b, a ** b))
```

```
I am Asabeneh Yetayeh. I teach Python
4 + 3 = 7
4 - 3 = 1
4 * 3 = 12
4 / 3 = 1.33
4 % 3 = 1
4 // 3 = 1
4 ** 3 = 64
```

In [110]:

```
#index()
language = 'Python'
first_letter = language[0]
print(first_letter) # P
second_letter = language[1]
print(second_letter) # y
last_index = len(language) - 1
last_letter = language[last_index]
print(last_letter) # n
#If we want to start from right end we can use negative indexing. -1 is the last index.

language = 'Python'
last_letter = language[-1]
print(last_letter) # n
second_last = language[-2]
print(second_last) # o
```

```
P
y
n
n
o
```

In [111]:

```
#Slicing Python Strings
#In python we can slice strings into substrings.

language = 'Python'
first_three = language[0:3] # starts at zero index and up to 3 but not include 3
last_three = language[3:6]
print(last_three) # hon
# Another way
last_three = language[-3:]
print(last_three)   # hon
last_three = language[3:]
print(last_three)   # hon
```

```
hon
hon
hon
```

In [112]:

```
#Reversing a String
#We can easily reverse strings in python.

greeting = 'Hello, World!'
print(greeting[::-1]) # !dlroW ,olleH
```

```
!dlroW ,olleH
```

In [113]:

```
#Skipping Characters While Slicing
#It is possible to skip characters while slicing by passing step argument to slice meth
od.

