

## Pycharm Setup

In today's video, I'm going to show you how to install PyCharm and connect that to our Neon SDK. If you haven't already installed your Neon SDK, I'll link the video for that below. Let's get started.

### Installing PyCharm

Alright, we're going to continue our Neon setup now by installing PyCharm. PyCharm is an IDE or integrated development environment used for Python programs. So we're going to open up the terminal here. They have the command written out on their website, so it's going to be a `sudo snap install PyCharm community`, with the classic option. The community version is free you can also pay for the professional version which has some more features, but the free one should do just fine. Type in our password there. So, PyCharm is a snap package. Snap packages are kind of cool, they basically run on any flavor of Linux. It's the type of package that the Ubuntu store for example uses, and you can find out more in their website [snapcraft.io](https://snapcraft.io).

### Pycharm Initial Setup

So it looks like PyCharm has finished installing here, go ahead and start it up. I'm going to add that to my favorites, we're going to be using that a bunch. Nothing to import. I prefer the dark theme, but you can select light if you wish. And then we're also going to install our markdown and bash support packages. Markdown is what we're going to use to look at readme files for skills and core, and the bash support just gives us some helpful tools for reading all of our bash functions. Go ahead and start using pycharm.

### Connecting to the Neon SDK

And then we already have Neon installed, so we're going to open that just as the NeonAI folder, perfect. It can take a little while to index all the files, but you can start using it before it completes, it just might be a little bit slower, since it's working in the background. First thing I'm going to do here is go to settings and then under version control, and git. Excuse me under, version control, not under git. We're going to look at our git roots here. We want to make sure that we have this `/home/neon/NeonAI` registered as well as our `data/skills` which it'll show is under `/home/neon/NeonAI` so that's our core and our skills repositories. I've seen before where skills isn't automatically populated, so just to show you what you would do in that case, I'm going to remove it, and then manually add it. What you would do is simply browse back to it, hit OK, it automatically finds that `.git` directory and registers it as a git VCS (version control system). Okay, and that's done.

### Custom PyCharm VM Options

Another useful thing to look at in PyCharm is if you go under help and edit custom vm options create that file and this `Xms` and `Xms`, this is the amount of memory that it will allocate to the virtual machine it runs everything in. If you do like to do a lot of debugging from the python console or the terminal down

here, you should probably turn this up. Something like 2 gigs of ram would be plenty, just make sure you leave enough for PyCharm and your system to both run smoothly. Since I'm running this in a virtual machine with limited memory as it is, I'm going to bring that back down to 1024. Close that out.

### Testing Git Settings

So it looks like it's done indexing all of our files. I'm going to do just a quick check to make sure that everything is linked up and working as we'd expect, so I'm just gonna go into skills, make a new file and just call it test.txt, add it, and this green file name indicates that it's a new file that's not on the remote, as we expect because I just made it. So looks like git is picking up our repository settings. Go ahead and delete that, because we don't need it.

### Checking Git Ignore Settings

And I will also take a look here at our git ignore file, just to show what's in here. So as you make changes to the core, anything in any of these directories is going to be— anything that matches anything in this file is not going to commit to git. So importantly, things like your yml files aren't going to get committed, which is good because those are going to vary depending on the machine you're on. Same with anything in the documents, pictures, video, music. If we look at those, those are actually shortcuts to your user folder directories, so we wouldn't want to commit that to our project. Likewise, our default location for skills and caching, this data folder is also in there. So if you do make changes to git ignore, make sure you don't change those unless you really know that you want to, because you probably don't. If we look at the skills folder, we have the same git ignore and this one it ignores the just the .idea, .pyc, and then some other directories that we don't want to sync up with our repository.

### Summary

So that's a quick overview of installing and starting to work with PyCharm, and Neon AI. In the next videos we'll start looking at actually making some changes and start looking at the actual code. Thanks for watching this Neon AI tutorial. Be sure to head over to [neongecko.com](http://neongecko.com) for more information, including a written transcript of this tutorial and any code snippets we may have used. See you next time.