Basic Code.py

In [6]:
```
# Introduction
# Day 1 - 30DaysOfPython Challenge
```

In [7]:
```
print(3 + 2)   # addition(+)
print(3 - 2)   # subtraction(-)
print(3 * 2)   # multiplication(*)
print(3 / 2)   # division(/)
print(3 ** 2)  # exponential(**)
print(3 % 2)   # modulus(%)
print(3 // 2)  # Floor division operator(//)
```

```
5
1
6
1.5
9
1
1
```

In [8]:
```
# Checking data types
```

In [9]:
```
print(type(10))                   # Int
print(type(3.14))                 # Float
print(type(1 + 3j))               # Complex
print(type('prakashsenapati'))    # String
print(type([1, 2, 3]))            # List
print(type({'name':'senapati'})) # Dictionary
print(type({9.8, 3.14, 2.7}))     # Set
print(type((9.8, 3.14, 2.7)))     # Tuple
print(type(3 == 3))               # Bool
print(type(3 >= 3))               # Bool
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'str'>
<class 'list'>
<class 'dict'>
<class 'set'>
<class 'tuple'>
<class 'bool'>
<class 'bool'>
```

Variables.py

In [10]:
```python
# Variables in Python

first_name = 'Deepti'
last_name = 'Jaiswar'
country = 'HYD'
city = 'TELENGANA'
age = 40087
is_married = True
skills = [ 'Python','NLP','Flask']
person_info = {
    'firstname':'Asabeneh',
    'lastname':'Yetayeh',
    'country':'Finland',
    'city':'Helsinki'
    }
```

In [11]:
```python
# Printing the values stored in the variables

print('First name:', first_name)
print('First name length:', len(first_name))
print('Last name: ', last_name)
print('Last name length: ', len(last_name))
print('Country: ', country)
print('City: ', city)
print('Age: ', age)
print('Married: ', is_married)
print('Skills: ', skills)
print('Person information: ', person_info)
```

```
First name: Deepti
First name length: 6
Last name:  Jaiswar
Last name length:  7
Country:  HYD
City:  TELENGANA
Age:  40087
Married:  True
Skills:  ['Python', 'NLP', 'Flask']
Person information:  {'firstname': 'Asabeneh', 'lastname': 'Yetayeh', 'countr
y': 'Finland', 'city': 'Helsinki'}
```

In [12]:
```
# Declaring multiple variables in one line

first_name, last_name, country, age, is_married = 'Asabeneh', 'Yetayeh', 'He

print(first_name, last_name, country, age, is_married)
print('First name:', first_name)
print('Last name: ', last_name)
print('Country: ', country)
print('Age: ', age)
print('Married: ', is_married)
```

```
Asabeneh Yetayeh Helsink 250 True
First name: Asabeneh
Last name:  Yetayeh
Country:  Helsink
Age:  250
Married:  True
```

String.py

In [14]:
```
# Single line comment
letter = 'P'                # A string could be a single character or a bunc
print(letter)               # P
print(len(letter))          # 1
greeting = 'Hello, World!'  # String could be  a single or double quote,"Hel
print(greeting)             # Hello, World!
print(len(greeting))        # 13
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 [15]:
```
# 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 [16]:
```
# 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 [17]:
```
# String Concatenation
first_name = 'Asabeneh'
last_name = 'Yetayeh'
space = ' '
full_name = first_name  +  space + last_name
print(full_name) # Asabeneh Yetayeh
```

```
Asabeneh Yetayeh
```

In [18]:
```
# Checking length of a string using len() builtin function
print(len(first_name))  # 8
print(len(last_name))   # 7
print(len(first_name) > len(last_name)) # True
print(len(full_name)) # 15
```

```
8
7
True
16
```

In [19]:
```
#### Unpacking characters
language = 'Python'
a,b,c,d,e,f = language # unpacking sequence characters into variables
print(a) # P
print(b) # y
print(c) # t
print(d) # h
print(e) # o
print(f) # n
```

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

In [20]:
```
# Accessing characters in strings by index
language = 'Python'
first_letter = language[0]
print(first_letter) # P
second_letter = language[1]
print(second_letter) # y
last_index = len(language) - 1
last_letter = language[last_index]
print(last_letter) # n
```

