

Writing Your Setup Script

The Distutils setup script is a Python script. In theory, it can do anything Python can do. In practice, it should do as little as possible, in as standard a way as possible. Setup scripts should be boring. The more exotic your installation process is, the more exotic your bug reports will be.

The first line of every Distutils setup script is always the same:

```
from distutils.core import setup
```



This imports the `setup()` function, which is the main entry point into Distutils. 95% of all Distutils setup scripts consist of a single call to `setup()` and nothing else. (I totally just made up that statistic, but if your Distutils setup script is doing more than calling the Distutils `setup()` function, you should have a good reason. Do you have a good reason? I didn't think so.)

The `setup()` function [can take dozens of parameters](#). For the sanity of everyone involved, you must use [named arguments](#) for every parameter. This is not merely a convention; it's a hard requirement. Your setup script will crash if you try to call the `setup()` function with non-named arguments.

The following named arguments are required:

- **name**, the name of the package.
- **version**, the version number of the package.
- **author**, your full name.
- **author_email**, your email address.
- **url**, the home page of your project. This can be your [PyPI](#) package page if you don't have a separate project website.

Although not required, I recommend that you also include the following in your setup script:

- **description**, a one-line summary of the project.
- **long_description**, a multi-line string in [reStructuredText](#) format. PyPI converts this to html and displays it on your package page.
- **classifiers**, a list of specially-formatted strings described in the next section.

Setup script metadata is defined in [PEP 314](#).

Now let's look at the `chardet` setup script. It has all of these required and recommended parameters, plus one I haven't mentioned yet: `packages`.

```
from distutils.core import setup
setup(
    name = 'chardet',
    packages = ['chardet'],
    version = '1.0.2',
    description = 'Universal encoding detector',
    author='Mark Pilgrim',
    #...
)
```

The `packages` parameter highlights an unfortunate vocabulary overlap in the distribution process. We've been talking about the "package" as the thing you're building (and potentially listing in The Python "Package" Index). But that's not what this `packages` parameter refers to. It refers to the fact that the `chardet` module is a multi-file module, sometimes known as... a "package." The `packages` parameter tells Distutils to include the `chardet/` directory, its `__init__.py` file, and all the other `.py` files that constitute the `chardet` module. That's kind of important; all this happy talk about documentation and metadata is irrelevant if you forget to include the actual code!