Introduction to Programming with Python

Day 3

Vivek Murugesan
vivek@bellatrix.aero

March 20, 2021

Course Overview

1. Introduction
2. Lists, Tuples, Dicts
3. Conditional and Looping Statements
4. Errors and Error Handlers
5. Functional Programming
6. External Libraries, Numpy, Scipy, Matplotlib
7. Advanced Plotting
8. Working with Files (CSV, Excel, Text)
9. GUI creation with Tkinter
10. Useful Utilities and Way Forward!

Errors and Error Handling

Common Errors

Some of the most common error we find in python are,

- SyntaxError
- IndentationError
- ValueError
- TypeError
- NameError
- ImportError
- IndexError
- ZeroDivisionError
- UnboundLocalError
- FileNotFoundError

Simple Error Handling!

```
try:
        Code to be run
except:
        Code to be run when something happened
```

```
i=20
try:
        print(i/0)
except:
        print("Some Error Occured!")
```

Comprehensive Error Handling

```
i=20
j=5

try:
        print(i/j)

except ZeroDivisionError:
        print(“”You cannot divide by Zero!”)

except TypeError:
        print(“”You cannot divide by a String!”)
print(“”End of program”)
```

Error Handling - finally

```
i=20
j=5
try:
        print(i/j)

except ZeroDivisionError:
        print(“”You cannot divide by Zero!”)

finally:
        print(“”This will get Executed no matter what!”)
print(“”End of program”)
```

Functions

Functional Programming in Python

- functions are exceptionally useful in programming
- functions will reduce the code length
- functions can be reused n number of time
- functions reduced the development time
- functions makes the code looks cleaner

Defining a Function in Python

- To define a function we use def command

```
def functionName():
        Code that is to be executed
```

```
def greet():
        print("Hello There!")

greet()
```

Function with Arguments

- You can send data to functions. This data is called arguments

```
def greet(name):
        print("Good Evening ".format(name))

greet("Vivek")
```

Function with return

- You can also receive data from functions!

```
def greet(name):
        return("Good Evening ".format(name))

message=greet("Vivek")
print(message)
```

Function with Default Arguments

```
def add(x,y=10):
        return (x+y)

value1=add(2,5)
value2=add(2)
print(value1)
print(value2)
```

Function with Arbitrary Arguments

```
def summation(*values):
        sum =0
        for value in values:
                sum=sum+value
        return sum

value1=sum(1,2,3,4)
value2=sum(1,2,3,4,5,6,7,8,9)
print(value1)
print(value2)
```

print(“Thank You!”)