

Tech Tutorial : PyCharm

<https://www.jetbrains.com/pycharm/>

What is PyCharm?

PyCharm is an Integrated Developer Environment (IDE) focused on software development in Python. It is developed by JetBrains, a company that also produces an IDE for Java (IntelliJ) which has been extended for Android app development (Android Studio). PyCharm is designed for Python, and incorporates a large number of related development tools under the same roof.

Why use PyCharm?

PyCharm contains many features that make writing code easier and more enjoyable. It provides automatic notification of obvious syntax errors, as well as possible errors at runtime. It provides notifications for violations of correct style, which leads to better knowledge of style guides and more understandable and regular code. It provides autocompletion for code. These features and the entire environment are highly customizable, which may make it more difficult to understand at first but provide a very powerful environment. Perhaps most importantly, it allows for easy installation of Python packages and integrates easily with many other development tools like Jupyter, Anaconda, web development tools and more. This makes it possible to manage large projects that are Python based but also require outside development tools.

All of these features are great, but many different IDEs have similar capabilities. Why would one prefer PyCharm over other environments? One main benefit of PyCharm is that it is dedicated to Python. It is designed explicitly for work in Python, and the development team focuses on creating Python specific features. Another reason commonly cited is that Python integrates seamlessly with version control software such as GitHub and SVN. PyCharm manages VCS repositories seamlessly within its own environment, giving full capabilities without needing to resort to outside resources. This makes for a fluid integration of coding and managing a project.

In summary, PyCharm provides many useful features:

- Code correction
- Suggestions for better style
- Autocompletion
- Integration of Jupyter/IPython and other tools
- Easy package management
- Highly customizable
- Integrated of version control systems
- Designed for Python, with strong developer community

- Fun color themes

And while many of these are available elsewhere, PyCharm packs them all into one tool.

Installation

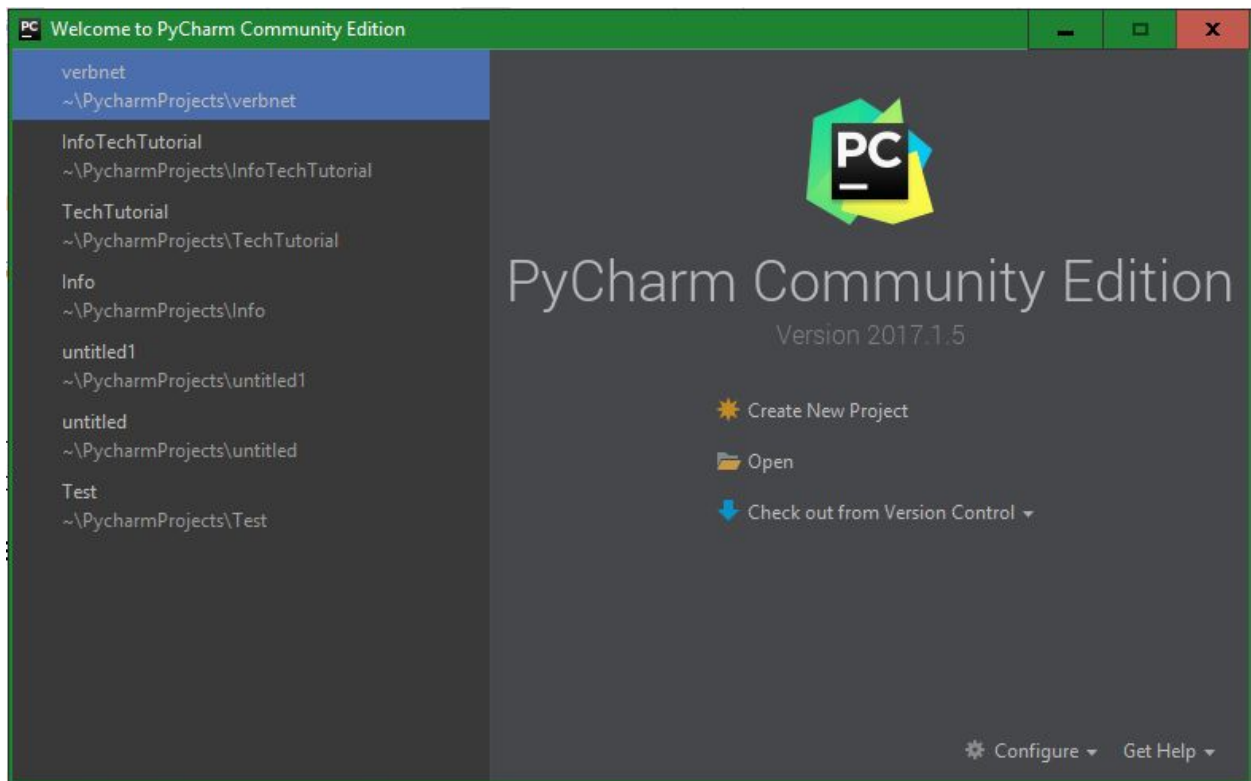
PyCharm is available for Windows, macOS, and Linux. Installation files and instructions for each can be found at:

