Virtual environments

Use a virtual environment to manage the dependencies for your project, both in development and in production.

What problem does a virtual environment solve? The more Python projects you have, the more likely it is that you need to work with different versions of Python libraries, or even Python itself. Newer versions of libraries for one project can break compatibility in another project.

Virtual environments are independent groups of Python libraries, one for each project. Packages installed for one project will not affect other projects or the operating system's packages.

Python 3 comes bundled with the **venv** module to create virtual environments. If you're using a modern version of Python, you can continue on to the next section.

If you're using Python 2, see Install virtualenv first.

Create an environment

Create a project folder and a `venv` folder within:

```
$ mkdir myproject
$ cd myproject
$ python3 -m venv venv
```

On Windows:

```
$ py -3 -m venv venv
```

If you needed to install virtualenv because you are using Python 2, use the following command instead:

```
$ python2 -m virtualenv venv
```

On Windows:

```
> \Python27\Scripts\virtualenv.exe venv
```

Activate the environment

Before you work on your project, activate the corresponding environment:

```
$ . venv/bin/activate
```

On Windows:

```
> venv\Scripts\activate
```

Your shell prompt will change to show the name of the activated environment.

Install Flask

Within the activated environment, use the following command to install Flask:

```
$ pip install Flask
```

Flask is now installed. Check out the Quickstart or go to the Documentation Overview.

Living on the edge

If you want to work with the latest Flask code before it's released, install or update the code from the master branch:

```
$ pip install -U https://github.com/pallets/flask/archive/master.tar.gz
```

Install virtualenv

If you are using Python 2, the venv module is not available. Instead, install virtualenv.

On Linux, virtualenv is provided by your package manager:

```
# Debian, Ubuntu
$ sudo apt-get install python-virtualenv

# CentOS, Fedora
$ sudo yum install python-virtualenv

# Arch
$ sudo pacman -S python-virtualenv
```

If you are on Mac OS X or Windows, download get-pip.py, then:

```
$ sudo python2 Downloads/get-pip.py
$ sudo python2 -m pip install virtualenv
```

On Windows, as an administrator:

```
> \Python27\python.exe Downloads\get-pip.py
> \Python27\python.exe -m pip install virtualenv
```