Creating and Calling Funtions

In [1]:
```
# writing  first function
def printInfo():         #defines what the function does when called
    print("Name: Hrishikesh Devdikar")
    print("Age: 22")
printInfo()              #calls the function to run
printInfo()              #calls the function again
```

```
Name: Hrishikesh Devdikar
Age: 22
Name: Hrishikesh Devdikar
Age: 22
```

In [2]:
```
# performing a calculation in a function
def calc():
    x,y =5,10
    print(x+y)
calc()      # will run the block of code within calc and output 15
```

```
15
```

In [3]:
```
"""
Exercise 1:Define a function called myName, and have it print out your name
when called.

"""
def myName():
    print("Name: Hrishikesh Devdikar")
myName()
```

```
Name: Hrishikesh Devdikar
```

In [4]:
```
"""
Exercise 2:: Define a function that prints out all your favorite pizza toppings
called pizzaToppings. Call the function three times.
"""
def pizzaToppings():
    print("1. Chesse  2.Oregano   3.Chilli Flakes  4.Peperoni")
for i in range(3):
    pizzaToppings()
```

```
1. Chesse  2.Oregano   3.Chilli Flakes  4.Peperoni
1. Chesse  2.Oregano   3.Chilli Flakes  4.Peperoni
1. Chesse  2.Oregano   3.Chilli Flakes  4.Peperoni
```

Parameters

In [5]:
```
# passing a single parameter into a function
def printName(full_name):
    print(f"Your name is: {full_name}")
printName("Hrishikesh Devdikar")
printName("Prathmesh Devdikar")
```

```
Your name is: Hrishikesh Devdikar
Your name is: Prathmesh Devdikar
```

In [6]:
```
# passing multiple parameters into a function
```

```
def addNums(num1,num2):
    result = num1 + num2
    print(f"{num1} + {num2} = {result}")
addNums(5,8)
addNums(3.5,5.50)
```

```
5 + 8 = 13
3.5 + 5.5 = 9.0
```

In [7]:
```
# using a function to square all information
num1=[2,4,5,10]
num2=[1,3,6]
def squares(nums):
    for num in nums:
        print(num**2)
squares(num1)
squares(num2)
```

```
4
16
25
100
1
9
36
```

In [8]:
```
#setting default paramter values
def calArea(r, pi=3.14):
    area = pi * (r**2)
    print("Area of the circle: {}".format(area))
calArea(2)
#default parameters should always go after non-default parameters
```

```
Area of the circle: 12.56
```

In [11]:
```
#setting default parameter values(optional)
def printName(first,last,middle=""):
    if middle:
        print(f"{first} {middle} {last}")
    else:
        print(f"{first} {last}")
printName("Albert", "Einstein")
printName("Subhash" ,"Bose","Chandra" )
```

```
Albert Einstein
Subhash Chandra Bose
```

In [14]:
```
# explicitly assigning values to parameters by referencing the name
def addNums(num1,num2):
    print(num2)
    print(num1)
addNums(5,num2 = 2.5)
```

```
2.5
5
```

In [2]:
```
# using args parameter to take in a tuple of arbitary values
def outputData(name, *args):
    print(name)
    print(type(args))
    for arg in args:
        print(arg)
outputData("Hrishikesh Devdikar", 5, True, "Hello World")
```

```
Hrishikesh Devdikar
<class 'tuple'>
5
True
Hello World
```

In [4]:
```
# using kwargs parameter to take in a dictionary of arbitary values
def outputData(**kwargs):
    print(type(kwargs))
    print(kwargs["name"])
    print(kwargs["num"])
outputData(name = "Hrishikesh Devdikar", num = 5, b=True)
```

```
<class 'dict'>
Hrishikesh Devdikar
5
```

In [9]:
```
"""
Exercise 3:Ask the user to input a word, and pass that word into a function
that checks if the word starts with an uppercase. If it does output “True”,
otherwise “False”
"""
def verify(ask):
    print(ask)
    if ask == ask.title():
        print("True")
    else:
        print("False")

ask = input("Enter a word: ")
verify(ask)
```

```
Enter a word: Ikigai
Ikigai
True
```

In [4]:
```
"""
Exercise 4:Define a function that takes in two arguments, first_name and last_
name, and makes both optional. If no values are passed into the parameters, it
should output “No name passed in”; otherwise, it should print out the name.
"""
def optName(first_name="",last_name=""):
    if first_name and last_name:
        print(first_name,last_name)
    else:
        print("No name passed in.")
optName("Harry","Hole")
```

```
Harry Hole
```