<https://www.jetbrains.com/pycharm/download/#section=windows>

JetBrains provides a free “Community” edition as well as a “Professional” version. I have no experience with the Professional version; the Community version works great for all the development I’ve done.

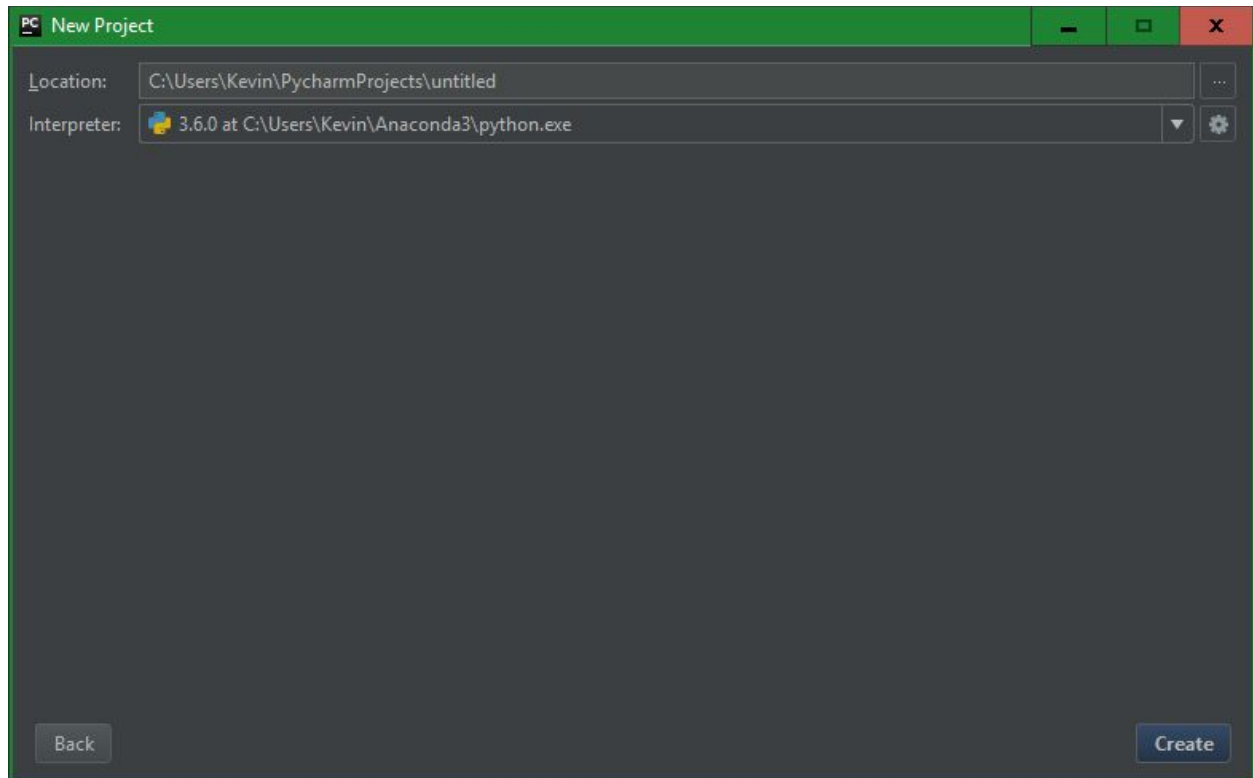
Basic Operations

New projects



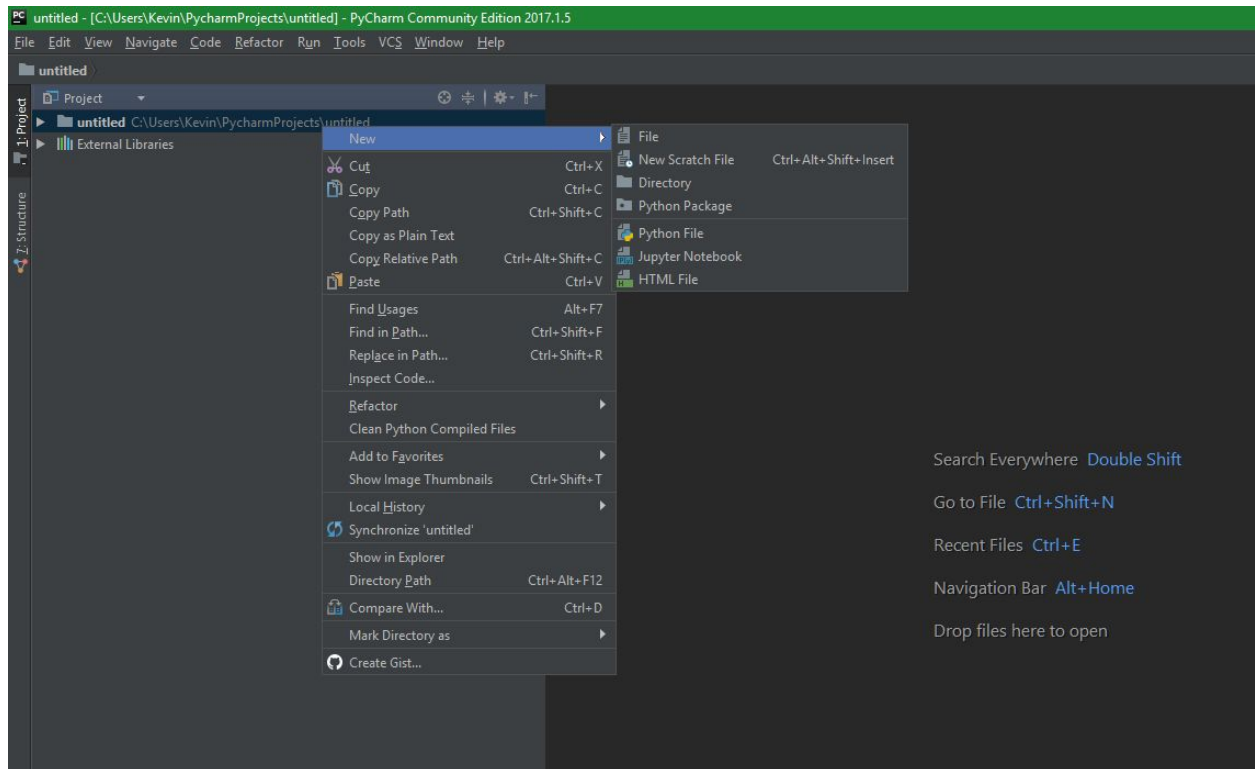
Run PyCharm, and you are given a couple of options. You can Create a New Project, Open (an existing project), or check something out from Version Control. If you

Create a New Project, you can use VCS later to check something out, or even add your own project as a VCS Repository. To get started, click Create New Project.

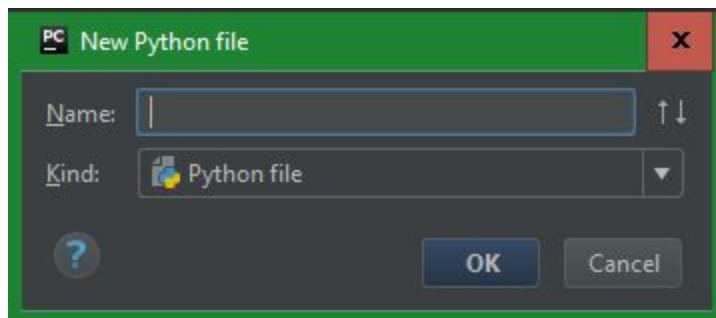


From here, you can select the location of your project. It will also allow you to select which Python interpreter you'd like to use. If you have multiple versions of Python installed, you can select which one you'd like. You can later switch between interpreters for any project.

