**[OnlineCourses](http://www.southampton.ac.uk/~fangohr/blog/installation-of-python-spyder-numpy-sympy-scipy-pytest-matplotlib-via-anaconda.html" \l "id4)**

<https://www.learnpython.org/>

<https://www.datacamp.com/courses/intro-to-python-for-data-science>

<https://www.datacamp.com/courses/intermediate-python-for-data-science>

<https://www.codecademy.com/learn/learn-python>

<https://matplotlib.org/tutorials/index.html>

Scikit-Learn, XGBoost, pandas, SciPy, Matploltib, CoreML

[**Python packages**](http://www.southampton.ac.uk/~fangohr/blog/installation-of-python-spyder-numpy-sympy-scipy-pytest-matplotlib-via-anaconda.html#id4)

The packages we generally need are

* numpy (NUMeric Python): matrices and linear algebra
* scipy (SCIentific Python): many numerical routines
* matplotlib: (PLOTting LIBrary) creating plots of data

We also need for our teaching:

* sympy (SYMbolic Python): symbolic computation
* pytest (Python TESTing): a code testing framework

## **Shortcuts for useful functions**

**Note**

The following are the default shortcuts; however, those marked with \* can be customized through the Keyboard shortcuts tab in the Preferences. Also, macOS users should substitute Command for Ctrl, and Option for Alt.

* F5\* executes the current file.
* F9\* executes the currently highlighted chunk of code; this is very useful to (say) update definitions of functions in the Console session without having to run the whole file again. If nothing is selected, F9\* executes the current line.
* Tab\* auto-completes commands, function names, variable names, and methods in the Console and the Editor. This feature is very useful, and should be employed routinely. Do try it now if auto-completion is new to you. Assume you have defined a variable:
* mylongvariablename = 42

Suppose we need to write code that computes mylongvariablename + 100. We can simply type my and then press the Tab key. The full variable name will be completed and inserted at the cursor position if the name is unique, and then we can carry on and type `` + 100``. If the name is not uniquely identifiable given the letters my, a list field will be displayed from which the desired variable can be chosen. Choosing from the list can be done with the Up and Down keys with the Enter key to select, or by typing more letters of the name in question (the selection will update automatically) and confirming by pressing Enter when the appropriate name is identified.

* Ctrl-Enter\* executes the current cell (menu entry Run ‣ Run cell). A cell is defined as the code between two lines which start with the characters #%%, # %% or # <codecell>.
* Shift-Enter\* executes the current cell and advances the cursor to the next cell (menu entry Run ‣ Run cell and advance).

Cells are useful to execute a large file/code segment in smaller units. (It is a little bit like a cell in an IPython notebook, in that chunks of code can be run independently).

* Alt-Up\* moves the current line up. If multiple lines are highlighted, they are moved up together. Alt-Down\* works correspondingly, moving line(s) down.
* Ctrl-LeftMouseButton or Alt-G\* on a function/method in the Editor opens a new Editor tab showing the definition of that function.
* Shift-Ctrl-Alt-M\* maximizes the current window (or changes the size back to normal if pressed in a maximized window).
* Ctrl-Shift-F\* activates the Find in Files pane, allowing grep-like searches across all files in a specified scope.
* Ctrl - = will increase the font size in the Editor or the Console, whereas Ctrl - - will decrease it.

The font face and size for other parts of the UI can be set under Preferences ‣ General ‣ Appearance ‣ Fonts.

* Ctrl-S\* in the Editor saves the file currently being edited. This also forces various warning triangles in the left column of the Editor to be updated (otherwise they update every 2.5 seconds by default, which is also configurable).
* Ctrl-S\* in the Console saves the current IPython session as an HTML file, including any figures that may be displayed inline. This is useful as a quick way of recording what has been done in a session.

(It is not currently possible to load this saved record back into the session -- if you need functionality like this, look for the IPython Notebook).

* Ctrl-I\* when pressed while the cursor is on an object opens documentation for that object in the help pane.

[**Matplotlib**](http://www.southampton.ac.uk/~fangohr/blog/installation-of-python-spyder-numpy-sympy-scipy-pytest-matplotlib-via-anaconda.html#id4)

<https://matplotlib.org/tutorials/index.html>