```
P
y
n
```

In [21]:
```
# If we want to start from right end we can use negative indexing. -1 is the
language = 'Python'
last_letter = language[-1]
print(last_letter) # n
second_last = language[-2]
print(second_last) # o
```

```
n
o
```

In [22]:
```
# Slicing

language = 'Python'
first_three = language[0:3] # starts at zero index and up to 3 but not inclu
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
```

```
hon
hon
hon
```

In [24]:
```
# Skipping character while splitting Python strings
language = 'Python'
pto = language[0:6:2] #
print(pto) # pto

# Escape sequence
print('I hope every one enjoying the python challenge.\nDo you ?') # line br
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!\"')
```

```
Pto
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 Methods

In [25]:
```
# 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 [26]:
```
# count(): returns occurrences of substring in string, count(substring, star

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
```

In [27]:
```
# 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 [28]:
```
# expandtabs(): Replaces tab character with spaces, default tab size is 8. I

challenge = 'thirty\tdays\tof\tpython'
print(challenge.expandtabs())   # 'thirty  days    of      python'
print(challenge.expandtabs(10)) # 'thirty    days      of        python'
```

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

In [29]:
```
# find(): Returns the index of first occurrence of substring

challenge = 'thirty days of python'
print(challenge.find('y'))  # 5
print(challenge.find('th')) # 0
```

```
5
0
```

In [30]:
```
# format()  formats string into nicer output
first_name = 'Asabeneh'
last_name = 'Yetayeh'
job = 'teacher'
country = 'Finland'
sentence = 'I am {} {}. I am a {}. I live in {}.'.format(first_name, last_na
print(sentence) # I am Asabeneh Yetayeh. I am a teacher. I live in Finland.
```

```
I am Asabeneh Yetayeh. I am a teacher. I live in Finland.
```

In [31]:
```
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
```

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

In [32]:
```
# index(): Returns the index of substring
challenge = 'thirty days of python'
print(challenge.find('y'))  # 5
print(challenge.find('th')) # 0
```

```
5
0
```

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

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

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

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

challenge = 'thirty days of python 2019'
print(challenge.isalnum()) # False
```

```
True
True
False
False
```

In [36]:
```python
# isalpha(): Checks if all characters are alphabets

challenge = 'thirtydaysofpython'
print(challenge.isalpha()) # True
num = '123'
print(num.isalpha())      # False
```

```
True
False
```

In [42]:
```python
# find(): Returns the index of first occurrence of substring

challenge = 'thirty days of python'
print(challenge.find('y'))  # 5
print(challenge.find('th')) # 0
```

```
5
0
```

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

challenge = 'Thirty'
print(challenge.isdigit()) # False
challenge = '30'
print(challenge.isdigit())   # True
```

```
False
True
```

In [45]:
```
# isdecimal():Checks decimal characters

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

```
True
False
```

In [46]:
```
# 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()) # True
```

```
False
True
```

In [47]:
```
# 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 [48]:
```
# 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 [49]:
```
# isnumeric():Checks numeric characters

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

```
True
False
```

In [50]:
```
# join(): Returns a concatenated string

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

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

In [51]:
```
# strip(): Removes both leading and trailing characters

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

```
 thirty days of python
```

In [53]:
```
# replace(): Replaces substring inside

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

```
thirty days of coding
```

In [54]:
```
# split():Splits String from Left

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

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

In [55]:
```
# title(): Returns a Title Cased String

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

```
Thirty Days Of Python
```

In [56]:
```
# swapcase(): Changes each word case the lowercase to uppercase and uppercas

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 [57]:
```
# 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
```

In [59]:
```
# endwith(): Checks if String end with the Specified String

challenge = 'thirty days of python'
print(challenge.endswith('python')) # True
challenge = '30 days of python'
print(challenge.endswith('thirty')) # False
```

```
True
False
```

In [ ]:
```

```