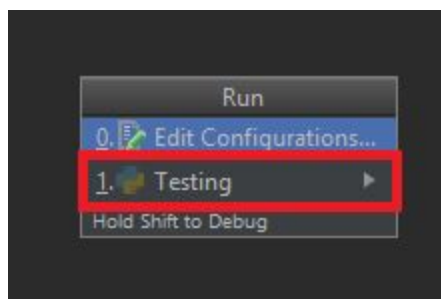
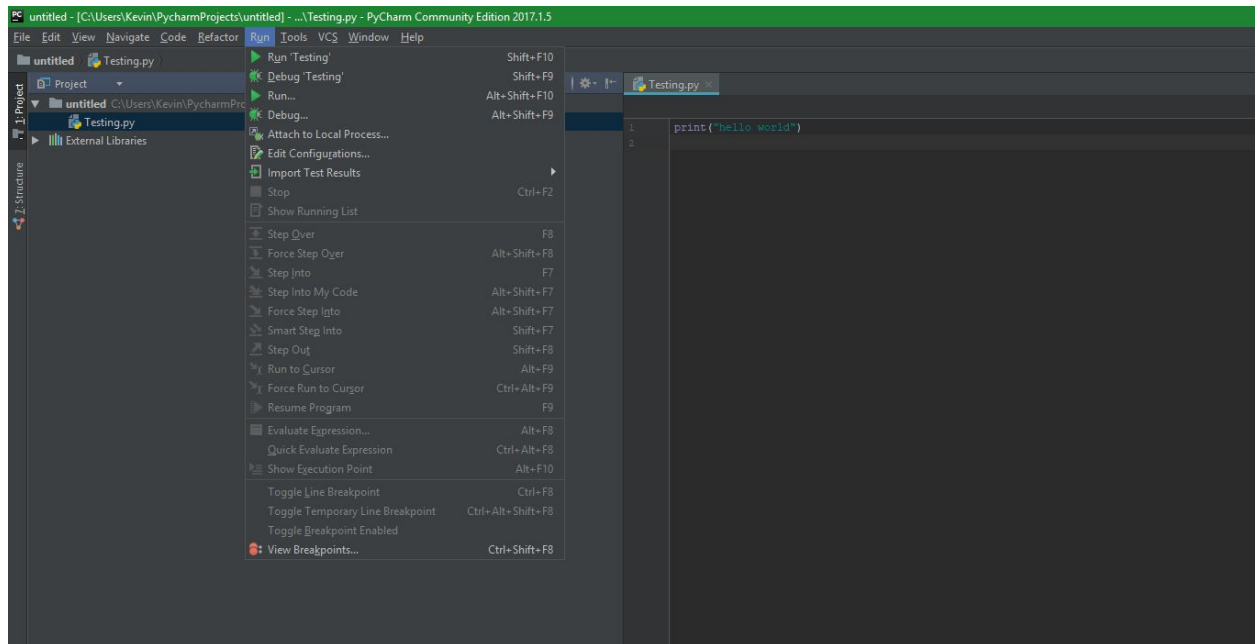
Creating this project puts you into the PyCharm environment with a new Python project.



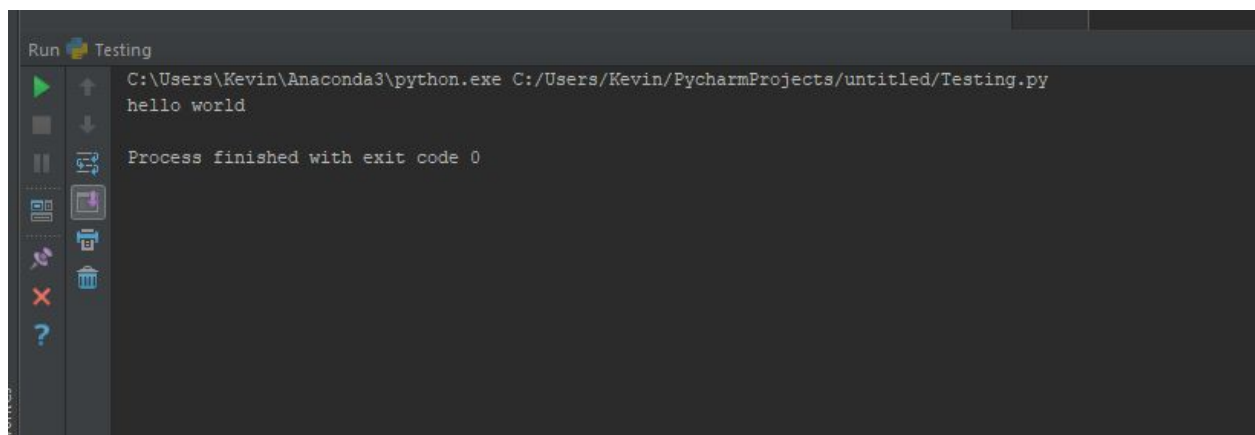
To get started, right click on your project folder (here “untitled”), and select New. You then have the option to create many kinds of new files, including blank files, Python files Jupyter Notebooks, and HTML files. For now, let’s create a new Python file.



