1. **Familiarizing Text Editor, IDE, Code Analysis Tools etc // Use any IDE**

**IDE**

An IDE (or Integrated Development Environment) is a program dedicated to software development. As the name implies, IDEs integrate several tools specifically designed for software development. These tools usually include:

* An editor designed to handle code (with, for example, syntax highlighting and auto-completion)
* Build, execution, and debugging tools
* Some form of source control

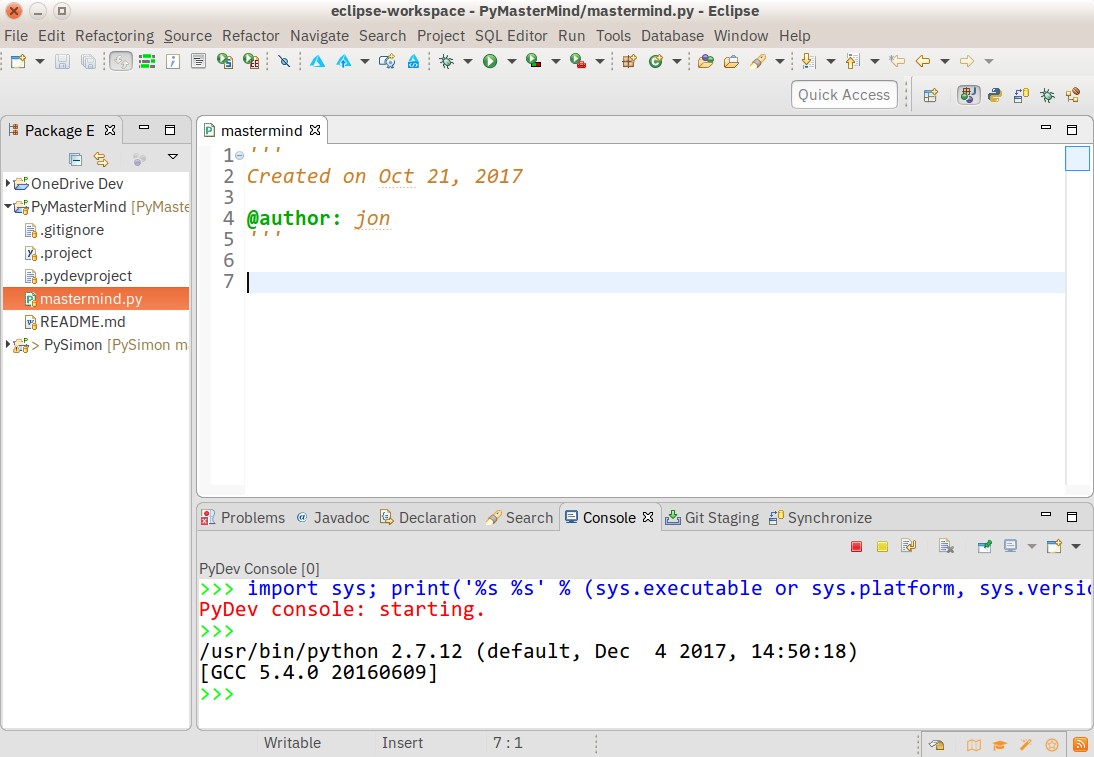
Most IDEs support many different programming languages and contain many more features. They can, therefore, be large and take time to download and install. You may also need advanced knowledge to use them properly.

In contrast, a dedicated code editor can be as simple as a text editor with syntax highlighting and code formatting capabilities. Most good code editors can execute code and control a [debugger](https://realpython.com/python-debugging-pdb/). The very best ones interact with source control systems as well. Compared to an IDE, a good dedicated code editor is usually smaller and quicker, but often less feature rich.

## **General Editors and IDEs with Python Support**

### **1.Eclipse + PyDev**

PyDev, which enables Python debugging, code completion, and an interactive Python console. Installing PyDev into Eclipse is easy: from Eclipse, select Help, Eclipse Marketplace, then search for PyDev. Click Install and restart Eclipse if necessary.

