welcome_to_python

April 26, 2020

1 Welcome to Python

You have two options for this class:

1. You can run the code online, using MyBinder. This means you only need your browser, and not necessarily a local Python installation. (At least not for now, which could be useful if lockdown is a limiting factor for you.)

2. You can install Python on your computer. If you're on Windows or OS X, I recommend the Anaconda distribution. It comes with a visual installer that will install basic Python, and a number of additional packages you will need for this class.

Both of these options are explained in further detail below. Choose one, and scroll down to the relevant section.

1.1 Welcome to MyBinder

This could say "Welcome to Jupyter Notebooks", as that's what you're really looking at. This is a tool for creating interactive worksheets that combine text (which you're reading right now) and code (which you'll see a bit later).

Jupyter notebooks can be rendered in different ways. You might be reading this as a PDF document, via GitHub, or in an interactive session running on your own computer. Alternatively, you might already be looking at an interactive session on MyBinder.

MyBinder is a website that can run Jupyter Notebooks within a browser. This is great, because it means that you can run Python code in your browser *without* having a local Python installation. If your computer doesn't allow you to install anything on it (e.g. because it's locked down by an IT department), this is the option for you.

The link to this worksheet on MyBinder is:
mybinder.org/v2/gh/esdalmaijer/Cambridge_SSRMP_Python/master

1.1.1 Using MyBinder

If you clicked on the link above, you should have been taken to a page with several folders. You're looking for the folder `00_welcome_to_python`, and in that folder the notebook `welcome_to_python` (this one!).

The Jupyter Notebook interface has this text and code snippets, and a bar on top with several options. The text bits are pretty simple: You can read them using your eyeballs, and/or text-to-voice or text-to-touch software on your computer. More interesting are the code bits.

Code can be run in several ways. You can hover over a code snippet, and an arrow-like symbol will appear to the left. Clicking on this makes the code run. Try it on the snippet below!

```
In [ ]: print("Test me!")
```

If all went well, you should now see the text "Test me!" directly below the code snippet.

Another way of running code is by using the "Run" button in the menu on the top of the page.

One important thing to keep in mind is that any variables in memory are created *as you run snippets*. This means that the order in which *you* run things is important, not the order in which snippets appear. For example, try running the two snippets below in order (top first, bottom second):

```
In [ ]: print(a)
```

```
In [ ]: a = 3
```

If you did it in the top-to-bottom order, the first snippet should have caused an error.

Now do it again, but run the `a = 3` snippet before the `print(a)` snippet.

If all went well, the `print(a)` snippet didn't result in an error now, because the variable it referenced was created by the `a = 3` snippet.

This might not feel like a problem NOW, but inevitably you'll run into a situation where you forgot to run a snippet, or are trying to run a previous snippet after running later snippets. This WILL result in issues.

If you run into a weird issue and you think it might be to do with the order in which snippets were run, the best thing is to simply restart the whole worksheet. You do this by clicking on the "Kernel" option in the menu, and then click "Restart and clear output". This will simply restart the Python session, and thus removes all potential weirdness that was built up in memory.

Finally, choosing "Kernel" and then "Restart and run all" will make all snippets in this worksheet run from the top.

1.2 Welcome to Anaconda

If you're choosing to run code on your own computer, your best bet is to download the Anaconda Python distribution. This is maintained by the company Continuum. It incorporates basic Python, and many additional packages that are useful in scientific contexts.

