

```
In [11]: #Function and Lambda Function Revisit
# reading input values from user
username = input('What is your name? ')
age = int(input('What is your age? '))
greeting = input('write your greetings: ')
def my_function_with_args(username, age, greeting):
    print("Hello, %s , Your age is %d, From My Function!, I wish you %s"%(user
name, age, greeting))
my_function_with_args(username, age, greeting)
#Lambda:
double = lambda x: x * 2
print(double(5))
```

```
What is your name? sagar
What is your age? 24
write your greetings: happy
Hello, sagar , Your age is 24, From My Function!, I wish you happy
10
```

```
In [12]: #List Comprehension revisit
```

```
In [13]: sentence = "the quick brown fox jumps over the lazy dog"
words = sentence.split()
word_lengths = []
for word in words:
    if word != "the": word_lengths.append(len(word))
print(words)
print(word_lengths)
#Using List Comprehension:
word_lengths = [len(word) for word in words if word != "the"]

['the', 'quick', 'brown', 'fox', 'jumps', 'over', 'the', 'lazy', 'dog']
[5, 5, 3, 5, 4, 4, 3]
```

```
In [14]: def double(x):
    return x*2
list1 = [1, 2, 3, 4, 5, 6]
results = []
for i in list1:
    results.append(double(i))
#The following shows the use of map() in the above case:
def double(x):
    return x*2
list1 = [1, 2, 3, 4, 5, 6]
results = [x for x in map(double, list1)] #Lambda functions can also be used
print(results)

[2, 4, 6, 8, 10, 12]
```

```
In [15]: def filterVowels(letter):
          vowels = ['a', 'e', 'i', 'o', 'u']
          if(letter in vowels):
              return True
          else:
              return False
          filteredVowels = filter(filterVowels,letters)
          print('The filtered vowels are:')
          for vowel in filteredVowels:
              print(vowel)
```

```
-----
NameError                                Traceback (most recent call last)
<ipython-input-15-eb29fd0e4ca3> in <module>
      5     else:
      6         return False
----> 7 filteredVowels = filter(filterVowels,letters)
      8 print('The filtered vowels are:')
      9 for vowel in filteredVowels:

NameError: name 'letters' is not defined
```

```
In [16]: def add(a, b):
          return a+b
          # importing functools for reduce()
          import functools
          # initializing list
          list1 = [1, 3, 5, 6, 2] # using reduce to compute sum of list
          print ("The sum of the list elements is : ", end="")
          print (functools.reduce(add,list1))
```

The sum of the list elements is : 17

```
In [17]: veggies = ["Beans", "Broc", "Ban"]
          for veggie in veggies:
              if "Broc" in veggie:
                  continue
              print(veggie)
```

Beans
Ban

```
In [20]: # Creating the class
          class Pet(object):
              """Class object for a pet."""
              def __init__(self, species, name):
                  """Initialize a Pet."""
                  self.species = species
                  self.name = name
```

```
In [21]: # Creating an instance of a class
my_dog = Pet(species="dog",
              name="Scooby")
print (my_dog)
print (my_dog.name)

<__main__.Pet object at 0x000001F559BDD2C8>
Scooby
```

```
In [22]: # Creating the class
class Pet(object):
    """Class object for a pet."""
    def __init__(self, species, name):
        """Initialize a Pet."""
        self.species = species
        self.name = name
    def __str__(self):
        """Output when printing an instance of a Pet."""
        return f"{self.species} named {self.name}"
```

```
In [23]: # Creating an instance of a class
my_dog = Pet(species="dog",
              name="Scooby")

print (my_dog)
print (my_dog.name)
```

dog named Scooby
Scooby

```
In [24]: # Creating the class
class Pet(object):
    """Class object for a pet."""
    def __init__(self, species, name):
        """Initialize a Pet."""
        self.species = species
        self.name = name
    def __str__(self):
        """Output when printing an instance of a Pet."""
        return f"{self.species} named {self.name}"
    def change_name(self, new_name):
        """Change the name of your Pet."""
        self.name = new_name
```

```
In [25]: # Creating an instance of a class
my_dog = Pet(species="dog", name="Scooby")
print (my_dog)
print (my_dog.name)
```

dog named Scooby
Scooby

```
In [26]: # Using a class's function
my_dog.change_name(new_name="Scrappy")
print (my_dog)
print (my_dog.name)
```

dog named Scrappy
Scrappy

```
In [27]: #Inheritance
class Dog(Pet):
    def __init__(self, name, breed):
        super().__init__(species="dog", name=name)
        self.breed = breed
    def __str__(self):
        return f"A {self.breed} doggo named {self.name}"

scooby = Dog(breed="Great Dane", name="Scooby")
print (scooby)
```

A Great Dane doggo named Scooby

```
In [28]: scooby.change_name("Scooby Doo")
print (scooby)
```

A Great Dane doggo named Scooby Doo

```
In [29]: def filterVowels(letter):
    vowels = ['a', 'e', 'i', 'o', 'u']
    if(letter in vowels):
        return True
    else:
        return False
letters = "oop python"
filteredVowels = filter(filterVowels, letters)
print('The filtered vowels are:')
for vowel in filteredVowels:
    print(vowel)
```

The filtered vowels are:

o
o
o

In []: