# How to install Django on Windows**[¶](https://docs.djangoproject.com/en/4.1/howto/windows/" \l "how-to-install-django-on-windows" \o "Permalink to this headline)**

This document will guide you through installing Python 3.8 and Django on Windows. It also provides instructions for setting up a virtual environment, which makes it easier to work on Python projects. This is meant as a beginner’s guide for users working on Django projects and does not reflect how Django should be installed when developing patches for Django itself.

The steps in this guide have been tested with Windows 10. In other versions, the steps would be similar. You will need to be familiar with using the Windows command prompt.

## Install Python**[¶](https://docs.djangoproject.com/en/4.1/howto/windows/" \l "install-python" \o "Permalink to this headline)**

Django is a Python web framework, thus requiring Python to be installed on your machine. At the time of writing, Python 3.8 is the latest version.

To install Python on your machine go to <https://www.python.org/downloads/>. The website should offer you a download button for the latest Python version. Download the executable installer and run it. Check the boxes next to “Install launcher for all users (recommended)” then click “Install Now”.

After installation, open the command prompt and check that the Python version matches the version you installed by executing:

**...\>** py --versio

## About **pip[¶](https://docs.djangoproject.com/en/4.1/howto/windows/" \l "about-pip" \o "Permalink to this headline)**

[pip](https://pypi.org/project/pip/) is a package manager for Python and is included by default with the Python installer. It helps to install and uninstall Python packages (such as Django!). For the rest of the installation, we’ll use **pip** to install Python packages from the command line.

## Setting up a virtual environment**[¶](https://docs.djangoproject.com/en/4.1/howto/windows/" \l "setting-up-a-virtual-environment" \o "Permalink to this headline)**

It is best practice to provide a dedicated environment for each Django project you create. There are many options to manage environments and packages within the Python ecosystem, some of which are recommended in the [Python documentation](https://packaging.python.org/guides/tool-recommendations/): https://packaging.python.org/guides/tool-recommendations/. Python itself comes with [venv](https://docs.python.org/3/tutorial/venv.html" \o "(in Python v3.10)): https://docs.python.org/3/tutorial/venv.html  for managing environments which we will use for this guide.

To create a virtual environment for your project, open a new command prompt, navigate to the folder where you want to create your project and then enter the following:

**...\>** py -m venv project-name

This will create a folder called ‘project-name’ if it does not already exist and set up the virtual environment. To activate the environment, run:

**...\>** project-name\Scripts\activate.bat

The virtual environment will be activated and you’ll see “(project-name)” next to the command prompt to designate that. Each time you start a new command prompt, you’ll need to activate the environment again.

## Install Django**[¶](https://docs.djangoproject.com/en/4.1/howto/windows/" \l "install-django" \o "Permalink to this headline)**

Django can be installed easily using **pip** within your virtual environment.

In the command prompt, ensure your virtual environment is active, and execute the following command:

**...\>** py -m pip install Django

This will download and install the latest Django release.

After the installation has completed, you can verify your Django installation by executing **django-admin --version** in the command prompt.

See [Get your database running](https://docs.djangoproject.com/en/4.1/topics/install/" \l "database-installation): https://docs.djangoproject.com/en/4.1/topics/install/#database-installation for information on database installation with Django.