language = 'Python'
pto = language[0:6:2] #
print(pto) # Pto
```

```
Pto
```

String Methods
There are many string methods which allow us to format strings. See some of the string methods in the following example:
capitalize(): Converts the first character of the string to capital Letter
challenge = 'thirty days of python'
print(challenge.capitalize()) # 'Thirty days of python'

count(): returns occurrences of substring in string, count(substring, start=.., end=..)
challenge = 'thirty days of python'
print(challenge.count('y')) # 3
print(challenge.count('y', 7, 14)) # 1
print(challenge.count('th')) # 2`

endswith(): Checks if a string ends with a specified ending
challenge = 'thirty days of python'
print(challenge.endswith('on')) # True
print(challenge.endswith('tion')) # False

expandtabs(): Replaces tab character with spaces, default tab size is 8. It takes tab size argument
challenge = 'thirty\tdays\tof\tpython'
print(challenge.expandtabs()) # 'thirty days of python'
print(challenge.expandtabs(10)) # 'thirty days of python'

find(): Returns the lowest index of the first occurrence of a substring, if not found returns -1
challenge = 'thirty days of python'
print(challenge.find('y')) # 5
print(challenge.find('th')) # 0

rfind(): Returns the highest index of the first occurrence of a substring, if not found returns -1
challenge = 'thirty days of python'
print(challenge.find('y')) # 5
print(challenge.find('th')) # 1

format(): formats string into a nicer output
More about string formating check this link
first_name = 'Asabeneh'
last_name = 'Yetayeh'
job = 'teacher'
country = 'Finland'
sentence = 'I am {} {}. I am a {}. I live in {}.'.format(first_name, last_name, job, country)
print(sentence) # I am Asabeneh Yetayeh. I am a teacher. I live in Finland.

radius = 10
pi = 3.14
area = pi * radius ** 2
result = 'The area of a circle with radius {} is {}'.format(str(radius), str(area))
print(result) # The area of a circle with radius 10 is 314

index(): Returns the lowest index of a substring, additional arguments indicate starting and ending index (default 0 and string length - 1)
challenge = 'thirty days of python'
sub_string = 'da'
print(challenge.index(sub_string)) # 7
print(challenge.index(sub_string, 9)) # error

rindex(): Returns the highest index of a substring, additional arguments indicate starting and ending index (default 0 and string length - 1)
challenge = 'thirty days of python'
sub_string = 'da'
print(challenge.index(sub_string)) # 8
print(challenge.index(sub_string, 9)) # error

isalnum(): Checks alphanumeric character
challenge = 'ThirtyDaysPython'
print(challenge.isalnum()) # True

challenge = '30DaysPython'
print(challenge.isalnum()) # True

challenge = 'thirty days of python'
print(challenge.isalnum()) # False, space is not an alphanumeric character

challenge = 'thirty days of python 2019'
print(challenge.isalnum()) # False

isalpha(): Checks if all string elements are alphabet characters (a-z and A-Z)
challenge = 'thirty days of python'
print(challenge.isalpha()) # False, space is once again excluded
challenge = 'ThirtyDaysPython'
print(challenge.isalpha()) # True
num = '123'
print(num.isalpha()) # False
isdecimal(): Checks if all characters in a string are decimal (0-9)
challenge = 'thirty days of python'
print(challenge.isdecimal()) # False
challenge = '123'
print(challenge.isdecimal()) # True
challenge = '\u00B2'
print(challenge.isdigit()) # False
challenge = '12 3'
print(challenge.isdecimal()) # False, no space allowed

isdigit(): Checks if all characters in a string are numbers (0-9 and some other unicode characters for numbers)
challenge = 'Thirty'
print(challenge.isdigit()) # False
challenge = '30'
print(challenge.isdigit()) # True
challenge = '\u00B2'
print(challenge.isdigit()) # True

isnumeric(): Checks if all characters in a string are numbers or number related (just like isdigit(), just accepts more symbols, like ½)
num = '10'

print(num.isnumeric()) # True
num = '\u00BD' # ½
print(num.isnumeric()) # True
num = '10.5'
print(num.isnumeric()) # False
isidentifier(): Checks for a valid identifier - means it checks, if a string is a valid variable name
challenge = '30DaysOfPython'
print(challenge.isidentifier()) # False, because it starts with a number
challenge = 'thirty_days_of_python'
print(challenge.isidentifier()) # True

islower(): Checks if all alphabet characters in the string are lowercase
challenge = 'thirty days of python'
print(challenge.islower()) # True
challenge = 'Thirty days of python'
print(challenge.islower()) # False

isupper(): Checks if all alphabet characters in the string are uppercase
challenge = 'thirty days of python'
print(challenge.isupper()) # False
challenge = 'THIRTY DAYS OF PYTHON'
print(challenge.isupper()) # True

join(): Returns a concatenated string
web_tech = ['HTML', 'CSS', 'JavaScript', 'React']
result = '# '.join(web_tech)
print(result) # 'HTML# CSS# JavaScript# React'

strip(): Removes all given characters starting from the beggining and end of the string
challenge = 'thirty days of pythoonnn'
print(challenge.strip('noth')) # 'irty days of py'

replace(): Replaces substring with a given string
challenge = 'thirty days of python'
print(challenge.replace('python', 'coding')) # 'thirty days of coding'

split(): Splits the string, using given string as a separator
challenge = 'thirty days of python'
print(challenge.split()) # ['thirty', 'days', 'of', 'python']
challenge = 'thirty, days, of, python'
print(challenge.split(', ')) # ['thirty', 'days', 'of', 'python']

title(): Returns a title cased string
challenge = 'thirty days of python'
print(challenge.title()) # Thirty Days Of Python

swapcase(): Converts all uppercase characters to lowercase and all lowercase characters to uppercase characters
challenge = 'thirty days of python'
print(challenge.swapcase()) # THIRTY DAYS OF PYTHON
challenge = 'Thirty Days Of Python'
print(challenge.swapcase()) # tHIRTY dAYS oF pYTHON

startswith(): Checks if String Starts with the Specified String
challenge = 'thirty days of python'
print(challenge.startswith('thirty')) # True
challenge = '30 days of python'
print(challenge.startswith('thirty')) # False

Concatenate the string 'Thirty', 'Days', 'Of', 'Python' to a single string, 'Thirty Days Of Python'.
Concatenate the string 'Coding', 'For' , 'All' to a single string, 'Coding For All'.
Declare a variable named company and assign it to an initial value "Coding For All".
Print the variable company using print().
Print the length of the company string using len() method and print().
Change all the characters to capital letters using upper() method.
Change all the characters to lowercase letters using lower() method.
Use capitalize(), title(), swapcase() methods to format the value of the string Coding For All.
Cut(slice) out the first word of Coding For All string.
Check if Coding For All string contains a word Coding using the method index, find or other methods.
Replace the word coding in the string 'Coding For All' to Python.
Change Python for Everyone to Python for All using the replace method or other methods.
Split the string 'Coding For All' using space as the separator (split()) .
"Facebook, Google, Microsoft, Apple, IBM, Oracle, Amazon" split the string at the comma.
What is the character at index 0 in the string Coding For All.
What is the last index of the string Coding For All.
What character is at index 10 in "Coding For All" string.
Create an acronym or an abbreviation for the name 'Python For Everyone'.
Create an acronym or an abbreviation for the name 'Coding For All'.
Use index to determine the position of the first occurrence of C in Coding For All.
Use index to determine the position of the first occurrence of F in Coding For All.
Use rfind to determine the position of the last occurrence of l in Coding For All People.
Use index or find to find the position of the first occurrence of the word 'because' in the following sentence: 'You cannot end a sentence with because because because is a conjunction'
Use rindex to find the position of the last occurrence of the word because in the following sentence: 'You cannot end a sentence with because because because is a conjunction'
Slice out the phrase 'because because because' in the following sentence: 'You cannot end a sentence with because because because is a conjunction'
Find the position of the first occurrence of the word 'because' in the following sentence: 'You cannot end a sentence with because because because is a conjunction'
Slice out the phrase 'because because because' in the following sentence: 'You cannot end a sentence with because because because is a conjunction'
Does "Coding For All' start with a substring Coding?
Does 'Coding For All' end with a substring coding?
' Coding For All ' , remove the left and right trailing spaces in the given string.
Which one of the following variables return True when we use the method isidentifier():
30DaysOfPython
thirty_days_of_python
The following list contains the names of some of python libraries: ['Django', 'Flask', 'Bottle', 'Pyramid', 'Falcon']. Join the list with a hash with space string.
Use the new line escape sequence to separate the following sentences.
I am enjoying this challenge.
I just wonder what is next.
Use a tab escape sequence to write the following lines.
Name Age Country
Asabeneh 250 Finland
Use the string formating method to display the following:
radius = 10

area = 3.14 ***radius ** 2***
The area of a cricle with radius 10 is 314 meters square.
Make the following using string formating methods:
8 + 6 = 14
8 - 6 = 2
8 **6 = 48**
8 / 6 = 1.33
8 % 6 = 2
8 // 6 = 1
8 ** 6 = 262144

In [ ]:

```
index(),len(),[],split(),count(),capitlize(),isalpha(),isalnum(),isdigit()
```

In [7]:

```
if __name__ == '__main__':
    a=10 #global
    b=20
    def sum(a,b):
        s=a+b
        d=0#local
        print(d)
        return s

    c=sum(a,b)
#print(d)

    print(c)
```

```
0
30
```

In [ ]:

```
namespace-->global and local
```

In [17]:

```
s='hello i\t\\tdeepak'
print(s)
```

```
hello i \tdeepak
```

In [21]:

```
#string concatention
a="hello"
b="@"
c="python"
s=a+b+c
print(s)
```

```
hello@python
```

In [24]:

```
#index()
a="hehllo"
print(a.index("h"))
```

```
0
```

In [55]:

```
#slicing
a="hello python"
a[0:12:2]
a[4:9]
print(a[-4])
print(a[-12])
print(a[-6:])
print(a[8:4:-1])
print(a[4:8:-1])
```

```
t
h
python
typ
```

In [41]:

```
print(len(a))
for i in range(0,len(a)):
    print(i,a[i])
```

```
12
0 h
1 e
2 l
3 l
4 o
5
6 p
7 y
8 t
9 h
10 o
11 n
```

In [51]:

```
s="working"
if s[-3:]=="ing":
    print("yes")
    s=s[0:4]+"ly"
    print(k)
```

```
yes
workly
```

In [57]:

```
"""fgnsfglskj"""
```

Out[57]:

```
'fgnsfglskj'
```

In [58]:

```

```

```
---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-58-e2040636488c> in <module>
      1 lang="abcdefghijklmno"
----> 2 pto = lang[0,6:2]

TypeError: string indices must be integers
```

In [63]:

```
#captilize
a="hello"
a=a.capitalize()
print(a)
```

```
Hello
```

In [61]:

```
#builtin ,method

print(len(a)) #--> builtin
print(a.capitalize()) #-->method
```

```
5
Hello
```

In [68]:

```
#find
a="hello"
a.find("p")
```

Out[68]:

```
-1
```

In [71]:

```
if "h" in a:
    print(a.index("h"))
else:
    print("-1")
```

```
0
```

In [74]:

```
a="hello python"
a.rindex("h")
```

Out[74]:

```
9
```

In [78]:

```
a=input()
if a.isalpha():
    print("yes")
else:
    print("false")
```

```
asdf
yes
```

In [81]:

```
a=input()
print(a.isalpha())
```

```
@#$
False
```

In [84]:

```
a=input()
print(a.isdigit())
```

```
assf
False
```

In [87]:

```
a=input()
print(a.isalnum())
```

```
!@#
False
```

In [96]:

```
a=input()
for i in a:
    if a[i].isalpha():
        pass
    elif a[i].isdigit():
        pass
    else:
        print("it is a special char")
```

```
123asdas

---------------------------------------------------------------------------
-
TypeError                                 Traceback (most recent call las
t)
<ipython-input-96-99abb131ac31> in <module>
      1 a=input()
      2 for i in a:
----> 3     if a[i].isalpha():
      4         pass
      5     elif a[i].isdigit():

TypeError: string indices must be integers
```

In [100]:

```
a=input()
print(a.endswith("e"))
print(a.islower())
print(a.isupper())
print(a.lower())
print(a.upper())
```

```
abcd
False
True
False
abcd
ABCD
```

In [102]:

```
a=input()
if a.endswith("y"):
    print("y")
else:
    print("no y")
```

```
abcd
no y
```

In [104]:

```
a="hellohpython"
a.rindex("h")
```

Out[104]:

```
9
```

In [108]:

```
a="hellohpython"
b=a[::-1]
print(b.index("h"))
print(len(a)-1-b.index("h"))
```

```
2
9
```

In [9]:

```
a="helloworld"
for i in a:
    print(i)
```

```
h
e
l
l
o
w
o
r
l
d
```

In [10]:

```
a.rfind("lo")
```

Out[10]:

```
3
```

In [17]:

```
a="helloworld world wor"
substring="ld"
a.count(a)
```

Out[17]:

```
1
```

In [26]:

```
a=input().split("-")
print(a)
```

```
ab-cd-ef
['ab', 'cd', 'ef']
```

In [25]:

```
#title
a="hello world"
a.title()
```

Out[25]:

```
'Hello World'
```

In [34]:

```
a=input().split("-")
print(a)
c=[]

for i in a:
    c.append(" ".join(i))
print(c)
```

```
ab-cd-fg
['ab', 'cd', 'fg']
['a b', 'c d', 'f g']
```

In [32]:

```
web_tech = ['HTML', 'CSS', 'JavaScript', 'React']
result = '##'.join(web_tech)
print(result) # 'HTML# CSS# JavaScript# React
```

```
HTML##CSS##JavaScript##React
```

In [35]:

```
a="hello"
a.swapcase()
```

Out[35]:

```
'HELLO'
```

In [39]:

```
a="hello"
c=""
for i in a:
    c+="".join(i.upper())
    print(c)
print(c)
```

```
H
HE
HEL
HELL
HELLO
HELLO
```

In [41]:

```
a="helloWORLD"
c=""
for i in a:
    if i.isupper():
        c+="".join(i.lower())
    else:
        c+="".join(i.upper())


print(c)
```

```
HELLOworld
```

In [46]:

```
a="swa\case"
a.isidentifier()
```

Out[46]:

```
False
```

In [59]:

```
a="aabbb"
c=""
for i in a:
    if i=="a":
        c+="".join(a.replace("a","b"))
    elif i=="b":
        c+="".join(a.replace("b","a"))
print(c)
```

```
bbbbbbbbbbaaaaaaaaaaaaaaa
```

In [91]:

```
a=[10,40,30]
b=20.025
for i in a:
    print("{bab} is not equal to {oui} and {bab}".format(oui=i,bab=round(b)))
```

```
20 is not equal to 10 and 20
20 is not equal to 40 and 20
20 is not equal to 30 and 20
```

In [76]:

```
print("{}{}".format(10,20))
```

```
1020
```

In [77]:

```
a=10
b=20
print("{}{}".format(a,b))
```

```
1020
```

In [87]:

```
print("{n:.{1}f}".format(n=10,p=12))
```

```
---------------------------------------------------------------------------
-
IndexError                                Traceback (most recent call las
t)
<ipython-input-87-c3881b5b2e9f> in <module>
----> 1 print("{n:.{1}f}".format(n=10,p=12))

IndexError: tuple index out of range
```

In [88]:

```
a=10.125
round(a)
```

Out[88]:

```
10
```

In [104]:

```
a="hello world python"
print(a.strip('pho'))
```

```
ello world python
```

In [103]:

```
challenge = 'thirty days of pythoonnn'
print(challenge.strip('noth')) # 'irty days of py'
```

```
irty days of py
```

In [ ]:

```

```