In [1]:

```
# Single line comment
letter = 'P'                # A string could be a single character or a bunch of texts
```

In [3]:

```
print(letter)
```

```
P
```

In [5]:

```
print(len(letter))
```

```
1
```

In [7]:

```
grettings = 'Hello World!'
```

In [9]:

```
print(grettings)
```

```
Hello World!
```

In [13]:

```
print(len(grettings))
```

```
12
```

In [15]:

```
sentence = 'I am enjoying 30 days of python challenge'
print(sentence)
```

```
I am enjoying 30 days of python challenge
```

In [17]:

```
# 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 [19]:

```
# 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 [27]:

```
#String concantenation
first_name = 'Ratnaprava'
last_name = 'Mohapatra'
space = ' '
full_name = first_name + space + last_name
```

In [29]:

```
print(full_name)
```

```
Ratnaprava Mohapatra
```

In [31]:

```
#checking the length of string using len() inbuilt function
print(len(first_name))
```

```
10
```

In [33]:

```
print(len(last_name))
```

```
9
```

In [35]:

```
print(len(full_name))
```

```
20
```

In [43]:

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

```
True
```

Unpacking Characters

In [46]:

```
language = 'Python'
a, b, c, d, e, f = language # unpacking sequence to characters into variables
```

In [48]:

```
print(a)
```

```
P
```

In [50]:

```
print(b)
```

```
y
```

In [52]:

```
print(c)
```

```
t
```

In [54]:

```
print(d)
```

```
h
```

In [56]:

```
print(e)
```

```
o
```

In [58]:

```
print(f)
```

```
n
```

In [62]:

```
# Acessing  chracters in strings by index
language = 'python'
first_letter  = language[0]
print(first_letter)
```

```
p
```

In [64]:

```
second_letter = language[1]
print(second_letter)
```

```
y
```

In [72]:

```
last_index = len(language) -1
last_letter = language[last_index]
print(last_letter)
```

```
n
```

In [74]:

```
# 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
```

In [76]:

```
second_letter = language[-2]
print(second_letter)
```

```
o
```

Slicing

In [79]:

```
language = 'Python'
first_three = language[0:3]  # start index at 0 at ends with before 3 not include 3
print(first_three)
```

```
Pyt
```

In [81]:

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

```
hon
```

In [89]:

```
# another Way
last_three = language[-3:]
print(last_three)
```

```
hon
```

In [91]:

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

```
hon
```

In [95]:

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

```
pto
```

In [97]:

```
# Escape Sequence
print('I am enjoying python challenge. \n Do you ?.')  #line break
```

```
I am enjoying python challenge.
 Do you ?.
```

In [99]:

```
print('Days\tTopics\tExercise')
```

```
Days    Topics  Exercise
```

In [101…

```python
print('Day 1\t3\t5')
```

```
Day 1	3	5
```

In [103…

```python
print('Day 2\t3\t5')
```

```
Day 2	3	5
```

In [105…

```python
print('Day 3\t3\t5')
```

```
Day 3	3	5
```

In [107…

```python
print('Day 4\t3\t5')
```

```
Day 4	3	5
```

In [109…

```python
print('This is a back slash  symbol (\\)') # To write a back slash
print('In every programming language it starts with \"Hello, World!\"')
```

```
This is a back slash  symbol (\)
In every programming language it starts with "Hello, World!"
```

In [115…

```python
## String Methods
# capitalize(): Converts the first character the string to Capital Letter
challenge = 'thirty days of python'
print(challenge.capitalize())
```

```
Thirty days of python
```

In [117…

```python
# count(): returns occurrences of substring in string, count(substring, start=.., end=..)

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

```
3
```

In [121…

```python
print(challenge.count('y',7,14))
```

```
1
```

In [123…

```python
print(challenge.count('th'))
```

```
2
```

In [125…

```python
# endswith(): Checks if a string ends with a specified ending
print(challenge.endswith('on'))
```