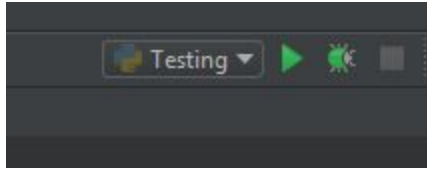
After creating your file, you can then write some python code, and run the interpreter. This is done by selecting Run -> Run (or alt+shift+f10), and the choosing which Python file you’d like to run.



This executes the Python code, outputting the result to the terminal below.



Once you've run a Python file, you can easily re-run using the green arrow in the upper right. The box stores all the recent runs you've done, so you can easily go back and select which file you'd like to run.

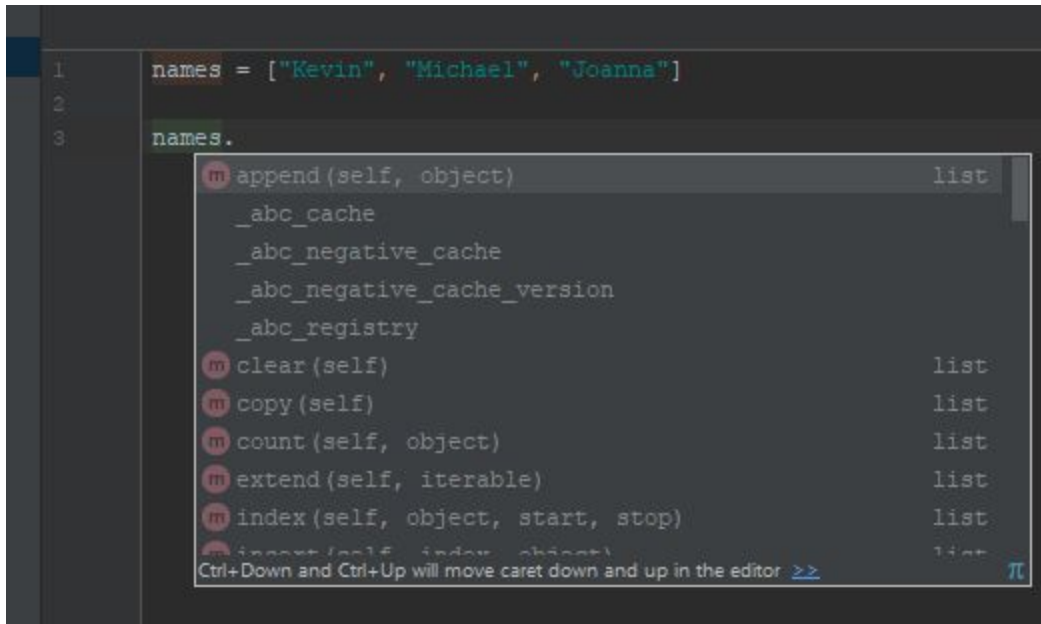


Code completion

One of the most exciting features to me about advanced IDEs is code completion. To see how this works, create a new list in python:

```
1 names = ["Kevin", "Michael", "Joanna"]
2
3
```

