

SUMMER SCHOOL

Introduction to Python Programming

Reading and Writing Files

Binjamin Barsch

Summary of commands used in this lecture

Command	Meaning
print()	A Python function that prints text
open()	A Python function that opens a file
read()	A Python function that reads the contents of a file
readline()	A Python function that returns one line from the file
write()	A Python function that writes text to a file
close()	A Python function that closes a file
os.remove()	A Python function that removes a file
os.rmdir()	A Python function that removes a directory
os.listdir()	A Python function that lists the contents of a directory

Reading Files & Writing Files

File handling is one of the useful features of Python programming. To be able to read, edit, and create files are just some of the features that are available. Let us first start off with reading files.

In CoCalc, let us first create a file with some text in it in the Linux terminal:

```
~$ touch myfile.txt
~$ nano myfile.txt
~$ cat myfile.txt
Summer School 2021
Welcome!
~$
```

Now let us read and print the contents of the file in the Python interpreter:

```
~$ python
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> f = open("myfile.txt","r")
>>> print(f.read())
Summer School 2021
Welcome!
>>>
>>> f.close()
```

We used the “**open()**” function to open the file “**myfile.txt**” in order to read it using “**r**”. Then we used the “**read()**” function to read all its contents.

Now let us try that again but only read one line at a time:

```
>>> f = open("myfile.txt","r")
>>> print(f.readline())
Summer School 2021

>>> print(f.readline())
Welcome!
>>>
>>> f.close()
```

But we know now that any repetitive task should be automated! We can use a “**for**” loop to print all the lines one-by-one:

```
>>> f = open("myfile.txt","r")
>>> for line in f:
...     print(line)
...
Summer School 2021

Welcome!

>>>
>>> f.close()
>>>
```

Now let us write to a file. We have two options. Either we can append the text to the file or overwrite the entire file. If the file does not exist it will create it in both cases. In the “**open ()**” function the letter “**a**” stands for append and the letter “**w**” stands for overwrite:

```
>>> f = open("myfile.txt","a")
>>> f.write("A new line of text")
18
>>> f.close()
>>>
>>> f = open("myfile.txt","r")
>>> print(f.read())
Summer School 2021
Welcome!
A new line of text
>>>
```

Lastly to delete a file or folder we need to import a very handy module called “**os**” and use one of its pre-defined functions. Let us first create a random file and folder in the Linux terminal:

```
~$
~$ ls
2021-02-03-165527.term  'Welcome to CoCalc.ipynb'
~$
~$ touch myfile.txt
~$ mkdir myfolder
~$ ls
2021-02-03-165527.term  'Welcome to CoCalc.ipynb'  myfile.txt  myfolder
~$
```

Now in the Python interpreter let us see if we can first see this file and folder using the function “`os.listdir()`”:

```
~$ python
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import os
>>> os.listdir('/home/user')
['.local', '.ssh', '.bash_history', '.config', '.sage', '.jupyter-blobs-
v0.db', 'myfile.txt', '.gitexcludes', '.gitconfig', 'myfolder', '2021-02-
03-165527.term', '.Welcome to CoCalc.ipynb.sage-jupyter2', '.bash_profile',
'.python_history', '.snapshots', 'Welcome to CoCalc.ipynb', '.2021-02-03-
165527.term-0.term', '.smc', '.bashrc']
>>>
```

Now we can delete these files:

```
>>> os.remove("myfile.txt")
>>> os.rmdir("myfolder")
>>>
>>> os.listdir('.')
['.local', '.ssh', '.bash_history', '.config', '.sage', '.jupyter-blobs-
v0.db', '.gitexcludes', '.gitconfig', '2021-02-03-165527.term', '.Welcome
to CoCalc.ipynb.sage-jupyter2', '.bash_profile', '.python_history',
'.snapshots', 'Welcome to CoCalc.ipynb', '.2021-02-03-165527.term-0.term',
'.smc', '.bashrc']
>>>
>>>
~$ ls
 2021-02-03-165527.term  'Welcome to CoCalc.ipynb'
~$
```

If you think about it, what we just did in this lesson is the same as what we could do in the Linux terminal? So, which do we choose to use Python or Linux terminal...wow!

Summary

This is the end of lesson 7 where we looked at reading and writing files in Python. The next few lessons will look at more advanced features of Python!