Fourth Industrial Summer School

Day 4

Python Programming IV

Session Objectives

- ✓ Functions
- ✓ Classes
  - Objects
- ✓ Files
  - Open
  - Read
  - Write
  - Close
  - Delete

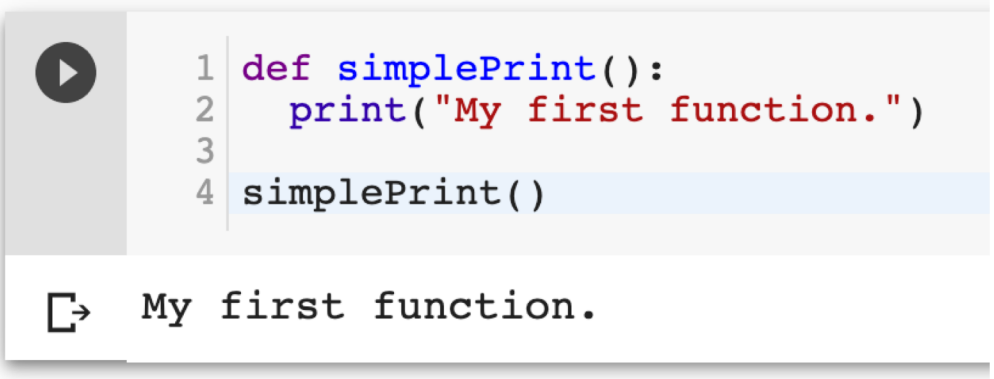
Part A

Functions / Classes

Functions

- A block of code to be executed when called

- A function can have:
  - Input parameters
  - Output returned



```
def simplePrint():
  print("My first function.")

simplePrint()
```

```
My first function.
```

Functions

- Parameters
  - Input passed to the function
  - Add as many as you want separated with a comma
  - You can send any data type as a parameter to a function (string, number, list, dictionary etc.)

```
1 def printString(str):
2   print(str + " world!")
3
4 printString("Hello")
5 printString("Greetings")
```

```
Hello world!
Greetings world!
```

Functions

- Default value for a parameter

```
1 def printString(str = "Greetings"):
2   print(str + " world!")
3
4 printString("Hello")
5 printString()
```

```
Hello world!
Greetings world!
```

Functions

- Passing a list as a parameter

```
1 def printList(param):
2   for x in param:
3     print(x)
4
5 cars = ["BMW", "Lexus", "Honda"]
6 printList(cars)
```

```
BMW
Lexus
Honda
```

Functions

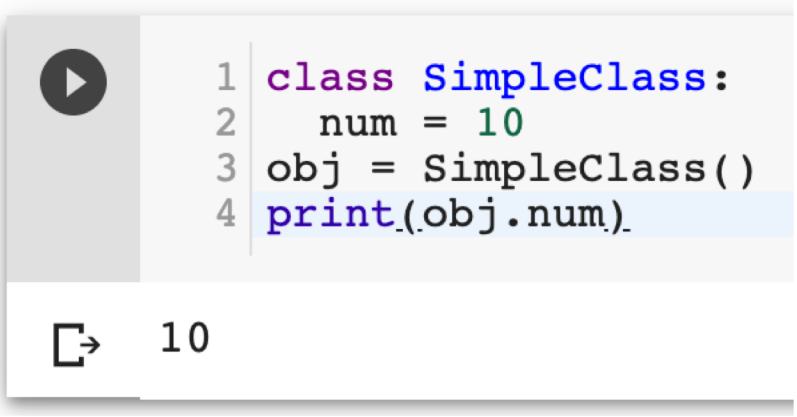
- Return value

```
1 def multi(x,y):
2   return x * y
3 def div(x,y):
4   return x / y
5 print(multi(3,4))
6 print(div(9,4))
```

```
12
2.25
```

Classes

- Python is an Object Oriented programming language.
  - class keyword



Classes

- Need to initialize an object
  - __init__() function
  - Always executed when class is initiated

```
1 class Student:
2   def __init__(self,name,id):
3     self.name = name
4     self.id = id
5
6 stu = Student("Ahmed AlAhmed",18)
7
8 print(stu.name)
9 print(stu.id)
```

```
Ahmed AlAhmed
18
```

Classes

- Object methods

```
class Student:
  def __init__(self,name,id):
    self.name = name
    self.id = id

  def printName(self):
    print("My name is ",self.name)

stu = Student("Ahmed AlAhmed",18)

stu.printName()
```

```
My name is  Ahmed AlAhmed
```

