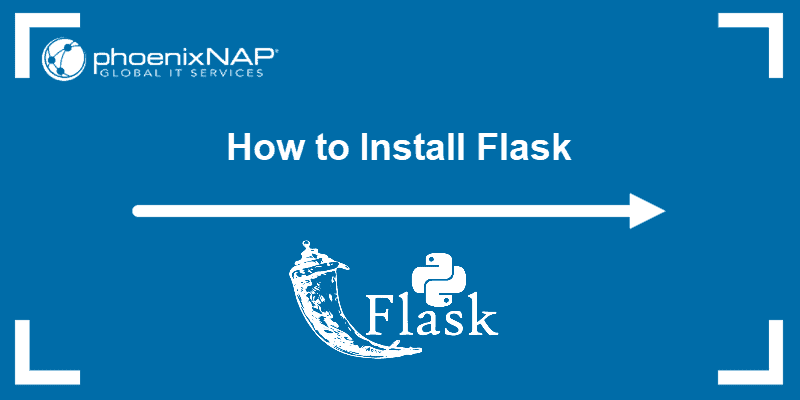
# How To Install Flask

**Introduction**

Flask is one of the most popular web application frameworks written in Python. It is a microframework designed for an easy and quick start. Extending with tools and libraries adds more functionality to Flask for more complex projects.

**This article explains how to install Flask in a virtual testing environment and create a simple Flask application.**



**Prerequisites**

* Installed Python 2.7 or Python 3.5 and newer
* CLI with administrator privileges

**Note:** Python 2 has reached the end-of-life maintenance status. It officially no longer has support as of 2020. Follow one of our guides on installing Python 3: [How to install Python 3 on CentOS 7](https://phoenixnap.com/kb/how-to-install-python-3-centos-7), [How to install Python 3 on CentOS 8](https://phoenixnap.com/kb/install-python-on-centos-8), [How to install Python 3 on Ubuntu](https://phoenixnap.com/kb/how-to-install-python-3-ubuntu), [How to install Python on Windows](https://phoenixnap.com/kb/how-to-install-python-3-windows).

**Step 1: Install Virtual Environment**

Install Flask in a virtual environment to avoid problems with conflicting libraries. [Check Python version](https://phoenixnap.com/kb/check-python-version) before starting:

* Python 3 comes with a virtual environment module called *venv*preinstalled. **If you have Python 3 installed, skip to Step 2.**
* Python 2 users must install the *virtualenv* module. **If you have Python 2, follow the instructions outlined in Step 1.**

**Install virtualenv on Linux**

The package managers on Linux provides *virtualenv*.

* **For Debian/Ubuntu:**

1. Start by opening the Linux terminal.

2. Use **apt**to install *virtualenv*on Debian, Ubuntu and other related distributions:

sudo apt install python-virtualenv

* **For CentOS/Fedora/Red Hat:**

1. Open the Linux terminal.

2. Use **yum**to install *virtualenv*on CentOS, Red Hat, Fedora and related distributions:

sudo yum install python-virtualenv

**Install virtualenv on MacOS**

1. Open the terminal.

2. Install *virtualenv* on Mac using **pip**:

sudo python2 -m pip install virtualenv

**Install virtualenv on Windows**

1. Open the command line with administrator privileges.

2. Use **pip** to install *virtualenv*on Windows:

py -2 -m pip install virtualenv

**Note:** To install pip on Windows, follow our [How to install pip on Windows](https://phoenixnap.com/kb/install-pip-windows) guide.

**Step 2: Create an Environment**

1. Make a separate directory for your project:

mkdir <project name>

2. Move into the directory:

cd <project name>

3. Within the directory, create the virtual environment for Flask. When you create the environment, a new folder appears in your project directory with the environment’s name.