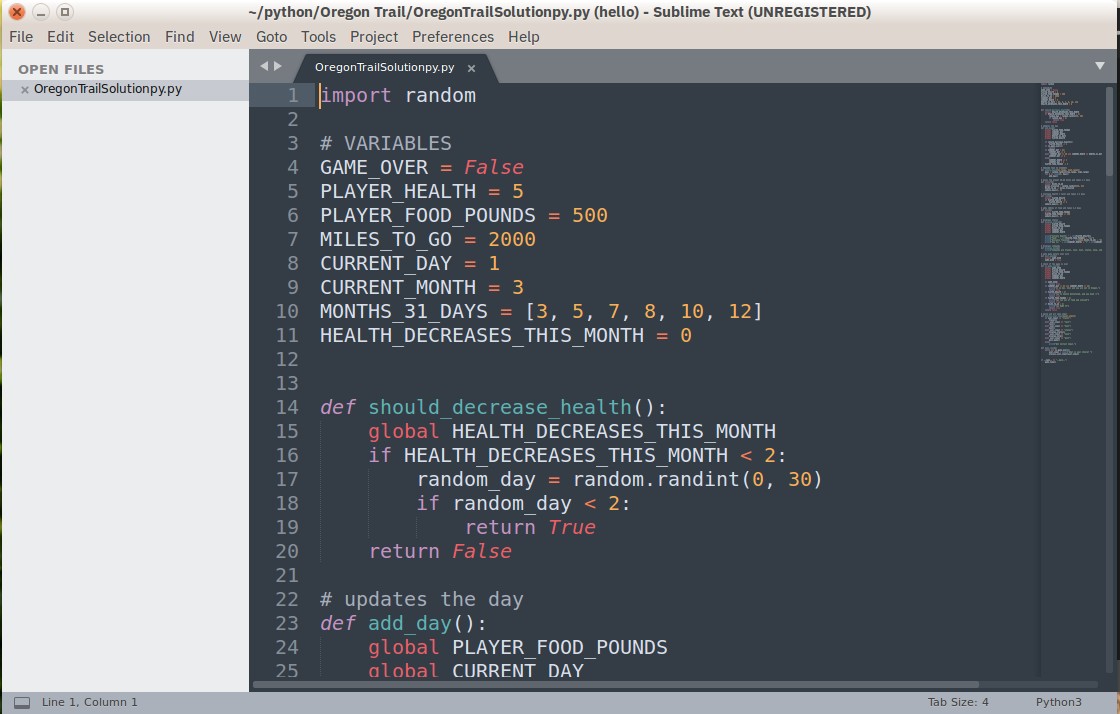
**Pros:** If you’ve already got Eclipse installed, adding PyDev will be quicker and easier. PyDev is very accessible for the experienced Eclipse developer.

**Cons:** If you’re just starting out with Python, or with software development in general, Eclipse can be a lot to handle. Remember when I said IDEs are larger and require more knowledge to use properly? Eclipse is all that and a bag of (micro)chips.

### **2.Sublime Text**

Written by a Google engineer with a dream for a better text editor, Sublime Text is an extremely popular code editor. Supported on all platforms, Sublime Text has built-in support for Python code editing and a rich set of extensions (called packages) that extend the syntax and editing features.

Installing additional [Python packages](https://realpython.com/python-modules-packages/) can be tricky: all Sublime Text packages are written in Python itself, and installing community packages often requires you to execute Python scripts directly in Sublime Text.

**Pros:** Sublime Text has a great following in the community. As a code editor, alone, Sublime Text is fast, small, and well supported.

**Cons:** Sublime Text is not free, although you can use the evaluation version for an indefinite period of time. Installing extensions can be tricky, and there’s no direct support for executing or debugging code from within the editor.

To make the most of your Sublime Text setup, read our [Python + Sublime Text setup guide](https://realpython.com/setting-up-sublime-text-3-for-full-stack-python-development/) and consider our [in-depth video course](https://realpython.com/products/sublime-python/) that shows you how to craft an effective Python development setup with Sublime Text 3.

### **3.Visual Studio**

Built by Microsoft, Visual Studio is a full-featured IDE, in many ways comparable to Eclipse. Built for Windows and Mac OS only, VS comes in both free (Community) and paid (Professional and Enterprise) versions. Visual Studio enables development for a variety of platforms and comes with its own marketplace for extensions.

