Assignment 2 - Python

In [1]:
```
# Operators
a = 5
b = 2
print(a+b)    # Addition
print(a-b)    # Subtraction
print(a*b)    # Multiplication
print(a/b)    # Division
print(a//b)   # Modulus
print(a%b)    # Percentage
print(a**b)   # Exponential
```

```
7
3
10
2.5
2
1
25
```

In [5]:
```
# Checking Data Types

print(type(10))       # Int
print(type(3.14))     # Float
print(type(1+3j))     # Complex
print(type("Vihari Nandan"))    # String
print(type[2,3,4,5,"AI"])  # List
print(type({"name":"Data Science"}))  # Dictionary
print(type({1,2,3,4,5}))           # Set
print(type((1,2,3,4,5)))          # Tuple
print(type(3>1))                # Bool
print(type(3<1))                # Bool
```

```
<class 'int'>
<class 'float'>
<class 'complex'>
<class 'str'>
type[2, 3, 4, 5, 'AI']
<class 'dict'>
<class 'set'>
<class 'tuple'>
<class 'bool'>
<class 'bool'>
```

In [6]:
```
# Variables

first_name = 'Vihari'
last_name = 'Nandan'
country = 'India'
city = 'Tadipatri'
age = 32
is_married = True
```

```
skills = ['Networking', 'IoT', 'DS', 'ML', "AI"]
person_info = {
    'first_name': 'Chandan',
    'last_name' : 'Kumar',
    'country' : 'Bangalore',
    'city' : 'Hyderabad'
}

# Printing the values stored in the variables

print("First Name is ", first_name)
print("First Name length is ", first_name)
print("Last Name is ", last_name)
print("Last Name length is ", last_name)
print("Country is ", country)
print("City is ", city)
print("Age is ", age)
print("Married: ", is_married)
print("Skills are ", skills)
print("Personal Information: ", person_info)
```

```
First Name is  Vihari
First Name length is  Vihari
Last Name is  Nandan
Last Name length is  Nandan
Country is  India
City is  Tadipatri
Age is  32
Married:  True
Skills are  ['Networking', 'IoT', 'DS', 'ML', 'AI']
Personal Information:  {'first_name': 'Chandan', 'last_name': 'Kumar', 'country': 'B
angalore', 'city': 'Hyderabad'}
```

In [7]:
```
# Declaring multiple variables in one line

first_name, last_name, country, age, is_married = 'vihari', 'nandan', 'India', 25,

print(first_name, last_name, country, age, is_married)
print("First Name is ", first_name)
print("Last Name is ", last_name)
print("Country is ", country)
print("Age is ", age)
print("Maritial Status is", is_married)
```

```
vihari nandan India 25 True
First Name is  vihari
Last Name is  nandan
Country is  India
Age is  25
Maritial Status is True
```

In [8]:
```
# Single Line Comment

letter = 'P'
print(letter)
print(len(letter))
```

```
greeting = "Hello Vihari"
print(len(greeting))

sentence = "I hope you are enjoying Full stack Data Science class"
print(sentence)
```

```
P
1
12
I hope you are enjoying Full stack Data Science class
```

In [9]:
```
# Multiline String

multiline_string = '''I am working as a IoT Data analytics Engineer.
I find anything IoT Projects.
That is why joined in Ful stack Data Science.'''
print(multiline_string)

# Another wat of doing the same thing
multiline_string = """I am working as a IoT Data analytics Engineer.
I find anything IoT Projects.
That is why joined in Ful stack Data Science."""
print(multiline_string)
```

```
I am working as a IoT Data analytics Engineer.
I find anything IoT Projects.
That is why joined in Ful stack Data Science.
I am working as a IoT Data analytics Engineer.
I find anything IoT Projects.
That is why joined in Ful stack Data Science.
```

In [13]:
```
# String Concatenation

first_name = "Vihari"
last_name = "Nandan"
space = ' '
full_name = first_name + space + last_name
print(full_name)

# 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))
```

```
Vihari Nandan
6
6
False
13
```

In [16]:
```
# Unpacking characters

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 [17]:
```
# 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 [18]:
```
# If we want to start from right end we can use negative indexing. -1 is the last i
language = 'Python'
last_letter = language[-1]
print(last_letter) # n
second_last = language[-2]
print(second_last) # o
```

```
n
o
```

In [19]:
```
# Slicing

language = 'Python'
first_three = language[0:3] # starts at zero index and up to 3 but not include 3
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 [20]:
```
# Skipping character while splitting Python strings
language = 'Python'
pto = language[0:6:2] #
print(pto) # pto
```

Pto

In [22]:
```
# String Methods
# capitalize(): Converts the first character the string to Capital Letter

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

# count(): returns occurrences of substring in string, count(substring, start=.., e

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

# 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

# expandtabs(): Replaces tab character with spaces, default tab size is 8. It takes

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

# 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_name, job
print(sentence) # I am Asabeneh Yetayeh. I am a teacher. I live in Finland.

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

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

# 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

# isalpha(): Checks if all characters are alphabets

challenge = 'thirty days of python'
print(challenge.isalpha()) # True
num = '123'
print(num.isalpha())      # False

# isdecimal(): Checks Decimal Characters

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

# isdigit(): Checks Digit Characters

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

# isdecimal():Checks decimal characters

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


# isidentifier():Checks for valid identifier means it check if a string is a valid

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


# 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

# 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


# isnumeric():Checks numeric characters

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

# join(): Returns a concatenated string

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

# strip(): Removes both leading and trailing characters

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

# replace(): Replaces substring inside

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

# split():Splits String from Left

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

# title(): Returns a Title Cased String

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

# 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

# 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
```

```
Thirty days of python
3
1
2
True
False
thirty  days    of      python
thirty    days      of        python
5
0
I am Asabeneh Yetayeh. I am a teacher. I live in Finland.
The area of circle with 10 is 3.14
5
0
True
True
False
False
False
False
5
0
False
True
False
False
True
True
False
False
True
True
False
HTML#, CSS#, JavaScript#, React
 thirty days of python
thirty days of coding
['thirty', 'days', 'of', 'python']
Thirty Days Of Python
THIRTY DAYS OF PYTHON
tHIRTY dAYS oF pYTHON
True
False
```

In [ ]: