Python Basics

Course 4- Coursera Week 4

Objectives

- Understand Reading Files with Open
- Understand Writing Files with Open
- Understand Pandas
- Understand One Dimensional Numpy
- Understand Two Dimensional Numpy
- Understand Simple APIs

Python File Open

File handling is an important part of any web application.

Python has several functions for creating, reading, updating, and deleting files.

File Handling

The key function for working with files in Python is the open() function.

The open() function takes two parameters; filename, and mode.There are four different methods (modes) for opening a file:

"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"	- Write - Opens a file for writing, creates the file if it does not exist
"x"	- Create - Creates the specified file, returns an error if the file exists
In addition you can specify if the file should be handled as binary or text mode	
"t"	- Text - Default value. Text mode
"b"	- Binary - Binary mode (e.g. images)

Syntax

To open a file for reading it is enough to specify the name of the file:

```
f = open("demofile.txt")
```

The code above is the same as:

```
f = open("demofile.txt", "rt")
```

Because "r" for read, and "t" for text are the default values, you do not need to specify them.

Note: Make sure the file exists, or else you will get an error.

Open a File on the Server

Assume we have the following file, located in the same folder as Python:

demofile.txt

Hello! Welcome to demofile.txt

This file is for testing purposes.

Good Luck!

Cont.

To open the file, use the built-in open() function.

The open() function returns a file object, which has a read() method for reading the content of the file:

```
1  f = open("demofile.txt", "r")
2  print(f.read())
```

If the file is located in a different location, you will have to specify the file path, like this:

```
1  f = open("D:\\myfiles\welcome.txt", "r")
2  print(f.read())
```

Read Only Parts of the File

By default the **read()** method returns the whole text, but you can also specify how many characters you want to return:

Try it

Return the 5 first characters of the file:

```
f = open("demofile.txt", "r")

print(f.read(5))
```

Read Lines

You can return one line by using the readline() method:

Try it

Read one line of the file:

```
f = open("demofile.txt", "r")

print(f.readline())
```

CONT.

By calling readline() two times, you can read the two first lines:

Try It

Read two lines of the file:

```
f = open("demofile.txt", "r")

print(f.readline())

print(f.readline())
```

CONT

By looping through the lines of the file, you can read the whole file, line by line:

Try It

Loop through the file line by line:

```
f = open("demofile.txt", "r")
for x in f:
  print(x)
```

Close Files

It is a good practice to always close the file when you are done with it.

Try It

Close the file when you are finish with it:

```
f = open("demofile.txt", "r")

print(f.readline())

f.close()
```

Python File Write

Write to an Existing File

To write to an existing file, you must add a parameter to the open() function:

"a" - Append - will append to the end of the file

"w" - Write - will overwrite any existing content

CONT

Try It

Open the file "demofile2.txt" and append content to the file:

```
f = open("demofile2.txt", "a")
f.write("Now the file has more content!")
f.close()

#open and read the file after the appending:
f = open("demofile2.txt", "r")
print(f.read())
```

Note: the "w" method will overwrite the entire file.

CONT

Try It

Open the file "demofile3.txt" and overwrite the content:

```
f = open("demofile3.txt", "w")
f.write("Woops! I have deleted the content!")
f.close()

#open and read the file after the appending:
f = open("demofile3.txt", "r")
print(f.read())
```

Note: the "w" method will overwrite the entire file.

Create a New File

To create a new file in Python, use the `open()` method, with one of the following parameters:

`"x"`	- Create - will create a file, returns an error if the file exist
`"a"`	- Append - will create a file if the specified file does not exist
`"w"`	- Write - will create a file if the specified file does not exist

CONT.

Try It

Create a file called "myfile.txt":

```
f = open("myfile.txt", "x")
```

Try It

Create a new file if it does not exist:

```
f = open("myfile.txt", "w")
```

NumPy Creating Arrays

NumPy is used to work with arrays. The array object in NumPy is called ndarray.

We can create a NumPy ndarray object by using the array() function.

Try It

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

print(type(arr))
```

Note:

type(): This built-in Python function tells us the type of the object passed to it. Like in above code it shows that arr is numpy.ndarray type.

CONT

To create an **ndarray**, we can pass a list, tuple or any array-like object into the **array**() method, and it will be converted into an **ndarray**:

Try It

Use a tuple to create a NumPy array:

```
import numpy as np

arr = np.array((1, 2, 3, 4, 5))

print(arr)
```

Dimensions in Arrays

A dimension in arrays is one level of array depth (nested arrays).

nested array: are arrays that have arrays as their elements.

0-D Arrays

0-D arrays, or Scalars, are the elements in an array. Each value in an array is a 0-D array.

Try It

Create a 0-D array with value 42

```
import numpy as np

arr = np.array(42)

print(arr)
```

1-D Arrays

An array that has 0-D arrays as its elements is called uni-dimensional or 1-D array.

These are the most common and basic arrays.

Try It

Create a 1-D array containing the values 1,2,3,4,5:

```
import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)
```

2-D Arrays

An array that has 1-D arrays as its elements is called a 2-D array.

These are often used to represent matrix or 2nd order tensors.

Try It

Create a 2-D array containing two arrays with the values 1,2,3 and 4,5,6:

```
import numpy as np

arr = np.array([[1, 2, 3], [4, 5, 6]])

print(arr)
```

3-D ARRAYS

An array that has 1-D arrays as its elements is called a 2-D array.

These are often used to represent matrix or 2nd order tensors.

Try It

Create a 2-D array containing two arrays with the values 1,2,3 and 4,5,6:

```
import numpy as np

arr = np.array([[1, 2, 3], [4, 5, 6]])

print(arr)
```