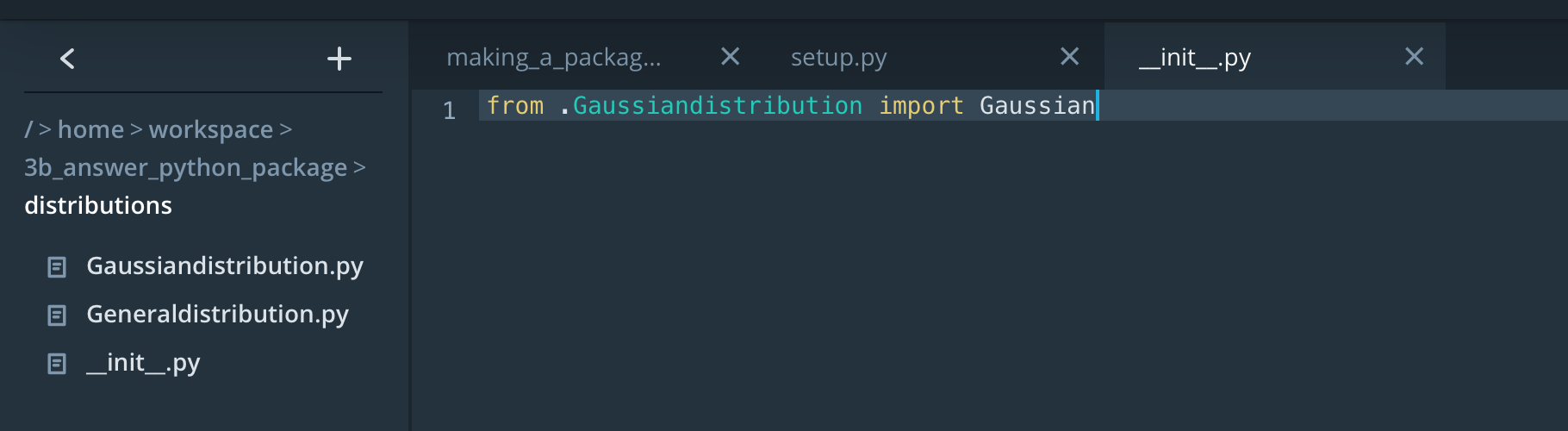
Making a Package and Pip Installing

You can put your code into the 3a\_python\_package folder in the workspace. Inside the 3a\_python\_package folder, you'll need to create a few folders and files:

\* a setup.py file, which is required in order to use pip install

\* a folder called 'distributions', which is the name of the Python package

\* inside the 'distributions' folder, you'll need the Gaussiandistribution.py file, Generaldistribution.py and an \_\_init\_\_.py file.



from setuptools import setup

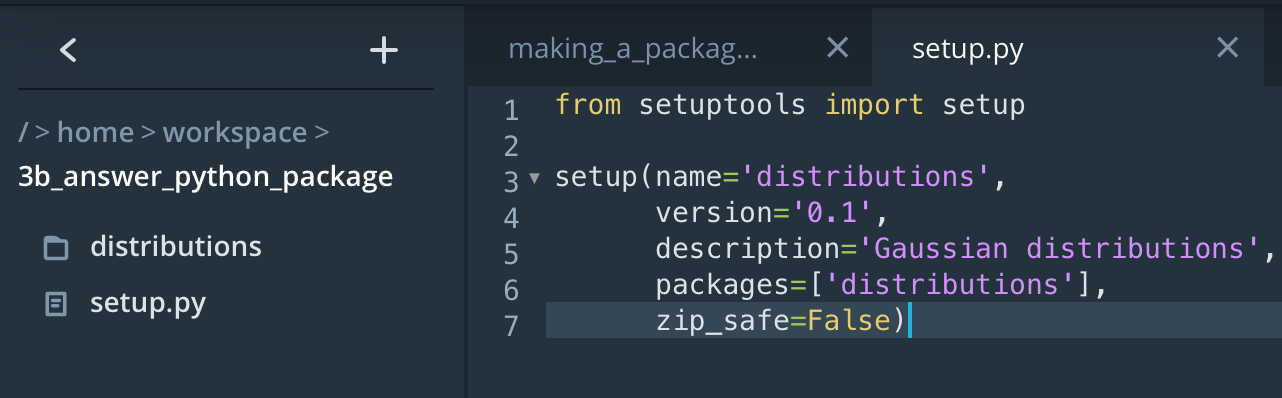
setup(name='distributions',

version='0.1',

description='Gaussian distributions',

packages=['distributions'],

zip\_safe=False)



Before you upload your code to PyPi, you should first pip install the package locally to make sure everything works as expected. The "Making a Package" and "Virtual Environments" lesson concepts should be helpful. You should also consider writing unit tests to test the functionality of your package. In the object-oriented programming lesson workspace, there were unit tests inside the 4a\_binomial\_package folder that you can use to help you get started. Those were in a file called test.py.

The object-oriented programming lesson included a complete, working package called dsnd-probability. We encourage you to code a project from scratch; however, if you get stuck, use the dsnd-probability package code as a template. It already contains all of the necessary files you'll need for creating a package. It's also a simple example of object-oriented code. You can use these files, including the setup.py and setup.cfg files, to help structure your own code.

Those files are located in the "Exercise: Upload to PyPi" section inside the "5\_exercise\_upload\_to\_pypi" folder.

Besides the lesson on object-oriented programming, you might find [this package building summary guide from the Python website](https://packaging.python.org/guides/distributing-packages-using-setuptools/) helpful.

For a much more detailed explanation of distributing Python packages, check out the documentation on Distutils.

1. [Introduction](https://docs.python.org/3/distutils/introduction.html)
2. [setup.py script](https://docs.python.org/3/distutils/setupscript.html)
3. [config file](https://docs.python.org/3/distutils/configfile.html)
4. [source distributions](https://docs.python.org/3/distutils/sourcedist.html)
5. [built distributions](https://docs.python.org/3/distutils/builtdist.html)
6. [uploading to PyPi](https://docs.python.org/3/distutils/packageindex.html)

If you are having trouble uploading your package to PyPi, here are a few issues that other students have faced.

# Troubleshooting Possible Errors

# **PyPi vs. Test PyPi**

Note that [pypi.org](https://pypi.org/) and [test.pypy.org](https://test.pypi.org/) are two different websites. You'll need to register separately at each website. If you only register at [pypi.org](https://pypi.org/), you will not be able to upload to the [test.pypy.org](https://test.pypi.org/)repository.

# **Unique Name**

Remember that your package name must be unique. If you use a package name that is already taken, you will get an error when trying to upload the package.

# **Re-uploading and Versioning**

Once you upload your package to PyPi, you cannot upload the same version again. All that means is that you need to go into your setup.py file and change the version number. For example, if you uploaded a package with version = 0.1.1, then you'll need to change this to something else like version = 0.1.2.

If you've ever wondered what all these version numbers mean, here is a tutorial about [semantic versioning](https://semver.org/). Semantic versioning is a standard for deciding when to move up to the next number in your versioning scheme.