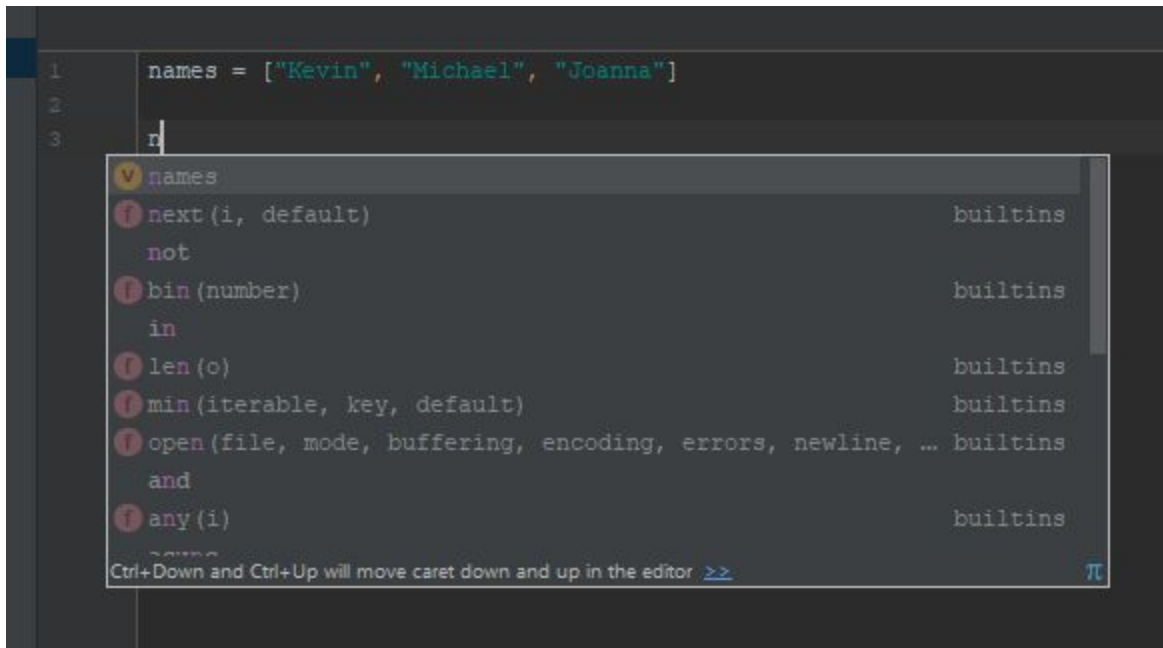
From here, type the name of the list, followed by a ".", indicating you're going to call a method on the object.



PyCharm will automatically bring up all the possible completions it can find. It knows the object is a list, so you can now see all the things a list can do, including append, clear, copy, count, etc. It also shows the arguments for these methods. This is helpful if you forget the exact parameters for a method.¹

This is just one practical example of code completion. PyCharm will do this whenever you type anything, even just variable names:

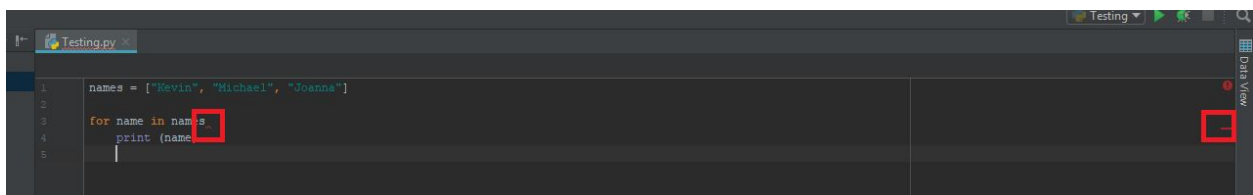
¹ It can also be dangerous, if you start using methods you don't understand because they look easy in the autocomplete. This doesn't happen frequently in my experience, but it is something to be aware of.



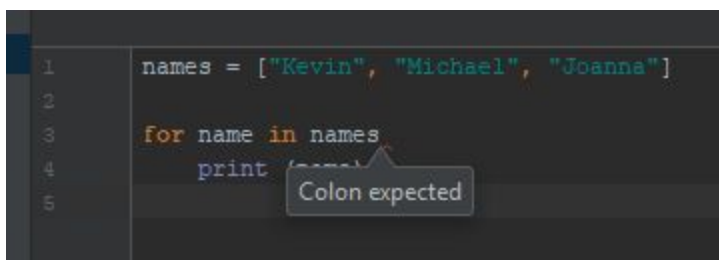
This shows all the things it knows that start with 'n': the variable you've declared, the function "next" and the keyword "not", as well as anything with an 'n' in it. Again, these are helpful to find things, but can be dangerous: make sure you know what things do before automatically using them.

