Single line comment

In [1]:
```
letter='P'
```

In [2]:
```
print(letter)
```

```
P
```

In [3]:
```
print(len(letter))
```

```
1
```

In [ ]:
```

```

In [4]:
```
greeting='Hello, World!'
```

In [5]:
```
print(greeting)
print(len(greeting))
```

```
Hello, World!
13
```

In [ ]:
```

```

In [6]:
```
sentence="I hope you are enjoying 30days of python challenge"
```

In [7]:
```
print(sentence)
```

```
I hope you are enjoying 30days of python challenge
```

In [ ]:
```

```

Multiline string

In [8]:
```
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.'''
```

In [9]:
```
print(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.
```

In [ ]:
```

```

In [10]:
```
multiline_string1="""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."""
```

In [11]:
```
print(multiline_string1)
```

```
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.
```

In [ ]:
```

```

String concatenation

In [12]:
```
first_name='Asabaneh'
```

In [13]:
```
last_name='yetayeh'
```

In [14]:
```
space=' '
```

In [15]:
```
full_name=first_name+space+last_name
```

In [16]:
```
print(full_name)
```

```
Asabaneh yetayeh
```

In [22]:
```
print(len(first_name))
print(len(last_name))
print(len(full_name))
```

```
8
7
16
```

In [23]:
```
print(len(first_name) > len(last_name))
```

```
True
```

In [ ]:
```

```

Unpacking characters

In [24]:
```
language = 'Python'
```

In [25]:
```
a,b,c,d,e,f=language  # unpacking sequence characters into variables
```

In [27]:
```
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

```
P
y
t
h
o
n
```

In [ ]:

Accessing characters instrings by index

In [28]:
```
language = 'Python'
```

In [30]:
```
first_letter = language[0]
print(first_letter)
```

```
P
```

In [31]:
```
second_letter = language[1]
print(second_letter)
```

```
y
```

In [34]:
```
last_index=len(language)-1
```

In [35]:
```
last_letter=language[last_index]
```

In [36]:
```
print(last_letter)
```

```
n
```

In [ ]:

In [37]:
```
last_letter1=language[-1]
```

In [38]:
```
print(last_letter)
```

```
n
```

In [39]:
```
second_last=language[-2]
print(second_last)
```

```
o
```

In [ ]:

Slicing

In [40]:
```
language = 'Python'
```

In [43]:
```
first_three=language[:3]
```

```
print(first_three)
```

```
Pyt
```

In [46]:
```
last_three=language[3:6]
last_three1=language[-3:]
```

In [47]:
```
print(last_three)
print(last_three1)
```

```
hon
hon
```

In [ ]:

Skipping character while splitting python strings

In [48]:
```
language='Python'
```

In [49]:
```
pto=language[0:6:2]
```

In [50]:
```
print(pto)
```

```
Pto
```

In [ ]:

Escaping sequence

In [63]:
```
print('I hope every one is enjoying the python challenge.\nDo you?')   # \n = next
```

```
I hope every one is enjoying the python challenge.
Do you?
```

In [64]:
```
'I hope everyone is enjoying the python challenge.\nDo you?'
```

Out[64]:
```
'I hope everyone is enjoying the python challenge.\nDo you?'
```

In [62]:
```
print('Days\tTopics\tExercices')     # \t = tab character
```

```
Days    Topics  Exercices
```

In [81]:
```
print('Day 1\t3\t5')              # first \t =tab character, next \t =double tab c
```

```
Day 1   3       5
```

In [91]:
```
print('Day   1\t3\t5')
```

```
Day   1 3       5
```

In [90]:
```
print('Day    1\t3\t5')
```

```
Day    1        3       5
```

In [71]:
```
print('Day 2\t3\t5')
```

```
Day 2   3       5
```

In [72]:
```
print('Day 3\t3\t5')
```

```
Day 3   3       5
```

In [73]:
```
print('Day 4\t3\t5')
```

```
Day 4   3       5
```

In [80]:
```
print('Day 4\t3\t5')
```

```
Day 4   3       5
```

In [ ]:

In [92]:
```
print('This is a black slash symbol(\\)')     # To write a black slash
```

```
This is a black slash symbol(\)
```

In [93]:
```
print('In every programming language it starts with \"Hello, world!\"')
```

```
In every programming language it starts with "Hello, world!"
```

In [ ]:

In [99]:
```
print(' \"hi" ')
```

```
 "hi"
```

In [ ]:

String methods

In [103…
```
# capitalize() = converts the first character in the string to upper case

challenge ='thirty days'
print(challenge.capitalize())
```

```
Thirty days
```

In [ ]:

In [ ]:
```
# Count() = returns occurences of substring in a string, count(substring, start=..,
```

In [104…
```
challenge='thirty days of python'
```

In [105…
```
print(challenge.count('y'))
```

```
3
```

In [108…
```
print(challenge.count('y',7,14))   # y in between 7th and 14th index
```

```
1
```

In [110…
```
print(challenge.count('th'))
```

```
2
```

In [ ]:
```

```

In [111…
```
# endswith() = checks if a string ends with a specfied ending
```

In [113…
```
print(challenge.endswith('th'))
```

```
False
```

In [114…
```
print(challenge.endswith('on'))
```

```
True
```

In [ ]:
```

```

In [ ]:
```
# expandtabs() = replaces tab character with spaces, default tab size is 8
```

In [119…
```
challenge= 'Thirty\tdays\tof\tython'
print(challenge)
```

```
Thirty  days    of      ython
```

In [121…
```
print(challenge.expandtabs())       # changing tab to space
```

```
Thirty  days    of      ython
```

In [122…
```
print(challenge.expandtabs(10))     # changing 8 size tab to 10 size tab
```

```
Thirty    days      of        ython
```

In [ ]:
```

```

In [123…
```
# find() = returns the index of first occurence of substring
```

In [125…
```
print(challenge.find('y'))      # 5th index first 'y'
```

```
5
```

In [127…
```
print(challenge.find('th'))
```

```
16
```

In [128…
```
print(challenge.find('Th'))
```

```
0
```

In [ ]:
```

```

In [129…
```
# format() = formats string into nicer output
```

In [130…
```
first_name='Asabeneh'
```

In [131…
```
last_name='yetayeh'
```

In [132…
```
job='teacher'
```

In [133…
```
country='finland'
```

In [136…
```
sentence= ' I am {} {}. I am a {}.I live in{}.'.format(first_name,last_name,job,cou
```

In [137…
```
print(sentence)
```

```
I am Asabeneh yetayeh. I am a teacher.I live infinland.
```

In [ ]:

In [138…
```
rad=10
pi=3.14
area=pi*rad**2
result = 'The area of circle with radius {} is {}'.format(rad,area)
```

In [139…
```
print(result)
```

```
The area of circle with radius 10 is 314.0
```

In [ ]:

In [140…
```
# index() = returns the index of substring
```

In [141…
```
challenge = 'Thirty days of pyhton'
```

In [142…
```
print(challenge.index('y'))
```

```
5
```

In [144…
```
print(challenge.index('ht'))
```

```
17
```

In [ ]:

In [145…
```
# isalnum() = checks if all are alphanumeric characters
```

In [146…
```
print(challenge.isalnum())   # False if the string contains any non-alphanumeric ch
```

```
False
```

In [148…
```
challenge='30daysofpython'     # True if all characters in the string are alphanume
print(challenge.isalnum())
```

```
True
```

In [149…
```
challenge='30 days of python'
print(challenge.isalnum())
```

```
False
```

In [ ]:

In [150…
```
# isalpha() = checks if all characters are alphabets
```

In [151…
```
challenge='30 days of python'
print(challenge.isalnum())
```

```
False
```

In [154…
```
challenge='30589'
print(challenge.isalnum())
```

```
True
```

In [ ]:

In [ ]:
```
# isdecimal() = checks if all characters are decimal characters
```

In [160…
```
num='30'
```

In [161…
```
print(num.isdecimal())
```

