In [5]:
```
# Single line comment
letter ='P'                 # A string could be a single character or a bunch c
print(letter)               # P
print(len(letter))          # 1
greeting = 'Hello, World!' #String could be a single or double quote, "Hello W
print(greeting)
print(len(greeting))
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
```

In [6]:
```
# Multiline String
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 [7]:
```
# 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.
```

In [10]:
```
# String Concatenation
first_name = 'Asabeneh'
last_name = 'Yetayeh'
space = ' '
full_name = first_name + space + last_name
print(full_name)
```

```
Asabeneh Yetayeh
```

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

```
8
7
True
16
```

In [14]:
```
### Unpacking characters
language = 'Python'
a,b,c,d,e,f = language # unpacking sequence characters into variables
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
```

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

In [21]:
```
# Accessing characters in strings by index
language = 'Python'
first_letter = language[0]
print(first_letter)   # P
second_letter = language[1]
print(second_letter)
last_index = len(language) -1
last_letter = language[last_index]
print(last_letter)
```

```
P
y
n
```

In [23]:
```
# If we want to start from right end we can use negative indexing. -1 is the l
language = 'Python'
last_letter = language[-1]
print(last_letter)
second_last = language[-2]
print(second_last)
```

```
n
o
```

In [26]:
```
# Slicing
language = 'Python'
first_three = language[0:3]     # starts at zero index and up to 3 but not inc
last_three = language[3:6]
print(last_three)
# Another way
last_three = language[-3:]
print(last_three)
last_three = language[3:]
print(last_three)
```

```
hon
hon
hon
```

In [27]:
```
# Skipping character while splitting Python strings
language = 'Python'
pto = language[0:6:2] #
print(pto) # pto
```

```
Pto
```

In [32]:
```
# Escape sequence
print('I hope every one enjoying the python challenge.\nDo you ?') # line brea
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 back slash  symbol (\\)') # To write a back slash
print('In every programming language it starts with \"Hello, World!\"')
```

```
I hope every one enjoying the python challenge.
Do you ?
Days    Topics  Exercises
Day 1   3       5
Day 2   3       5
Day 3   3       5
Day 4   3       5
This is a back slash  symbol (\)
In every programming language it starts with "Hello, World!"
```

String Method

In [2]:
```
# capitalize(): converts the first character the string to capital letter

challenge = 'thirty days of python'
print(challenge.capitalize())           #'Thirty days of python'
```

```
Thirty days of python
```

In [5]:
```
# count(): returns occurrences of substring in string , count(substring, start

challenge = 'thirty days of python'
print(challenge.count('y'))
print(challenge.count('y',7,14))
print(challenge.count('th'))
```

```
3
1
2
```

In [7]:
```
# 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
```

```
True
False
```

In [4]:
```
#expandtabs(): Replace tab character with spaces, default tab size is 8

challenge = 'thirty\tdays\tof\tpython'
print(challenge.expandtabs())
print(challenge.expandtabs(10))
```

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

In [6]:
```
# Find(): Returns the index of first occurrence of substring

challenge =  'thirty days of python '
print(challenge.find('y'))
print(challenge.find('th'))
```

```
5
0
```

In [9]:
```
# format()  formats string into nicer output
first_name = 'Prachi'
last_name = 'Singare'
job = 'Data Scientist'
country = 'India'
sentence = 'I am {} {}. I am a {}.'.format(first_name, last_name, job, country
print(sentence)
```

```
I am Prachi Singare. I am a Data Scientist.
```

In [18]:
```
radius = 10
pi = 3.14
area = pi * radius ** 2
result = 'The area of circle with {} is {}'.format(str(radius), str(area))
print(result)
```

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

In [20]:
```
# index(): Returns the index of substring
challenge = 'thirty days of python'
print(challenge.find('y'))
print(challenge.find('th'))
```

```
5
0
```

In [3]:
```
# isalnum(): checks alphanumeric character

challenge = 'ThirtyDaysPython'
print(challenge.isalnum())       # True

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

challenge = 'Thirty days of python'
print(challenge.isalnum())

challenge = 'Thirty days of python 2019'
print(challenge.isalnum())
```

```
True
True
False
False
```

In [7]:
```
# isalpha(): Checks if all characters are alphabets

challenge = 'thirty days of python'
print(challenge.isalpha()) # True

num = '123'
print(num.isalpha())      # False

challenge = 'thirtydaysofpython'
print(challenge.isalpha()) # True
```

```
False
False
True
```

In [8]:
```
# isdecimal(): Checks Decimal Characters
num = '10'
print(num.isdecimal()) # True
num = '10.5'
print(num.isdecimal()) # False
```

```
True
False
```

In [3]:
```
# isdigit(): Checks digit characters
challenge = 'Thirty'
print(challenge.isdigit())   # False
challenge = '30'
print(challenge.isdigit())
```

```
False
True
```

In [5]:
```
# isidentifier(): checks for valid identifier means it check if a string is a

challenge = '30DaysofPython'
print(challenge.isidentifier())   # False , because it starts with a number
challenge = 'thirty_days_of_python'
print(challenge.isidentifier())
```

```
False
True
```

In [8]:
```
# islower():Checks if all alphabets in a string are lowercase

challenge = 'thirty days of python'
print(challenge.islower())
challenge = 'Thirty days of python'
print(challenge.islower())
```

```
True
False
```

In [9]:
```
# isupper(): returns if all characters are uppercase characters

challenge = 'thirty days of python'
print(challenge.isupper()) #  False
challenge = 'THIRTY DAYS OF PYTHON'
print(challenge.isupper()) # True
```

```
False
True
```

In [12]:
```
# isnumeric():Checks numeric characters

num = '10'
print(num.isnumeric())      # True
print('ten'.isnumeric())    # False
```

```
True
False
```

In [13]:
```
# join(): Returns a concatenated string

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

```
HTML#,CSS#,JAVASCRIPT#,REACT
```

In [15]:
```
# strip(): Removes both leading and trailing characters

challenge = 'thirty days of python'
print(challenge.strip('y')) # 5
```

```
thirty days of python
```

In [17]:
```
# split():Splits String from Left

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

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

In [16]:
```
# title(): Returns a Title Cased String

challenge = 'thirty days of python'
print(challenge.title()) # Thirty Days Of Python
```

```
Thirty Days Of Python
```

In [18]:
```
# swapcase(): Checks if String Starts with the Specified String

challenge = 'thirty days of python'
print(challenge.swapcase())   # THIRTY DAYS OF PYTHON
challenge = 'Thirty Days Of Python'
print(challenge.swapcase())  # tHIRTY dAYS oF pYTHON
```

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

In [19]:
```
# 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
```

```
True
False
```