Syntax Checking/Style Suggestions

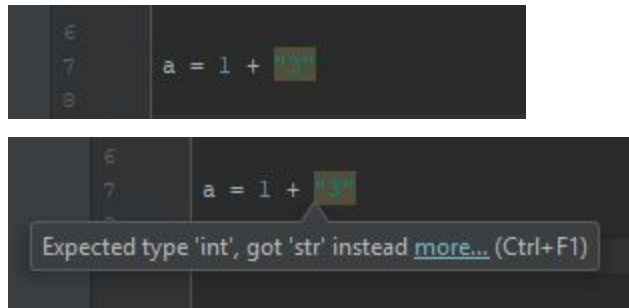
By default, PyCharm automatically checks your syntax for errors, which is incredibly valuable for debugging. Consider a simple for loop, with the ":" missing. PyCharm will highlight the error as well as the line containing it all before running the code.



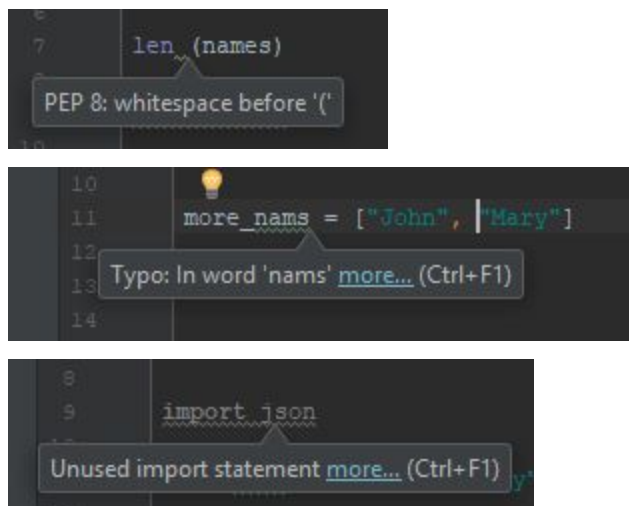
If you mouseover either of the red areas (the red squiggle or the line number marking on the right) it will tell you what the syntax error is.



Some errors it can find easily, while others are more difficult. Python does a lot of things at 'runtime', and these are harder for the IDE to find automatically. However, PyCharm still tries. For instance, if you try to add two things of different types together it will fail. Python doesn't know what type they are until the program is run, but PyCharm attempts to figure it out and gives a strong warning, indicating that it thinks the types are wrong.