**Create an Environment in Linux and MacOS**

* **For Python 3:**

To create a virtual environment for Python 3, use the *venv* module and give it a name:

python3 -m venv <name of environment>

* **For Python 2:**

For Python 2, use the *virtualenv* module to create a virtual environment and name it:

python -m virtualenv <name of environment>

Listing the directory structure with the [ls command](https://phoenixnap.com/kb/linux-ls-commands) shows the newly created environment:

Project directory with created virtual environment

**Create an Environment in Windows**

* **For Python 3:**

Create and name a virtual environment in Python 3 with:

py -3 -m venv <name of environment>

* **For Python 2:**

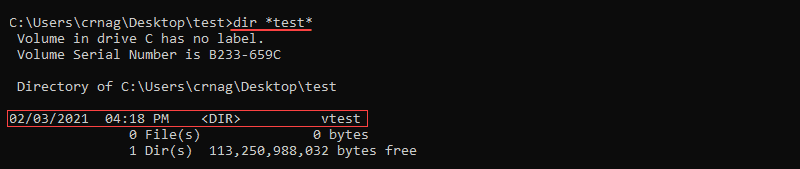
For Python 2, create the virtual environment with the *virtualenv* module:

py -2 -m virtualenv <name of environment>

List the folder structure using the **dir** command:

dir \*<project name>\*

The project directory shows the newly created environment:



**Step 3: Activate the Environment**

Activate the virtual environment before installing Flask. The name of the activated environment shows up in the CLI after activation.

**Activate the Environment on Linux and MacOS**

Activate the virtual environment in Linux and MacOS with:

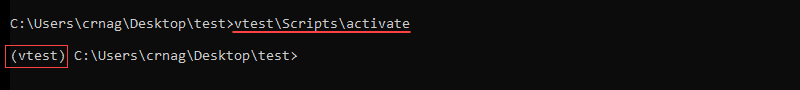
. <name of environment>/bin/activate

Activating environment terminal change

**Activate the Environment on Windows**

For Windows, activate the virtual environment with:

<name of environment>\Scripts\activate



**Step 4: Install Flask**

Install Flask within the activated environment using **pip**:

pip install Flask

Flask is installed automatically with all the dependencies.

**Note:** **pip** is a Python package manager. To install **pip** follow one of our guides: [How to install pip on CentOS 7](https://phoenixnap.com/kb/how-to-install-pip-centos-7), [How to install pip on CentOS 8](https://phoenixnap.com/kb/how-to-install-pip-on-centos-8), [How to install pip on Debian](https://phoenixnap.com/kb/how-to-install-pip-on-debian-9), or [How to install pip on Ubuntu](https://phoenixnap.com/kb/how-to-install-pip-on-ubuntu).

**Step 5: Test the Development Environment**

1. Create a simple Flask application to test the newly created development environment.

2. Make a file in the Flask project folder called *hello.py*.

3. Edit the file using a [text editor](https://phoenixnap.com/kb/best-linux-text-editors-for-coding) and add the following code to make an application that prints "*Hello world!*":

from flask import Flask

app = Flask(\_\_name\_\_)

@app.route('/')

def hello\_world():

return 'Hello world!'

**Note:** Pick any name for the project except *flask.py*. The Flask library is in a *flask.py* file.

4. Save the file and close.

5. Using the console, navigate to the project folder using the **cd** command.

6. Set the *FLASK\_APP* environment variable.

* **For Linux and Mac:**

export FLASK\_APP=hello.py

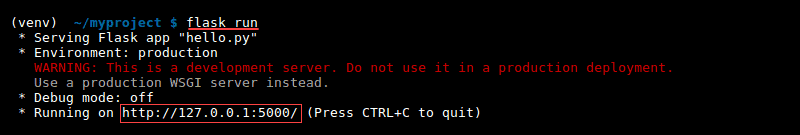
* **For Windows:**

setx FLASK\_APP "hello.py"

**Note:** Windows users must restart the console to set the environment variable. Learn more about setting environment variables by reading one of our guides: [How to set environmet variables in Linux](https://phoenixnap.com/kb/linux-set-environment-variable), [How to set environment variables in MacOS](https://phoenixnap.com/kb/set-environment-variable-mac), [How to set environment variables in Windows](https://phoenixnap.com/kb/windows-set-environment-variable).

7. Run the Flask application with:

flask run



The output prints out a confirmation message and the address.

8. Copy and paste the address into the browser to see the project running:



**Conclusion**

Flask web applications are easy to configure and run. It is one of the most popular web application frameworks for Python.

Read about the [best Python IDEs and code editors](https://phoenixnap.com/kb/best-python-ide-code-editor) to choose the best environment for further web development with Flask.