01. How to run Python code

December 26, 2021

1 Running Python code

1.1 Interactively (demo)

To run Python code interactively, one can use the standard Python prompt, which can be launched by typing `python` in your standard shell:

```
$ python
Python 3.5.2 (default, Jul 5 2016, 11:41:13)
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

The `>>>` indicates that Python is ready to accept commands. If you type `a = 1` then press enter, this will assign the value `1` to `a`. If you then type `a` you will see the value of `a` (this is equivalent to `print a`):

```
>>> a = 1
>>> a
1
```

The Python shell can execute any Python code, even multi-line statements, though it is often more convenient to use Python non-interactively for such cases.

The default Python shell is limited, and in practice, you will want instead to use the IPython (or interactive Python) shell. This is an add-on package that adds many features to the default Python shell, including the ability to edit and navigate the history of previous commands, as well as the ability to tab-complete variable and function names. To start up IPython, type:

```
$ ipython
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  5 2016, 11:41:13)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]:
```

The first time you start up IPython, it will display a message which you can skip over by pressing `ENTER`. The `>>>` symbols are now replaced by `In [x]`, and output, when present, is prepended with

`Out [x]`. If we now type the same commands as before, we get:

```
In [1]: a = 1

In [2]: a
Out[2]: 1
```

If you now type the up arrow twice, you will get back to `a = 1`.

1.2 Running scripts (demo)

While the interactive Python mode is very useful to exploring and trying out code, you will eventually want to write a script to record and reproduce what you did, or to do things that are too complex to type in interactively (defining functions, classes, etc.). To write a Python script, just use your favorite code editor to put the code in a file with a `.py` extension. For example, we can create a file called `test.py` containing:

```
a = 1
print(a)
```

We can then run the script on the command-line with:

```
$ python test.py
1
```

Note: The `print` function is necessary, because typing `a` on its own will only print out the value in interactive mode. In scripts, the printing has to be explicitly requested with the print command. To print multiple variables, just separate them with a comma after the print command:

```
print(a, 1.5, "spam")
```

1.3 Combining interactive and non-interactive use (demo)

It can sometimes be useful to run a script to set things up, and to continue in interactive mode. This can be done using the `%run` IPython command to run the script, which then gets executed. The IPython session then has access to the last state of the variables from the script:

```
$ ipython
Python 3.5.2 |Continuum Analytics, Inc.| (default, Jul  5 2016, 11:41:13)
Type "copyright", "credits" or "license" for more information.

IPython 5.1.0 -- An enhanced Interactive Python.
?         -> Introduction and overview of IPython's features.
%quickref -> Quick reference.
help      -> Python's own help system.
object?   -> Details about 'object', use 'object??' for extra details.

In [1]: %run test.py
1

In [2]: a + 1
Out[2]: 2
```