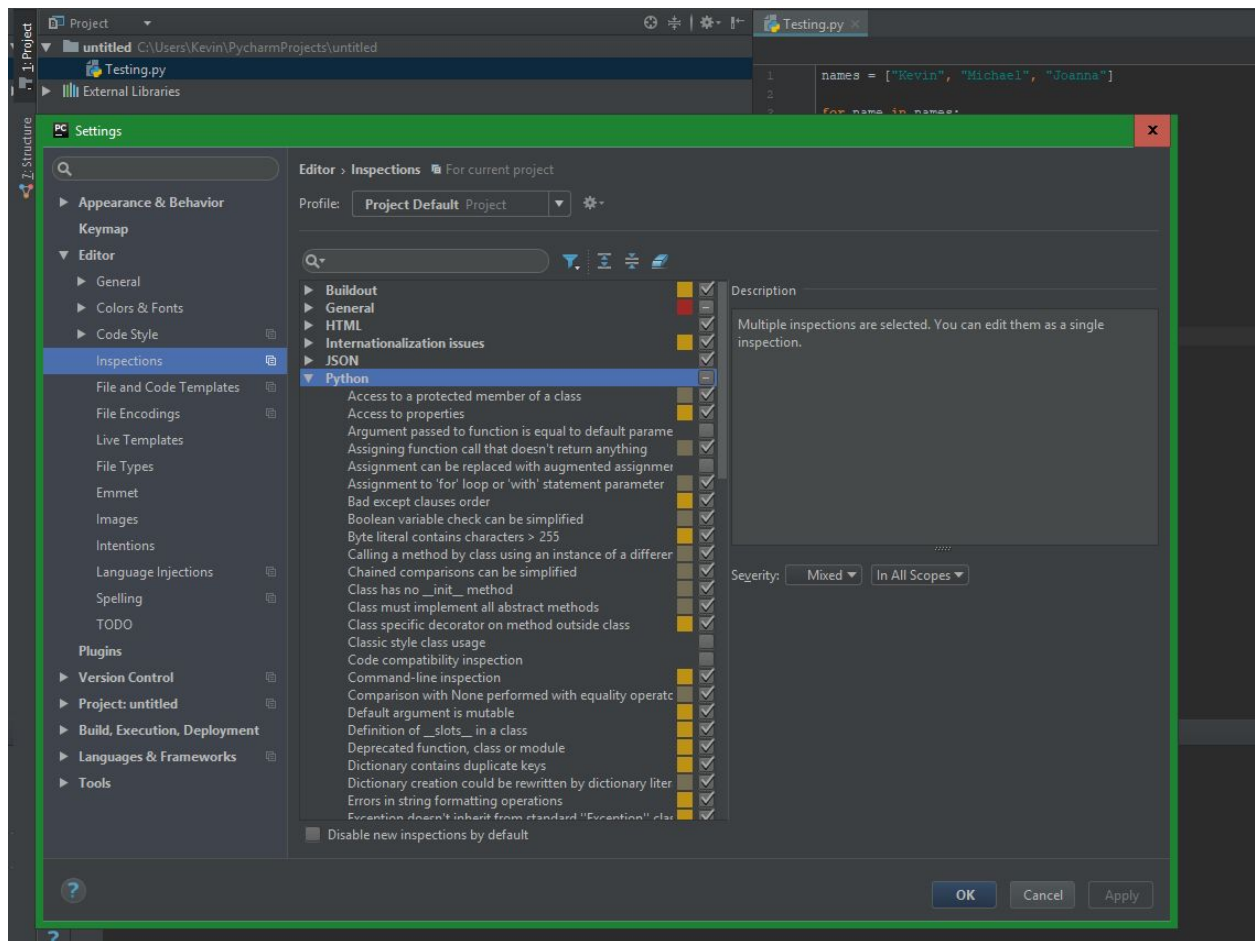


PyCharm has many more code warnings, intended to promote writing good code. It has a automatic checker for the Python style guide PEP 8, and will indicate if you're writing things that don't look quite write. These are soft warnings, really just reminders of what good Python is intended to be. I personally find these very valuable in writing code that other people will use, and improving my knowledge of how good Python is written. It also has a spell checker, which is sometimes annoying if you want to name a variable something that's not a word, as well as a system for checking unused code which reduces clutter.



All of these checkers can be useful but also annoying, and fortunately they are all entirely customizable. To change the warnings you receive, simply go to File -> Settings, and then under the Editor tab select Inspections. This shows all the possible

things the IDE can highlight. Navigating to Python will allow you to customize the code warnings you receive, as well as browsing possible errors.



PyCharm has many other fun features, and it integrates well with lots of software. Here are some quick instructions for other interesting options:

Load a GitHub repository

- VCS > Checkout from Version Control -> GitHub
- Select repository, local parent directory and name

Create a new Python/Jupyter Notebook file

This requires installation of Anaconda and Jupyter : for more see the [Jupyter Integration Help](#)

- Right click on a project

- New > Python File/Jupyter Notebook

Inspect your project

Click the "Structure" tab on the far right

This shows all the classes, functions, and variables associated with the current project.

Change Color Scheme

File > Settings > Editor > Colors & Fonts

Change Python Version

File > Settings > Project: (Your Project) > Python Interpreter

Other Resources

The best resource I've found is PyCharm's website, which includes in-depth tutorials for many projects and tasks.

<https://www.jetbrains.com/pycharm/>

JetBrains also host a PyCharm centered blog:

<https://blog.jetbrains.com/pycharm/>

It should also be noted that the PyCharm community is large and helpful - searches for errors and tasks are likely to return positive results.

If you're not sold, here's a good overview of other IDEs:

<https://realpython.com/python-ides-code-editors-guide/>