## Files - I

**In the following series of notebooks we will look at how to write to and read from files.**

**In this notebook we will look at the Python syntax for writing data to a text file. The syntax for opening a file is handle = open(filename, mode). The `open()` function returns a file object (also called a *file handle*) that contains methods for working on the file.**

In the code below, f is the name for the file handle. out.txt is the name of the file we are working with. The second argument in the open function is the mode and is used to tell Python exactly what we plan to do with the file. The value of w for mode indicates that we want to open this file for writing. Note that if the file exists and contains data, all that data will be wiped out and fresh data will be written into the file.

**To write to a text file, we use the `write()` method.**
**The argument to the `write()` method is a string value. Hence if we wish to write any other type of data, it must be first converted to string.**

**You must always close the file when you are done with it. In this course, I will treat it as an error if you do not do so, unless we use the `with/as` construct that I will discuss later.**

In [5]:

```python
f = open('out.txt', 'w') # opened for writing

for i in range(11):
    f.write('This is line ' + str(i) + '\n') # Note the newline character
print('done')

f.close()
```

```
done
```

**The `write()` method writes the string argument to the file and returns the number of characters written. In the following example, we write the string to the file while simultaneously printing out the number of characters that were written.**

In [6]:

```python
f = open('out.txt', 'w') # opened for writing

for i in range(11):
    # This line will write the string to the file and also print out the number of characters that were written
    print(f.write('This is line ' + str(i) + '\n'))
print('\ndone')

f.close()
```

```
15
15
15
15
15
15
15
15
15
15
16

done
```