Deploy App to Web with PythonAnyWhere

1. Create new project in Pycharm (where your app code and files will be located)

Choose new environment using Virtualenv

Select a python Base interpreter (no need to check the boxes under interpreter

1. Create a new .py file to start writing the code for your app. If you’ve already created the code for your app, copy those files into your new project folder.
2. Open terminal (and go to your project folder if necessary)
3. Activate your virt. environment by typing: .\venv\scripts\activate
4. Install any libraries and specific versions your app needs. My app uses the following (install any version you want):

pip install numpy==1.19.3

pip install pandas

pip install dash

pip install dash-bootstrap-components

1. Test the app on your local computer to make sure it works
2. Create requirements file by typing:

pip freeze > requirements.txt

1. Create a .gitignore file. Open it and add this line of text so GitHub does not load your venv into the repository:

venv/

1. Create local Git repository to prepare for GitHub:

git init

git add .

git commit -m "first commit"

1. Create new Github repository
2. Push an existing repository from the command line
3. Transfer repository to PythonAnyWhere

Create new PythonAnyWhere account

Open new Bash console

git clone [copy-paste clone link from GitHub]

1. Create virtual environment on PythonAnyWhere and install libraries:

mkvirtualenv myvirtualenv --python=/usr/bin/python3.8  
cd [git repository name from step 10]   
pip install -r requirements.txt

1. Add your new app to PythonAnyWhere:

Click on Add a New Web App button

Next 🡪 Flask 🡪 Python 3.8 🡪 Next

1. Once all libraries from requirements.txt have been installed, go to Web tab if need be, change the “Source Code” to:

home/myusername/[git repository name from step 10]].

1. Under the “Virtualenv” section, type myvirtualenv (which was the name you gave to your PythonAnyWhere virtual environment from step 13)
2. Got to the “WSGI configuration file” section and update the file.

Update the end of Project home’s path to match your app name.

Change last line of code to:

from elections import app

application = app.server

Save the file

1. Go back to “Web” tab and refresh the app with the Reload button.
2. Congratulations! Open your app.

**Bonus tips:**

1. Free accounts have only 512MB of file storage. If you delete an app, the virtual environment doesn’t get deleted with it. Therefore, to save space, remove the old virtual environment by opening your Bash and typing:

rm -rf /home/myusername/[.virtualenvs/myvirtualenv](https://www.pythonanywhere.com/user/charmingdata/webapps/)

1. To add more libraries to your current virtual environment, open Bash and type:

workon myvirtualenv

1. If your app doesn’t work, check the Error log.
2. Activate secure HTTPS
3. Activate web app password

**If this was helpful…**

I’m asking my viewers to support my Dash Plotly educational channel. A growing number of viewers are looking for high quality, professional content on Dash, which is hard to find. I am trying to fill that gap.

I believe that anyone working with data can benefit from knowing Dash Plotly, which is why I take the complex parts of Dash and break them down into bite-size tutorials for everyone to have.

My goal is to make this a sustainable project for my viewers, so if you appreciate my channel and are able support its existence, I would be grateful to you.

Become my supporter at: <https://www.patreon.com/charmingdata>

Thank you,  
Adam