

# The OS module

Thomas Schwarz, SJ

# Using the os-module

- With the os-module, you can obtain greater access to the file system
  - Here is code to get the files in a directory

```
import os

def list_files(dir_name):
    files = os.listdir(dir_name)
    for my_file in files:
        print(my_file,
              os.path.getsize(dir_name+"/"+my_file))

list_files("Example")
```

# Using the os-module

```
import os
```

```
def list_files(dir_name):  
    files = os.listdir(dir_name)  
    for my_file in files:  
        print(my_file,  
              os.path.getsize(dir_name+"/"+my_file))
```

```
list_files("Example")
```



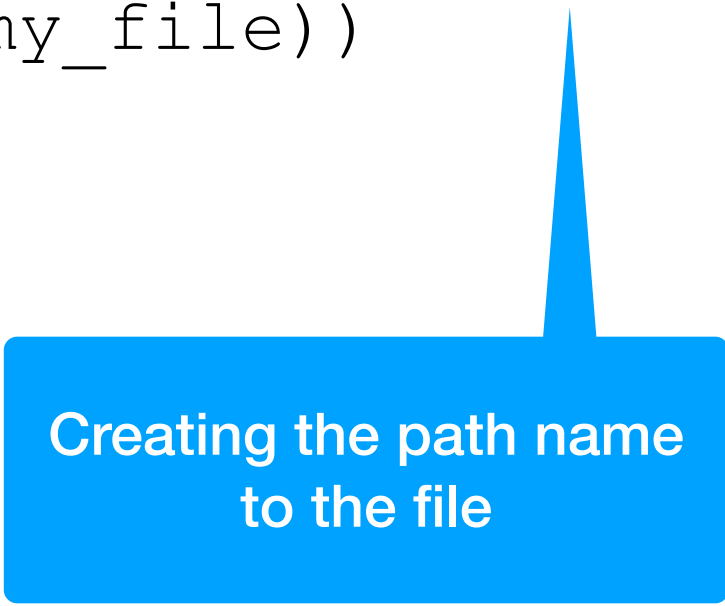
Get a list of file names in the directory

# Use the os-module

```
import os

def list_files(dir_name):
    files = os.listdir(dir_name)
    for my_file in files:
        print(my_file,
os.path.getsize(dir_name+"/"+my_file))

list_files("Example")
```



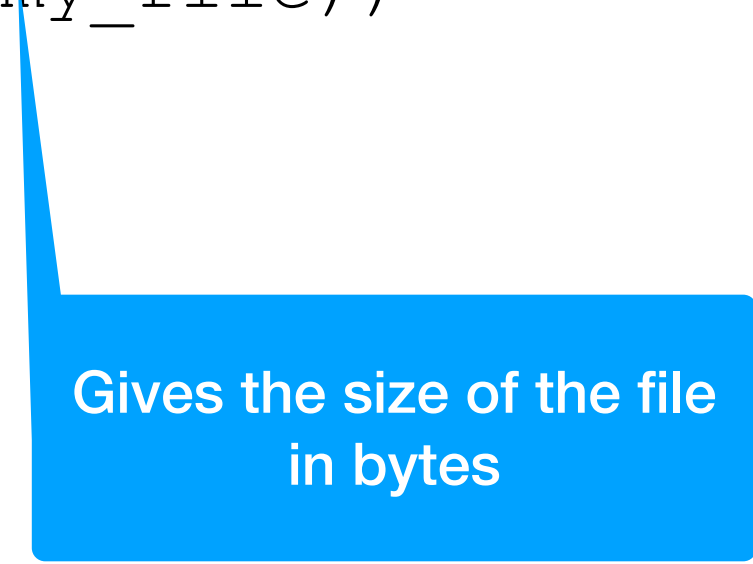
Creating the path name  
to the file

# Use the os-module

```
import os

def list_files(dir_name):
    files = os.listdir(dir_name)
    for my_file in files:
        print(my_file,
os.path.getsize(dir_name+"/"+my_file))

list_files("Example")
```



Gives the size of the file  
in bytes

# Use the os-module

```
import os

def list_files(dir_name):
    files = os.listdir(dir_name)
    for my_file in files:
        print(my_file,
os.path.getsize(dir_name+"/"+my_file))

list_files("Example")
```



List and

# Use the os-module

- Output:
  - Note the Mac-trash file

```
RESTART: /Users/thomasschwa  
le14/generator.py  
.DS_Store 6148  
results1.csv 384  
results0.csv 528  
results2.csv 432  
results3.csv 368  
results4.csv 464
```

# Use the os-module

- Using the listing capability of the os-module, we can process all files in a directory
  - To avoid surprises, we best check the extension
  - Assume a function `process_a_file`
    - Our function opens a comma-separated (.csv) file
    - Calculates the average of the ratios of the second over the first entries



# Use the os-module

- The process\_a\_file takes the file-name
- Calculates the average ratio

```
def process_a_file(file_name):  
    with open(file_name, "r") as infile:  
        suma = 0  
        nr_lines = 0  
        for line in infile:  
            nr_lines+=1  
            array = line.split(',')  
            suma+= float(array[1])/float(array[0])  
    return suma/nr_lines
```


```
1.290, 12.495  
2.295, 11.706  
3.063, 9.083  
4.058, 4.112  
1.147, 1.093  
1.997, 8.833  
2.781, 10.032  
0.929, 9.373  
1.858, 14.439  
3.022, 21.861  
3.751, 19.097  
1.147, 1.093  
1.997, 8.833  
2.781, 10.032  
4.225, 9.733  
5.455, 15.820  
6.151, 20.939  
6.573, 26.547  
8.058, 33.335  
9.132, 37.546  
10.474, 47.130  
11.207, 50.559  
1.147, 1.093  
1.997, 8.833  
2.781, 10.032  
3.751, 19.097  
4.225, 9.733  
5.455, 15.820  
6.151, 20.939  
6.573, 26.547  
8.058, 33.335  
9.132, 37.546  
10.474, 47.130  
11.207, 50.559  
0.280, 37.546  
37.029, 47.130  
37.459, 50.559  
27.295, 37.546  
34.994, 50.559  
37.458, 62.268  
66.393, 84.574  
62.255, 93.389  
84.116, 103.726  
87.145, 111.623  
0.933, 119.797  
0.048, 130.094  
0.667, 143.306  
0.947, 154.047  
0.509, 169.502  
0.398, 178.782  
0.806, 190.953  
0.448, 199.131  
0.716, 214.514  
0.198, 232.827  
0.358, 245.687  
0.137, 256.452  
0.7, 270.849  
0.063, 288.109  
33.288, 303.786
```

# Use the os-module

- To process the directory
  - Get the file names using os
  - For each file name:
    - Check whether the file name ends with .csv
    - Call the process\_a\_file function
    - Print out the result

# Use of the os-module

```
def process_files(dir_name):  
    files = os.listdir(dir_name)  
    for my_file in files:  
        if my_file.endswith('.csv'):  
            print(my_file, process_a_file(  
                "Example/{}".format(my_file)))
```



Using format to create the  
file name

# Use of the os-module

```
RESTART: /Users/thomasschwarz/Docu  
le14/generator.py  
>>> process_files('Example')  
results1.csv 5.2819632072675295  
results0.csv 5.920382285263983  
results2.csv 5.7506863373894666  
results3.csv 4.801235259621119  
results4.csv 6.409464135625922
```