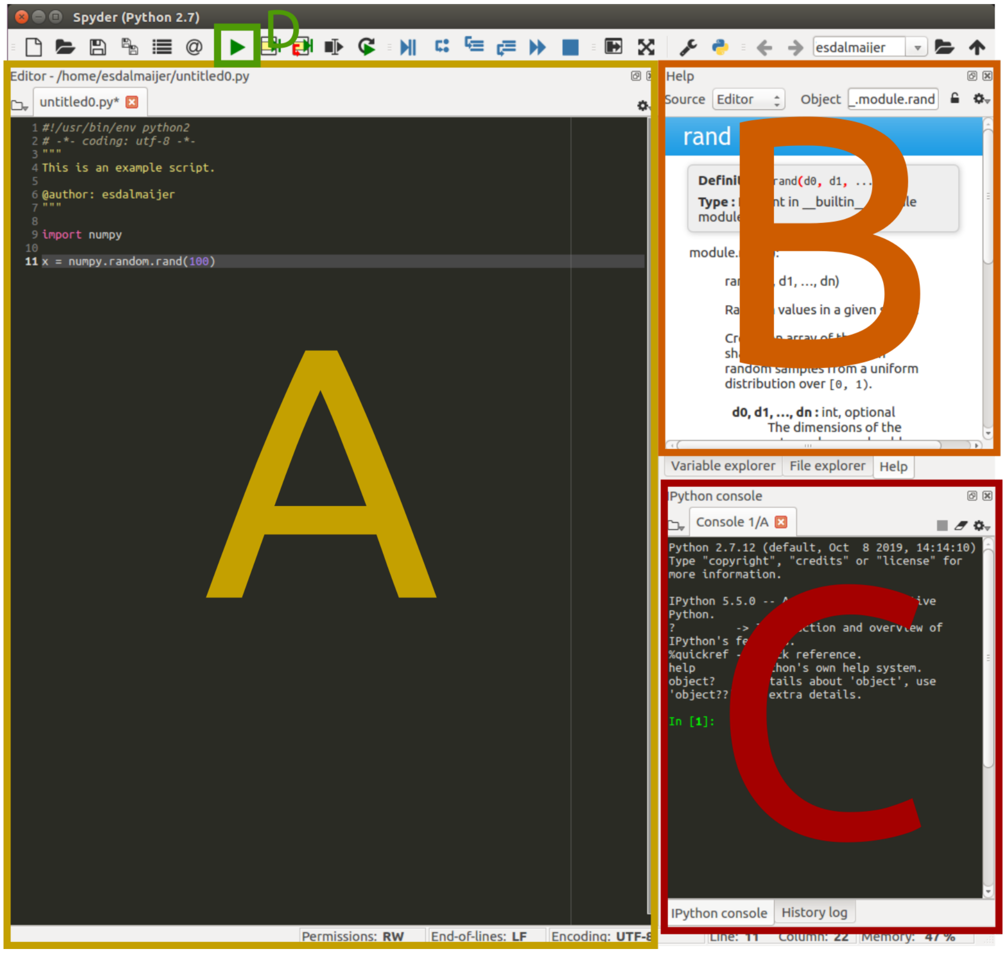
You can download this distribution here: www.anaconda.com/products/individual (make sure to get the Python 3.7 version!)

The download and installation procedure can take a while, so feel free to watch some cat videos while you're waiting.

1.2.1 Using Anaconda and Spyder

Anaconda comes with the Spyder code editor pre-installed. This looks as follows:

- Area A (yellow outline) is the code editor. This is where you will be writing script files. These are whole files that will not be run automatically, but only when you decide to run the whole file.
- Area B (orange outline) is the help window. If you're confused about a function in the script, hit Ctrl+I, and info will appear in this window.



```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
This is an example script.

@author: esdalmaijer
"""

import numpy

x = numpy.random.rand(100)
```

```
Python 2.7.12 (default, Oct  8 2019, 14:14:10)
Type "copyright", "credits" or "license" for
more information.

IPython 5.5.0 -- A                        tive
Python.
?          ->                ction and overview of
IPython's fe           .
%quickref -           k reference.
help                 thon's own help system.
object?              tails about 'object', use
'object??            extra details.

In [1]:
```

spyder screenshot

- Area C (red outline) is the **console**. This is an interactive Python session. If you type a line in here and hit Enter, it will run straight away. Restarting the console (by clicking on the little cross next to its name) will wipe the memory.
- Area D (green outline; all the way on the top) is the "Run" button. This will make the current script run in its entirety. It will be run line-by-line, using the active console. Any visible results will appear here too.

Spyder is a very decent editor, especially if you're used to Matlab or R Studio (both look very similar), or if you're completely new to coding. Code Editors are like text processors or slideshow software: There are several options, and each come with their own benefits and drawbacks. Like text or presentation files, script files are to construct a complete narrative to your computer.

For example, you could use the console as a quick calculator or to try out a few things. The code editor is for larger files, which could be entire analysis scripts, experiments, or a coherent set of functions for later use. Python scripts should be saved with the `.py` extension.