Classes

- Self parameter
  - Always first parameter
  - is a reference to the current object
  - Can be renamed to anything

```
1  class Student:
2    def __init__(anyname,name,id):
3      anyname.name = name
4      anyname.id = id
5
6    def printName(anyname):
7      print("My name is ",anyname.name)
8
9  stu = Student("Ahmed AlAhmed",18)
10
11 stu.printName()
```

```
My name is  Ahmed AlAhmed
```

Classes

- Modify attribute value
- Delete object attribute / object
  - del keyword

```
11 # Change value of an object attribute
12 stu.name = "Mohammed AlAhmed"
13 # Delete an attribute inside object
14 del stu.name
15 # Delete an object
16 del stu
```

Hands on session

Problem Solving

Part B

Files

Files

- File handling in Python is an easy task.

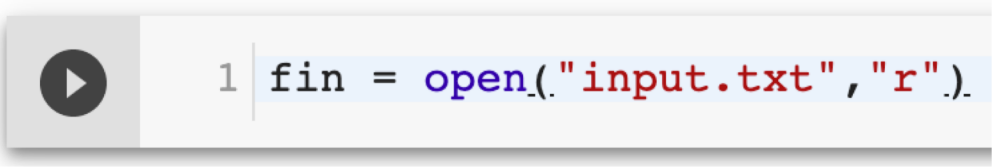
- Python has several functions to handle files:
  - Creating
  - Reading
  - Updating
  - Closing
  - Deleting

File Opening

- File opening in Python is using the open() function.
  - Two parameters: filename and mode

- Four modes to open the file

Mode	Functionality
"r"	Read - Default value. Opens a file for reading, error if the file does not exist
"a"	Append - Opens a file for appending, creates the file if it does not exist
"w"	Opens a file for writing, creates the file if it does not exist
"x"	Creates the specified file, returns an error if the file exists



File Opening

- Python by default open text files
  - however, if the file is binary (e.g. image), you need to specify that with the mode.

Mode	Functionality
"t"	Default value - Text
"b"	Binary mode

```
1 ftext = open("input.txt","rt")
2 fimage = open("Colab.png","rb")
```

File Reading

- Different functions for reading from a file
  - however, if the file is binary (e.g. image), you need to specify that with the mode.

Name	Functionality
read()	Reads file content
readline()	Reads single line

```
1 ftext = open("input.txt","r")
2 print(ftext.read())
```

```
KFUPM summer school
line 2
line 3
```

File Reading

```
1 ftext = open("input.txt","r")
2 print(ftext.readline())
```

```
KFUPM summer school
```

```
1 ftext = open("input.txt","r")
2 # loop through the content of the file
3 for x in ftext:
4   print(x)
```

```
KFUPM summer school

line 2

line 3
```

File Writing

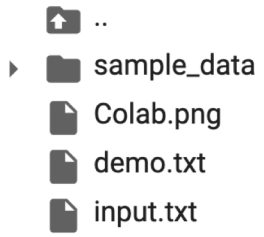
- Open file for writing
  - Three modes to open a file for writing using the open() function

Mode	Functionality
"a"	Append to the end of the existing file
"w"	Overwrite any existing content
"x"	create a file, returns an error if the file exist

- Use write() function to write to the file

```
1 fnew = open("demo.txt","w")
2 fnew.write("My first created file using Python")
3 fnew.close()
4
5 # Open the same file for reading
6
7 fin = open("demo.txt", "r")
8 print(fin.read())
```

```
My first created file using Python
```



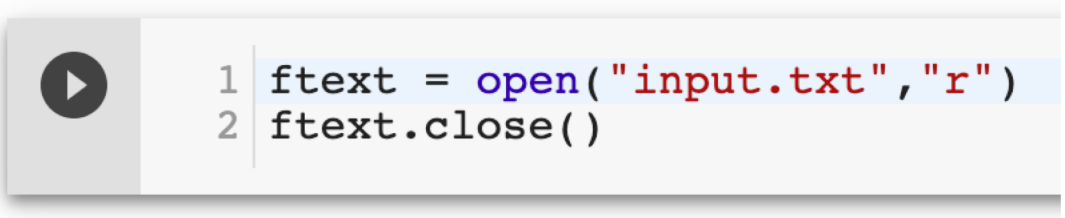
File Writing

- Open previous file in append mode

```
1 fnew = open("demo.txt","a")
2 fnew.write("Appending an existing file.")
3 fnew.close()
```

File Closing

- Close files after you are done
  - close() function



```
1 ftext = open("input.txt","r")
2 ftext.close()
```

File Deletion

- import the OS module
- Use the os.remove() function

```
1 import os
2 os.remove("demo.txt")
```

Hands on session

Problem Solving