Strings

In [4]:
```
letter='J'
print(letter)
```

```
J
```

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

```
1
```

In [6]:
```
greeting = "Hello World"
print(greeting)
```

```
Hello World
```

In [7]:
```
print(len(greeting))
```

```
11
```

In [1]:
```
sentence = "I hope you are enjoying Nit "
print(sentence)
```

```
I hope you are enjoying Nit
```

Single line comment

In [4]:
```
# Single line comment
letter = 'P'                # A string could be a single character or a bunch
print(letter)               # P
print(len(letter))          # 1
greeting = 'Hello World'  # String could be  a single or double quote,"Hello,
print(greeting)             # Hello, World!
print(len(greeting))        # 13
sentence = "I hope you are enjoying Nit"
print(sentence)
```

```
P
1
Hello World
11
I hope you are enjoying Nit
```

In [6]:
```
# Multiline String
multiline_string = '''I am a Data science student and enjoy kaggle dataset.
That is why I created project.'''
print(multiline_string)
```

```
I am a Data science student and enjoy kaggle dataset.
That is why I created project.
```

In [7]:
```
# Multiline String
multiline_string = """I am a Data science student and enjoy kaggle dataset.
That is why I created project."""
print(multiline_string)
```

```
I am a Data science student and enjoy kaggle dataset.
That is why I created project.
```

In [8]:
```
# String Concatenation
first_name = 'Janhavi'
last_name = 'Landge'
space = ' '
full_name = first_name  +  space + last_name
print(full_name)
```

```
Janhavi Landge
```

In [9]:
```
print(len(first_name))
print(len(last_name))
print(len(first_name) > len(last_name))
print(len(full_name))
```

```
7
6
True
14
```

In [10]:
```
#Unpacking Character
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
```

In [11]:
```
# Accessing characters in strings by index
language = 'Python'
first_letter = language[0]
print(first_letter)
second_letter = language[1]
print(second_letter)
last_index = len(language) - 1
last_letter = language[last_index]
print(last_letter)
```

```
P
y
n
```

In [12]:
```
#Unpacking Character
language = 'DataScience'
a,b,c,d,e,f,g,h,i,j,k = language
print(a)
print(b)
print(c)
print(d)
print(e)
print(f)
print(g)
print(h)
print(i)
print(j)
print(k)
```

```
D
a
t
a
S
c
i
e
n
c
e
```

In [13]:
```
# Accessing characters in strings by index
language = 'DataScience'
first_letter = language[0]
print(first_letter)
second_letter = language[1]
print(second_letter)
last_index = len(language) - 1
last_letter = language[last_index]
print(last_letter)
```

```
D
a
e
```

In [14]:
```
#Slicing
language = 'DataScience'
first_three = language[0:3] # starts at zero index and up to 3 but not include
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
```

```
aSc
nce
aScience
```

In [15]:
```
language = 'DataScience'
pto = language[0:6:2] #
print(pto) # pto
```

```
DtS
```

In [16]:
```
#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 [17]:
```
# capitalize(): Converts the first character the string to Capital Letter
```

In [19]:
```
challenge = '6 month of datascience'
print(challenge.capitalize())
```

```
6 month of datascience
```

In [23]:
```
# count(): returns occurrences of substring in string, count(substring, start=
```

In [21]:
```
challenge = 'thirty days of python'
print(challenge.count('y'))
print(challenge.count('y', 7, 14))
print(challenge.count('th'))
```

```
3
1
2
```

In [24]:
```
# endswith(): Checks if a string ends with a specified ending
```

In [22]:
```
challenge = '6 month of datascience'
print(challenge.endswith('ce'))   # True
print(challenge.endswith('tion')) # False
```

```
True
False
```

In [26]:
```
 #expandtabs(): Replaces tab character with spaces, default tab size is 8.
```

In [27]:
```
challenge = '6\tmonths\tof\tdatascience'
print(challenge.expandtabs())
print(challenge.expandtabs(10))
```

```
6       months  of      datascience
6         months    of        datascience
```

In [28]:
```
challenge = '6 month of datascience'
print(challenge.find('h'))
print(challenge.find('en'))
```