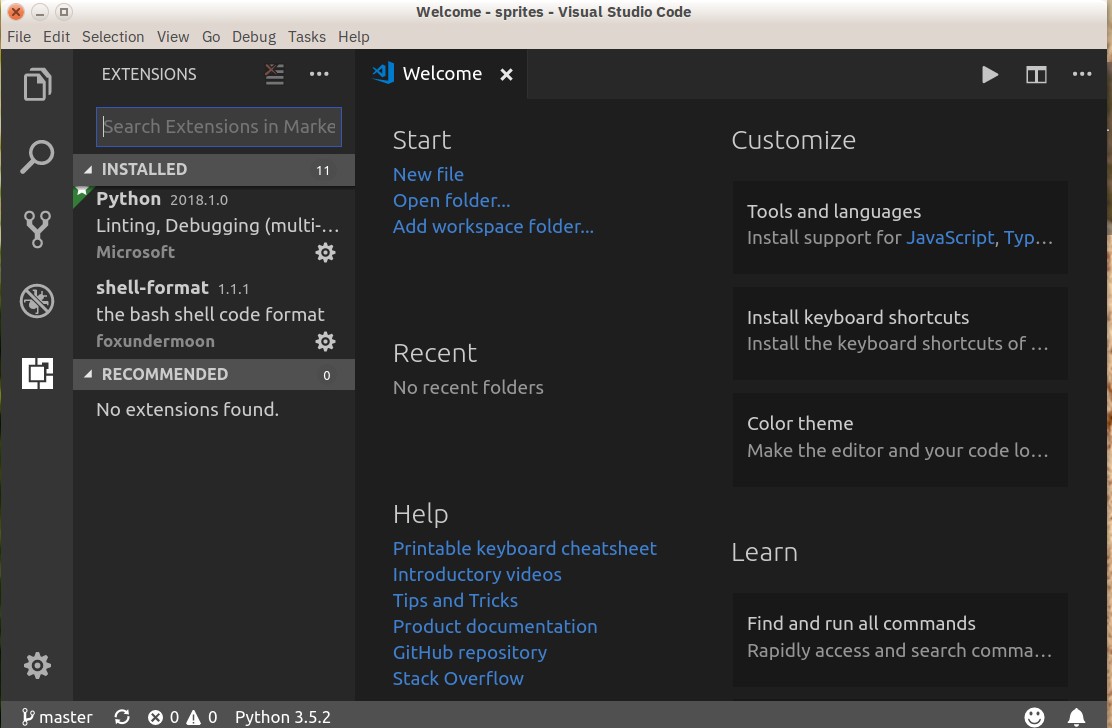
Python Tools for Visual Studio (aka PTVS) enables Python coding in Visual Studio, as well as Intellisense for Python, debugging, and other tools.

Not to be confused with full Visual Studio, Visual Studio Code (aka VS Code) is a full-featured code editor available for Linux, Mac OS X, and Windows platforms. Small and light-weight, but full-featured, VS Code is [open-source](https://github.com/Microsoft/vscode), [extensible](https://marketplace.visualstudio.com/vscode), and configurable for almost any task. Like Atom, VS Code is built on Electron, so it has the same advantages and disadvantages that brings.

Installing Python support in VS Code is very accessible: the Marketplace is a quick button click away. Search for Python, click Install, and restart if necessary. VS Code will recognize your Python installation and libraries automatically.

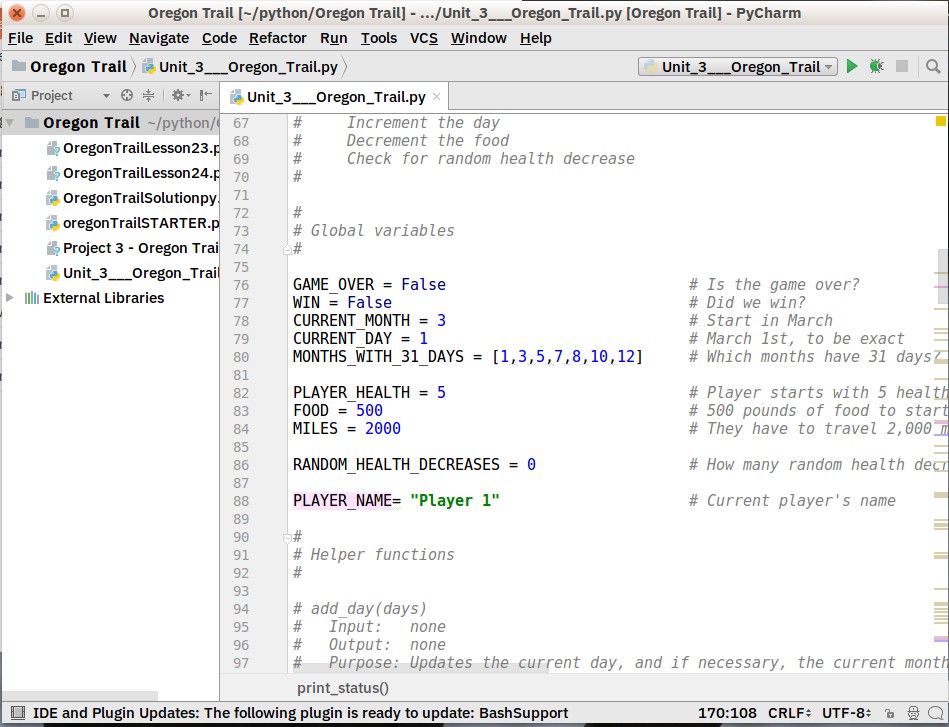
**Pros:** Thanks to Electron, VS Code is available on every platform, surprisingly full-featured despite having a small footprint, and open-source.

**Cons:** Electron means VS Code is not a native app. Plus, some people may have principled reasons to not use Microsoft resources.

