## [Install Python on Windows](https://docs.microsoft.com/en-us/visualstudio/python/configure-web-apps-for-iis-windows?view=vs-2019" \l "install-python-on-windows)

See this [blog](https://medium.com/@ayushi21095/steps-to-deploy-python-django-web-application-on-windows-iis-server-379b2b87fcf9) for detail how to on installing Python

I installed the latest version (3.8) just because that was easiest. I installed in the default installer location (C:\Program Files\Python38) but it might be better to use a root location like such as C:\Python38. Although with the virtual environment it may not matter either.

# [Create and Configure a Python Virtual Environment for Your Application](https://medium.com/@ayushi21095/steps-to-deploy-python-django-web-application-on-windows-iis-server-379b2b87fcf9)

Create directory C:\Django\Pigskinners. The naming is arbitrary, but this is what I used.

At command prompt in this directory:

*python -m venv penv (creates the virtual environment “penv”)*

*penv\scripts\activate (runs script to activate the virtual environment)*

This creates virtual environment “penv”. Again the name penv is arbitrary but what I used. The command prompt should change to have (penv) before the directory name. All subsequent commands should be run in this virtual environment.

At the same command prompt:

*python -m pip install — upgrade pip*  (upgrades pip to latest version in virtual env)

# [Download Django Application from github](https://medium.com/@ayushi21095/steps-to-deploy-python-django-web-application-on-windows-iis-server-379b2b87fcf9)

This assumes that git is already installed locally. It might be installed with Python install. I got it from <https://git-scm.com/download/win>

At same command prompt as above:

*git clone* [*https://github.com/dhchamber/PigSkinners.git*](https://github.com/dhchamber/PigSkinners.git)

*or*

*git remote add origin* [*https://github.com/dhchamber/PigSkinners.git*](https://github.com/dhchamber/PigSkinners.git)

*git pull origin master*

This is what I used. Not sure on the difference. I’m still learning a lot on git and github

## [Install Packages](https://docs.microsoft.com/en-us/visualstudio/python/configure-web-apps-for-iis-windows?view=vs-2019#install-packages)

You can install all of your app's requirements into the virtual environment simply by running this at command prompt

*pip install -r requirements.txt*

This will install all the required python and Django packages for the project, including wfastcgi which is needed to connect to IIS. It should if I have the requirements file setup correctly. I tried. Otherwise they can be done manually as needed:

*pip install Django (or whatever package name is required)*

## Start the Django development server

## At this point you should be able to run the development server and the app should run locally.

At same command prompt as above:

## *python manage.py runserver*

## *you should get a response like this:*

## https://miro.medium.com/max/400/1*pqmmiLTuX2N5wtl2tf1EPw.png

## In a browser on the server, go to [*http://localhost:8000*](http://localhost:8000/)

## *You should see the PigSkinners page*

## Go back to the Command Prompt and press Ctrl-C to stop the Django development server.

## [Install IIS](https://medium.com/@Jonny_Waffles/deploy-django-on-iis-def658beae92)

## First, make sure the host server’s IIS is installed and has CGI enabled. To verify, from Windows search or the server management window click “Add Roles and Features”. Skip through the dialogues, make sure the target server is selected under Server Selection, then pause on the Server Roles tab. Make sure Web Server IIS is selected, and add the optional Application Development checkbox for CGI like shown.

## https://miro.medium.com/max/786/1*4_asBljJ4WIgiRwJmeyXkA.png

## ****Verify the IIS Installation****

## Enter [*http://localhos*t](http://localhost/) in the address bar and press Enter. You should see the default IIS page.

## https://miro.medium.com/max/400/1*DHsFkM_g1h-1n-YIxHskBw.png

## [Configure IIS to Serve Django Applications](https://medium.com/@ayushi21095/steps-to-deploy-python-django-web-application-on-windows-iis-server-379b2b87fcf9)

## This blog has good detail steps on how to setup IIS for the new Django app. I did it like this first, then I added the web.config to configure the FastCGI at the IIS level. This seems a better way.

## [Set web.config to point to the Python interpreter](https://docs.microsoft.com/en-us/visualstudio/python/configure-web-apps-for-iis-windows?view=vs-2019#set-webconfig-to-point-to-the-python-interpreter)

## We need a web.config file in the project root folder (C:\Django\PigSkinners). I first did it without this, just be setting the config manually in IIS, and it did work, but then I broke it. The second time around I tried this way and I think it is better. For one it is easier to document and the configuration can be source controlled.

## [This](https://stackoverflow.com/questions/57163406/how-to-fix-attributeerror-module-has-no-attribute-wsgi-in-wfastcgi) also helped me in fixing my problems with wfastcgi.

## This sets up “FastCGI settings at the IIS Server level.

## <configuration>

## <system.webServer>

## <handlers>

## <add name="Python FastCGI"

## path="\*"

## verb="\*"

## modules="FastCgiModule"

## scriptProcessor="C:\Django\PigSkinners\penv\Scripts\python.exe|C:\Django\PigSkinners\penv\Lib\site-packages\wfastcgi.py"

## resourceType="Unspecified"

## requireAccess="Script" />

## </handlers>

## </system.webServer>

## <appSettings>

## <!-- Required settings -->

## <add key="WSGI\_HANDLER" value="PigSkinners.wsgi.application" />

## <add key="PYTHONPATH" value="C:\Django\PigSkinners" />

## <!-- Optional settings -->

## <add key="WSGI\_LOG" value="C:\Logs\PigSkinners\_wfastcgi.log" />

## <add key="DJANGO\_SETTINGS\_MODULE" value="PigSkinners.settings" />

## </appSettings>

## </configuration>

## [Create and Configure a New IIS Web Site](https://medium.com/@ayushi21095/steps-to-deploy-python-django-web-application-on-windows-iis-server-379b2b87fcf9)

## This creates the new site (at port 81) in IIS

## [Configuring the static resource and media directories](https://www.toptal.com/django/installing-django-on-iis-a-step-by-step-tutorial)

## Create virtual directory

## Configuring the static resource and media directories

## Alias: static

## Physical path: C:\Django\PigSkinners\PigSkinners\static

## Collect static files

At same command prompt as above:

## python manage.py collectstatic

## Appendix:

## <https://www.toptal.com/django/installing-django-on-iis-a-step-by-step-tutorial>

## <https://medium.com/@ayushi21095/steps-to-deploy-python-django-web-application-on-windows-iis-server-379b2b87fcf9>

## <https://docs.microsoft.com/en-us/visualstudio/python/configure-web-apps-for-iis-windows?view=vs-2019>

## <https://medium.com/@Jonny_Waffles/deploy-django-on-iis-def658beae92>

## <https://www.codementor.io/@aswinmurugesh/deploying-a-django-application-in-windows-with-apache-and-mod_wsgi-uhl2xq09e>

## <https://nitinnain.com/setting-up-and-running-django-on-windows-iis-server/>

## <https://stackoverflow.com/questions/57163406/how-to-fix-attributeerror-module-has-no-attribute-wsgi-in-wfastcgi>

## <https://stackoverflow.com/questions/34199976/iis-config-error-this-configuration-section-cannot-be-used-at-this-path>

## <https://pypi.org/project/wfastcgi/>

## On Linix

## <https://medium.com/@_christopher/deploying-my-django-app-to-a-real-server-part-i-de78962e95ac>