```
True
```

In [162…
```
num1='30kg'
```

In [163…
```
print(num1.isdecimal())
```

```
False
```

In [164…
```
num2='30.56'
```

In [166…
```
print(num2.isdecimal())    # . for fraction/floats are considered as non decimal
```

```
False
```

In [ ]:

In [167…
```
# isdigit() = checks for digits
```

In [168…
```
num1='156'
```

In [169…
```
print(num1.isdigit())
```

```
True
```

In [170…
```
num3='15.369'
```

In [171…
```
print(num.isdigit())      # . for fractions/floats are digitd
```

```
True
```

In [ ]:

In [172…
```
# isidentifier() = checks for valid identifier i.e., it checks if a string is suita
```

In [173…
```
str = '30daysofchallenge'
```

In [175…
```
print(str.isidentifier())    # starting with number
```

```
False
```

In [176…
```
str1 = 'days of'
```

In [177…
```
print(str1.isidentifier())    # has spcae
```

```
False
```

In [178…
```
str2='daysofmine'
```

In [181…
```
print(str2.isidentifier())    # can be used as variable name
```

```
True
```

In [ ]:
```

```

In [182…
```
# isupper() = checks if all alphabets in a string are uppercase
```

In [183…
```
str1=' thirty days'
```

In [184…
```
print(str1.isupper())
```

```
False
```

In [185…
```
str2='THIRTY DAYS'
```

In [186…
```
print(str2.isupper())    # spaces are neglected
```

```
True
```

In [ ]:
```

```

In [187…
```
# islower() = checks if all alphabets are in lower case
```

In [ ]:
```
str1=' thirty days'
```

In [188…
```
print(str1.islower())
```

```
True
```

In [190…
```
str2='THIRTy dAYS'
```

In [192…
```
print(str2.islower())
```

```
False
```

In [ ]:
```

```

In [193…
```
# isnumeric() = checks for numeric characters
```

In [194…
```
num='10'
```

In [195…
```
print(num.isnumeric())
```

```
True
```

In [196…
```
print('ten'.isnumeric())
```

```
False
```

In [ ]:
```

```

In [197…
```
# join() = returns a concatenated string
```

In [200…
```
web_tech = ['html','css','java','react']
```

In [201…
```
result= '#,'.join(web_tech)       # links substrings together
```

In [202…
```
print(result)
```

```
html#,css#,java#,react
```

In [ ]:
```

```

In [203…
```
# strip() = removes whitespace characters (spaces, tabs, newlines) from both leadin
```

In [221…
```
challenge = 'thirty days of python '
```

In [222…
```
print(challenge.strip())
```

```
thirty days of python
```

In [ ]:
```

```

In [223…
```
# replace() = replace substring
```

In [225…
```
print(challenge.replace('python','vscode'))
```

```
thirty days of vscode
```

In [ ]:
```

```

In [226…
```
# split() =splits string from left
```

In [227…
```
print(challenge.split())
```

```
['thirty', 'days', 'of', 'python']
```

In [ ]:
```

```

In [228…
```
# title = returns a title case string
```

In [230…
```
print(challenge.title())   # first letter of every word is capitalised
```
```
Thirty Days Of Python
```

In [ ]:
```

```

In [235…
```
# swapcase() = swaps the case of every letter in the string
```

In [232…
```
challenge
```

Out[232…
```
'thirty days of python '
```

In [239…
```
challenge1= 'Thirty Days Of Python'
```

In [240…
```
print(challenge.swapcase())
```
```
THIRTY DAYS OF PYTHON
```

In [241…
```
print(challenge1.swapcase())
```
```
tHIRTY dAYS oF pYTHON
```

In [ ]:
```

```

In [242…
```
# startswith() = checks if string starts with a specified substring
```

In [243…
```
print(challenge.startswith('of'))
```
```
False
```

In [244…
```
print(challenge.startswith('thirty'))
```
```
True
```

In [ ]:
```

```