```
6
18
```

In [29]:
```
# format()  formats string into nicer output
first_name = 'Janhavi'
last_name = 'Landge'
job = 'DataScience'
country = 'India'
sentence = 'I am {} {}. I am a {}. I live in {}.'.format(first_name, last_name
print(sentence)
```

```
I am Janhavi Landge. I am a DataScience. I live in India.
```

In [30]:
```
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
```

In [31]:
```
# index(): Returns the index of substring
```

In [32]:
```
challenge = '6 month of datascience'
print(challenge.find('h'))
print(challenge.find('ta'))
```

```
6
13
```

In [33]:
```
# isalnum(): Checks alphanumeric character
```

In [34]:
```
challenge = '6monthofdatascience'
print(challenge.isalnum())
```

```
True
```

In [35]:
```
challenge = '6monthdatascience'
print(challenge.isalnum())
```

```
True
```

In [36]:
```
challenge = '6 month of datascience'
print(challenge.isalnum())
```

```
False
```

In [37]:
```
challenge = '6 month of datascience 2019'
print(challenge.isalnum()) # False
```

```
False
```

In [38]:
```
## isalpha(): Checks if all characters are alphabets
```

In [39]:
```
challenge = '6 month of datascience'
print(challenge.isalpha()) # True
num = '123'
print(num.isalpha())
```

```
False
False
```

In [41]:
```
#isdecimal(): Checks Decimal Characters
```

In [42]:
```
challenge = '6 month of datascience'
print(challenge.find('o'))  # 5
print(challenge.find('sc')) # 0


```

```
3
15
```

In [43]:
```
# isdigit(): Checks Digit Characters
```

In [47]:
```
# isdecimal():Checks decimal characters

num = '10'
print(num.isdecimal()) # True
num = '10.5'
print(num.isdecimal()) # False
```

```
True
False
```

In [48]:
```
# isidentifier():Checks for valid identifier means it check if a string is a v

challenge = '3'
print(challenge.isidentifier()) # False, because it starts with a number
challenge = 'thirty_days_of_python'
print(challenge.isidentifier()) # True
```

```
False
True
```

In [49]:
```
# islower():Checks if all alphabets in a string are lowercase

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

```
True
False
```

In [1]:
```
# isupper(): returns if all characters are uppercase characters

challenge = '6 months of Datascience'
print(challenge.isupper()) #  False
challenge = '6 MONTH OF DATASCIENCE'
print(challenge.isupper()) # True
```

```
False
True
```

In [2]:
```
# isnumeric():Checks numeric characters

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

```
True
False
```

In [3]:
```
# join(): Returns a concatenated string

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

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

In [4]:
```
# strip(): Removes both leading and trailing characters
```

In [5]:
```
challenge = ' 6 months of datascience '
print(challenge.strip('t'))
```

```
 6 months of datascience
```

In [6]:
```
#replace(): Replaces substring inside
```

In [7]:
```
challenge = '6 moths of datascience'
print(challenge.replace('datascience', 'coding'))
```

```
6 moths of coding
```

In [8]:
```
# split():Splits String from Left
```

In [9]:
```
challenge = '6 months of datascience'
print(challenge.split())
```

```
['6', 'months', 'of', 'datascience']
```

In [10]:
```
# title(): Returns a Title Cased String

```

In [11]:
```
challenge = '6 months of datascience'
print(challenge.title())
```

```
6 Months Of Datascience
```

In [12]:
```
# swapcase(): Checks if String Starts with the Specified String
```

In [15]:
```
challenge = '6 months of datascience'
print(challenge.swapcase())
challenge = '6 Months of Datascience'
print(challenge.swapcase())
```

```
6 MONTHS OF DATASCIENCE
6 mONTHS OF dATASCIENCE
```

In [16]:
```
# startswith(): Checks if String Starts with the Specified String
```

In [18]:
```
challenge = 'six month of datascience'
print(challenge.startswith('six')) # True
challenge = '6 month of datascience'
print(challenge.startswith('six')) # False
```

```
True
False
```

In [ ]:
```

```