

Python Programming Reference Sheet – Advanced

This the companion cheat sheet to the “Python Programming Reference Sheet – Intermediate”. We cover more advanced topics in the Python Programming language including error handling, classes and objects.

1) Error Handling: Try and except

Python try & except are similar to try & catch in Java or PHP.

```
try:
    y=10/0
    print(z)
except ZeroDivisionError:
    print ("Division by zero was detected")
except NameError:
    print("Variable z was not defined")
except:
    print("Unknown error occurred")
else:
    print("No errors occurred")
```

#Which ever error is caught first will execute. All others will not.

2) Functions

There are many ways to define functions. Here are some examples.

```
def myFirstFunction(): #no arguments
    print("Hello World")

def mySecondFunction(age): #single argument
    print("You are " + str(age) + " years old")

def myThirdFunction(numA, numB): #2 arguments
    print("The sum of 2 numbers is: " + str(numA + numB))

def myFourthFunction(*myargs): #arbitrary arguments
    print("The third argument is " + myargs[2])

def myFifthFunction(**myvars): #arbitrary keyword args
    print("The temperature is " + str(myvars["temp"]))

def mySixthFunction(name = "John"): #default parameter
    print("Your name is " + name)

#calling functions above
myFirstFunction()
mySecondFunction(20)
myThirdFunction(10, 30)
myFourthFunction("Jane", "Mary", "Lucy", "Elizabeth")
myFifthFunction(color = "Red", temp = 25)
mySixthFunction()
```

3) Classes and Objects

The following example created 2 classes, Car and ElectricCar. The Car class is the parent class while ElectricCar class inherits methods and variables of the Car (parent) class.

```
#Sample Car class with 2 functions
class Car:
    def __init__(self, model, make): #constructor method
        self.model = model
        self.make = make
        self.year = 2021

    def myPrintCarInformation(self):
        print("Make: " + self.make)
        print("Model: " + self.model)

    def myCustomHonkFunction(self):
        print("Honk!! Honk!!")
```

#Instantiate new car

```
car = Car("Honda", "Civic")
car.myPrintCarInformation()
car.myCustomHonkFunction()
```

#sample child class illustrating class inheritance

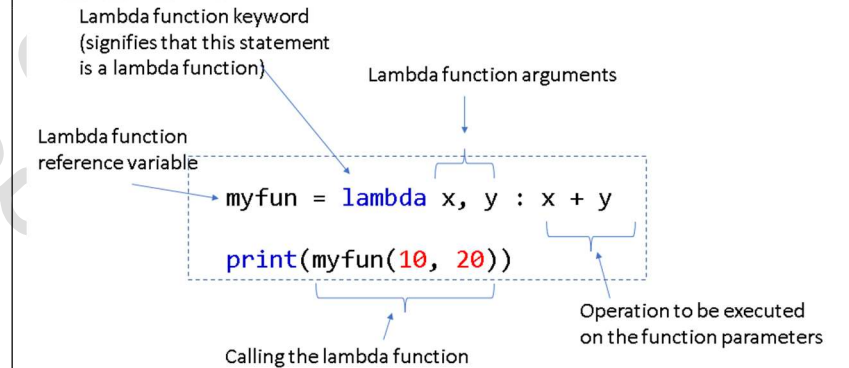
```
class ElectricCar(Car):
    #example class variable
    color = "Black"
    #example instance variable
    def myCustomSignature(self):
        print("Hi, I am an electric car!")
```

```
#Instantiate a new (electric) car
car2 = ElectricCar("Tesla", "Model S");
car2.myPrintCarInformation();
car2.myCustomSignature()
#illustrate using class variable
print(ElectricCar.color)
#illustrate using instance variable
print(car2.year)
```

Note that Python does not have a constant variable as in other languages. If you need a constant, you can create an equivalent by using a class property with only a getter method.

4) Lambda Functions

Python Lambda functions are relatively small functions with a single expression and no assigned name.



5) Variable Scope

In general, variables defined outside of classes and functions have global scope providing they are reference by the global keyword within the function. Here is an example.

```
a=200
b="some string"

def myFunction():
    global a
    a = 300
    b="another string" # this creates a function variable

myFunction()

print(a)
print(b)

#output
300
some string
```

Note that variable “b” does not see another variable “b” created outside of the function while the variable “a” within the function is declared of the global scope outside of the myFunction().