Single line comment

In [3]:
```
letter = 's'
print(letter)
```

```
s
```

In [4]:
```
print(len(letter))
```

```
1
```

In [9]:
```
greeting = 'Hello, World!'
print(greeting)
print(len(greeting))
```

```
Hello, World!
13
```

In [10]:
```
sentence = "I hope you are enjoying 30 days of python challenge"
print(sentence)
```

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

Multiline String

In [12]:
```
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)
```

```
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 [13]:
```
# 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)
```

```
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.
```

String Concatenation

In [14]:
```
first_name = 'Sidra'
last_name = 'Raheem'
space = ' '
full_name = first_name + space + last_name
print(full_name) # Sidra Raheem
```

```
Sidra Raheem
```

In [83]:
```
# Checking length of a string using len() builtin function
print(len(first_name))
print(len(last_name))
```

```
print(len(first_name) > len(last_name))
print(len(full_name))
```

```
5
6
False
12
```

Unpacking characters

In [84]:
```
language = 'Python'
a,b,c,d,e,f = language
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

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

Accessing characters in strings by index

In [19]:
```
name = 'Sidra'
first_letter = name[0]
print(first_letter) # s
```

```
S
```

In [30]:
```
second_letter = name[1]
print(second_letter) # i
```

```
i
```

In [31]:
```
last_index = len(name) - 1
last_letter = name[last_index]
print(last_letter) # a
```

```
a
```

In [25]:
```
# If we want to start from right end we can use negative indexing. -1 is the last i
name = 'Sidra'
last_letter = name[-1]
print(last_letter) # a
second_last = name[-2]
print(second_last) # r
```

```
a
r
```

Slicing

In [32]:
```
language = 'Python'
last_letter =language[0:3] # starts at zero index and up to 3 but not include 3
```

```
last_three = language[3:6]
print(last_three)
```

```
hon
```

In [37]:
```
# Another way
last_three = language[-3]
print(last_three)   # hon
```

```
h
```

In [36]:
```
last_three = language[3:]
print(last_three)
```

```
hon
```

In [38]:
```
# Skipping character while splitting Python strings
language = 'Python'
pto = language[0:6:2] #
print(pto) # pto
```

```
Pto
```

Escape sequence

In [40]:
```
print('I hope every one enjoying the python challenge.\nDo you ?') # line break
```

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

In [41]:
```
print('Days\tTopics\tExercises')
```

```
Days    Topics  Exercises
```

In [42]:
```
print('Day 1\t3\t5')
```

```
Day 1   3       5
```

In [43]:
```
print('Day 2\t3\t5')
```

```
Day 2   3       5
```

In [44]:
```
print('Day 3\t3\t5')
```

```
Day 3   3       5
```

In [45]:
```
print('Day 4\t3\t5')
```

```
Day 4   3       5
```

In [46]:
```
print('This is a back slash  symbol (\\)') # To write a back slash
```

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

In [47]:
```
print('In every programming language it starts with \"Hello, World!\"')
```

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

String Methods

capitalize():

In [48]:
```
challenge = 'thirty days of python'
print(challenge.capitalize()) # 'Thirty days of python'
```

```
Thirty days of python
```

count():

In [49]:
```
challenge = 'thirty days of python'
print(challenge.count('y')) # 3
print(challenge.count('y', 7, 14)) # 1
print(challenge.count('th')) # 2`
```

```
3
1
2
```

endswith():

In [50]:
```
challenge = 'thirty days of python'
print(challenge.endswith('on'))
print(challenge.endswith('tion'))
```

```
True
False
```

expandtabs():

In [51]:
```
challenge = 'thirty\tdays\tof\tpython'
print(challenge.expandtabs())
print(challenge.expandtabs(10))
```

```
thirty  days    of      python
thirty    days      of        python
```

find():

In [52]:
```
challenge = 'thirty days of python'
print(challenge.find('y'))
print(challenge.find('th'))
```

```
5
0
```

format()