```
True
```

In [127…

```python
print(challenge.endswith('tion'))
```

```
False
```

In [137…

```python
# 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
```

In [139…

```python
print(challenge.expandtabs(10))
```

```
thirty    days      of        python
```

In [145…

```python
# find(): Returns the index of first occurrence of substring
print(challenge.find('y'))
```

```
5
```

In [147…

```python
print(challenge.find('th'))
```

```
0
```

In [149…

```python
# format()      formats string into nicer output
first_name = 'Ratnaprava'
last_name = 'Mohapatra'
job = 'Data scientist'
Country = 'India'
sentence = 'I am {} {}. I am a {}.I live in {}'.format(first_name,last_name,job,Country)
print(sentence)
```

```
I am Ratnaprava Mohapatra. I am a Data scientist.I live in India
```

In [161…

```python
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 3.14
```

In [163…

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

```
5
```

In [165…

```
print(challenge.find('r'))
```

```
3
```

In [167…

```
# isalnum(): Checks alphanumeric character

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

```
True
```

In [169…

```
challenge = '30daysofpython'
print(challenge.isalnum())
```

```
True
```

In [171…

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

```
False
```

In [175…

```
# isalpha(): Checks if all characters are alphabets
challenge = 'thirtydaysofpython'
print(challenge.isalpha())
```

```
True
```

In [177…

```
challenge = '234'
print(challenge.isalpha())
```

```
False
```

In [181…

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

```
5
```

In [183…

```
# isdigit(): checks if all characters are digit
challange = 'thirty'
print(challenge.isdigit())
```

```
False
```

In [191…

```
challange = '567'
print(challange.isdigit())
```

```
True
```

In [193…

```
# isdecimal():Checks decimal characters
challenge = '67.90'
print(challenge.isdecimal())
```

```
False
```

In [195…

```
challenge = '135679'
print(challenge.isdecimal())
```

```
True
```

In [197…

```
# isidentifier():Checks for valid identifier means it check if a string is a valid variable name
challenge = '30daysofpython'
print(challenge.isidentifier())
```

```
False
```

In [199…

```
challenge = 'thirty_days'
print(challenge.isidentifier())
```

```
True
```

In [201…

```
# islower():Checks if all alphabets in a string are lowercase
challenge = 'thirty_days'
print(challenge.islower())
```

```
True
```

In [209…

```
challenge = 'Thirty_days'
print(challenge.islower())
```

```
False
```

In [211…

```
# isupper(): returns if all characters are uppercase characters

challenge = 'thirty days of python'
```

```
print(challenge.isupper()) #  False
```

```
False
```

In [213…

```
challenge = 'THIRTY DAYS OF PYTHON'
print(challenge.isupper()) # True
```

```
True
```

In [215…

```
# isnumeric():Checks numeric characters

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

```
True
```

In [217…

```
num = '0.90'
print(num.isnumeric())
```

```
False
```

In [219…

```
print('ten'.isnumeric())
```

```
False
```

In [225…

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

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

In [231…

```
# strip(): Removes both leading and trailing characters
challenge = ' thirty days of python '
print(challenge.strip('y'))
```

```
 thirty days of python
```

In [233…

```
# replace(): Replaces substring inside
challenge = 'thirty days of challenge python '
print(challenge.replace('python', 'coding'))
```

```
thirty days of challenge coding
```

In [235…

```
# split():Splits String from Left
challenge = 'thirty days of challenge python'
print(challenge.split())
```

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

In [237…

```
# title(): Returns a Title Cased String
challenge = 'thirty days of challenge python'
print(challenge.title())
```

```
Thirty Days Of Challenge Python
```

In [241…

```
# swapcase(): Checks if String Starts with the Specified String
challenge = 'thirty days of challenge python'
print(challenge.swapcase())
```

```
THIRTY DAYS OF CHALLENGE PYTHON
```

In [245…

```
challenge = 'ThirTy dAyS oF challeNge pythoN'
print(challenge.swapcase())
```

```
tHIRtY DaYs Of CHALLEnGE PYTHOn
```

In [249…

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

```
True
```

In [251…

```
print(challenge.startswith('python'))
```

```
False
```

In [ ]:

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js