**4.Pycharm**

One of the best (and only) full-featured, dedicated IDEs for Python is [PyCharm](https://realpython.com/pycharm-guide/). Available in both paid (Professional) and free open-source (Community) editions, PyCharm installs quickly and easily on Windows, Mac OS X, and Linux platforms.

Out of the box, PyCharm supports Python development directly. You can just open a new file and start writing code. You can run and debug Python directly inside PyCharm, and it has support for source control and projects.

**Pros:** It’s the de facto Python IDE environment, with tons of support and a supportive community. It edits, runs, and debugs Python out of the box.

**Cons:** PyCharm can be slow to load, and the default settings may need tweaking for existing projects.

### **5.Spyder**

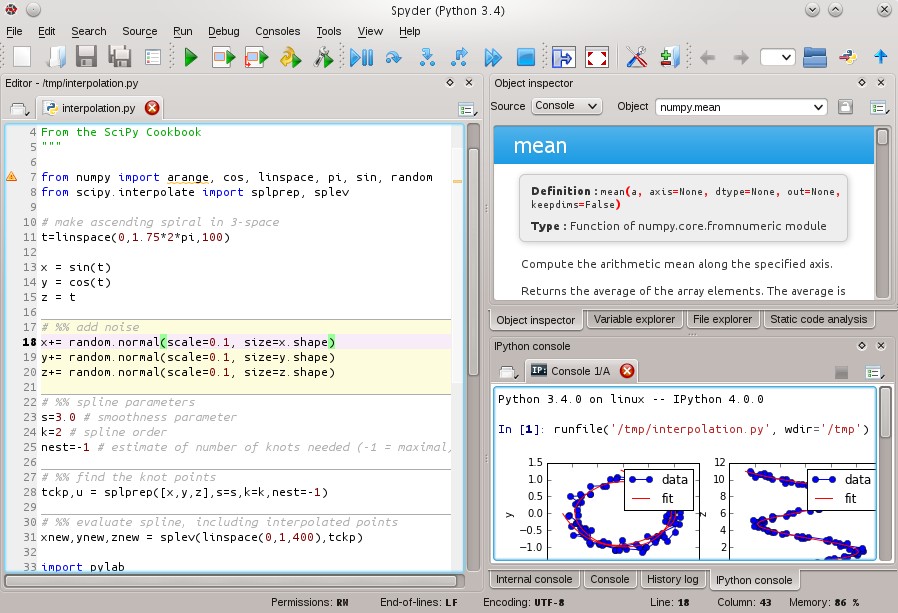
Spyder is an open-source Python IDE that’s optimized for data science workflows. Spyder comes included with the Anaconda package manager distribution, so depending on your setup you may already have it installed on your machine.

What’s interesting about Spyder is that it’s target audience is data scientists using Python. You’ll notice this throughout. For example, Spyder integrates well with common Python data science libraries like [SciPy](https://realpython.com/python-scipy-cluster-optimize/), [NumPy](https://realpython.com/tutorials/numpy/), and [Matplotlib](https://realpython.com/python-matplotlib-guide/).

Spyder features most of the “common IDE features” you might expect, such as a code editor with robust syntax highlighting, Python code completion, and even an integrated documentation browser.

A special feature that I haven’t seen in other Python editing environments is Spyder’s “variable explorer” that allows you to display data using a table-based layout right inside your IDE. Personally, I usually don’t have a need for this but it does look neat. If you regularly do data science work using Python, you might fall in love with this unique feature. The IPython/Jupyter integration is nice as well.

Overall, I’d say that Spyder feels more basic than other IDEs. I like to view it more as a special purpose tool rather than something I use as my primary editing environment every day. What is nice about this Python IDE is that it is available for free on Windows, macOS, and Linux and that it is fully open-source software.



**Pros:** You’re a data scientist using the Anaconda Python distribution.

**Cons:** More experienced Python developers might find Spyder too basic to work with on a daily basis and instead opt for a more complete IDE or customized editor solution.

***6.THONNY***

A recent addition to the Python IDE family, Thonny is billed as an IDE for beginners. Written and maintained by the [Institute of Computer Science](https://www.cs.ut.ee/) at the [University of Tartu](http://www.ut.ee/) in Estonia, Thonny is available for all major platforms, with installation instructions on the site.

By default, Thonny installs with its own bundled version of Python, so you don’t need to install anything else new. More experienced users may need to tweak this setting so already installed libraries are found and used.

**Pros:** You’re a beginning Python user, and want an IDE that’s ready to roll.

**Cons:** More experienced Python developers will find Thonny too basic for most uses, and the built-in interpreter is something to work around, not with. Plus, as a new tool, there may be issues you find which may not have immediate solutions.

If you’re interested in using Thonny as your Python editor, be sure to read our [dedicated article on Thonny](https://realpython.com/python-thonny/) which goes into more depth and shows you additional features.