In [54]:
```
first_name = 'Sidra'
last_name = 'Raheem'
job = 'teacher'
country = 'Finland'
sentence = 'I am {} {}. I am a {}. I live in {}.'.format(first_name, last_name, job
print(sentence)
```

```
I am Sidra Raheem. I am a teacher. I live in Finland.
```

In [55]:
```
radius = 10
pi = 3.14
area = pi
result = 'The area of circle with {} is {}'.format(str(radius), str(area))
print(result)
```

```
The area of circle with 10 is 3.14
```

index():

In [56]:
```
challenge = 'thirty days of python'
print(challenge.find('y'))
print(challenge.find('th'))
```

```
5
0
```

isalnum():

In [57]:
```
challenge = 'ThirtyDaysPython'
print(challenge.isalnum())
```

```
True
```

In [58]:
```
challenge = '30DaysPython'
print(challenge.isalnum())
```

```
True
```

In [59]:
```
challenge = 'thirty days of python'
print(challenge.isalnum())
```

```
False
```

In [60]:
```
challenge = 'thirty days of python 2019'
print(challenge.isalnum())
```

```
False
```

isalpha():

In [61]:
```
challenge = 'thirty days of python'
print(challenge.isalpha())
```

```
False
```

In [62]:
```
num = '123'
print(num.isalpha())
```

```
False
```

isdecimal():

In [63]:
```
challenge = 'thirty days of python'
print(challenge.find('y'))
print(challenge.find('th'))
```

```
5
0
```

isdigit():

In [64]:
```
challenge = 'Thirty'
print(challenge.isdigit())
```

```
False
```

In [67]:
```
challenge = '30'
print(challenge.isdigit())
```

```
True
```

isdecimal():

In [68]:
```
num = '10'
print(num.isdecimal())
```

```
True
```

In [69]:
```
num = '10.5'
print(num.isdecimal())
```

```
False
```

isidentifier():

In [70]:
```
challenge = '30DaysOfPython'
print(challenge.isidentifier())
challenge = 'thirty_days_of_python'
print(challenge.isidentifier())
```

```
False
True
```

islower():

In [71]:
```
challenge = 'thirty days of python'
print(challenge.islower())
challenge = 'Thirty days of python'
print(challenge.islower())
```

```
True
False
```

isupper():

In [72]:
```
challenge = 'thirty days of python'
print(challenge.isupper())
challenge = 'THIRTY DAYS OF PYTHON'
print(challenge.isupper())
```

```
False
True
```

isnumeric():

In [73]:
```
num = '10'
print(num.isnumeric())
print('ten'.isnumeric())
```

```
True
False
```

join():

In [74]:
```
web_tech = ['HTML', 'CSS', 'JavaScript', 'React']
result = '#, '.join(web_tech)
print(result)
```

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

strip():

In [76]:
```
challenge = ' thirty days of python '
print(challenge.strip('y'))
```

```
 thirty days of python
```

replace():

In [77]:
```
challenge = 'thirty days of python'
print(challenge.replace('python', 'coding'))
```

```
thirty days of coding
```

split():

In [78]:
```
challenge = 'thirty days of python'
print(challenge.split()) # ['thirty', 'days', 'of', 'python']
```

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

title():

In [79]:
```
challenge = 'thirty days of python'
print(challenge.title())
```

```
Thirty Days Of Python
```

swapcase():

In [81]:
```
challenge = 'thirty days of python'
print(challenge.swapcase())
challenge = 'Thirty Days Of Python'
print(challenge.swapcase())
```

```
THIRTY DAYS OF PYTHON
tHIRTY dAYS oF pYTHON
```

startswith():

In [82]:
```
challenge = 'thirty days of python'
print(challenge.startswith('thirty'))
challenge = '30 days of python'
print(challenge.startswith('thirty'))
```

```
True
False
```