Return Statement

In [5]:
```
#using return keyword to return the sum of two numbers
def addNums(num1,num2):
    return num1 + num2
num=addNums(5.5,4.5)        #saves returned value into num
print(num)
print(addNums(10,10))       #does'nt save the return value
```

```
10.0
```

```
20
```

In [6]:
```
# shortcut syntax using a ternary operator
def searchList(aList, el):
    return True if el in aList else False
result = searchList(["one",2,"three"], 2)
print(result)
```

```
True
```

In [9]:
```
"""
Exercise 5:Create a function that takes in a first and last name and returns the
two names joined together.
"""
def name(first_name,last_name):
    return first_name + last_name
out=name("Hrishikesh","Devdikar")
print(out)
```

```
HrishikeshDevdikar
```

In [12]:
```
"""
Exercise 6:Within a function, ask for user input. Have this function return that
input to be stored in a variable outside of the function. Then print out the input.
"""
def inputer():
     return input("Enter user Input: ")
store = inputer()
print(store)
```

```
Enter user Input: Beyond Infinity We Go!!
Beyond Infinity We Go!!
```

Scope

In [13]:
```
# where global variables can be accessed
number =5
def scopeTest():
    number += 1      #not accessible due to function level scope
scopeTest()
```

```
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
<ipython-input-13-27b5fe3d52ac> in <module>
      3 def scopeTest():
      4     number += 1
----> 5 scopeTest()

<ipython-input-13-27b5fe3d52ac> in scopeTest()
      2 number =5
      3 def scopeTest():
----> 4     number += 1
      5 scopeTest()

UnboundLocalError: local variable 'number' referenced before assignment
```

In [14]:
```
# accessing variables defined in a function
def scopeTest():
    word = "Function Variable"
    return word
```

```
value =scopeTest()
print(value)
```

```
Function Variable
```

In [15]:
```
#changing list items values by index
sports=["Soccer","Baseball","Chess","Tennis"]
def change(aList):
    aList[0] = "Hockey"
print("Before Altering: {}".format(sports))
change(sports)
print("After Altering: {}".format(sports))
```

```
Before Altering: ['Soccer', 'Baseball', 'Chess', 'Tennis']
After Altering: ['Hockey', 'Baseball', 'Chess', 'Tennis']
```

In [17]:
```
"""
Exercise 7: Create a function that will change the list passed in with a parameter
of name at a given index. Such that if I were to pass in “Bill” and index 1,
it would change “Rich” to “Bill.” Use the list and function definition in the
following:
>>> names = ['Bob', 'Rich', 'Amanda']
>>> def changeValue(aList, name, index):
"""
names = ['Bob', 'Rich', 'Amanda']
def changeValue(aList, name, index):
    aList[index] = name
print(f"Before Altering {names}")
changeValue(names,"Bill",1)
print(f"After ALtering: {names}")
```

```
Before Altering ['Bob', 'Rich', 'Amanda']
After ALtering: ['Bob', 'Bill', 'Amanda']
```

Project : Creating a Shopping Cart

In [24]:
```
# import necessary functions
from IPython.display import clear_output

#global list variable
cart=[]

#create function to add items in cart
def addItems(item):
    clear_output()
    cart.append(item)
    print("{} has been added to cart.".format(item))

#create function to remove items from cart
def removeItems(item):
    clear_output()
    try:
        cart.remove(item)
        print("{} has been removed.".format(item))
    except:
        print("We cannot remove that item")

#create a function to show items in cart
def showCart():
    clear_output()
```

```
    if cart:
        print("Here is your cart:")
        for item in cart:
            print("- {}".format(item))
    else:
        print("your cart is empty")

# create function to clear items from cart
def clearCart():
    clear_output()
    cart.clear()
    print("Your cart is empty")

# create main function that loops wntil the user quits
def main():
    done = False
    while not done:
        ans = input("add/remove/show/clear/quit : ").lower()
        # base case
        if ans == "quit":
            print("Thanks for using our program")
            showCart()
            done =True
        elif ans == "add":
            item = input("What would you like to add?").title()
            addItems(item)
        elif ans == "remove":
            showCart()
            item = input("What item would you like to remove?").title()
            removeItems(item)
        elif ans =="show":
            showCart()
        elif ans == "clear":
            clearCart()
        else:
            print("Sorry that was not an option")
main()     # run the program
```

```
Here is your cart:
- Apple
- Banana
- Chickoo
- Drangonfruit
```