PREV
A. Installation: Installing Python

NEXT
ings we Didn't Cover: There's Always More to Learn

Appendix B. Pythonanywhere: Deploying Your Webapp

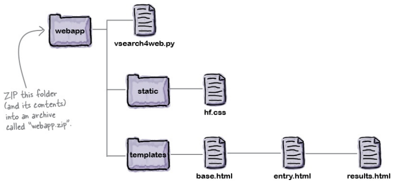
I can deploy my webapp to the cloud in about 10 minutes?!?!? I don't believe it...

At the end of Chapter 5, we claimed that deploying your webapp to the cloud was only 10 minutes away.

It's now time to make good on that promise. In this appendix, we are going to take you through the process of deploying your webapp on *PythonAnywhere*, going from zero to deployed in about 10 minutes. *PythonAnywhere* is a favorite among the Python programming community, and it's not hard to see why: it works exactly as you'd expect it to, has great support for Python (and Flask), and—best of all—you can get started hosting your webapp at no cost. Let's check out *PythonAnywhere*.

Step 0: A Little Prep

At the moment, you have your webapp code on your computer in a folder called `webapp`, which contains the `vsearch4web.py` file and the `static` and `templates` folders (as shown below). To prepare all this stuff for deployment, create a ZIP archive file of everything in your `webapp` folder, and call the archive file `webapp.zip`:



In addition to `webapp.zip`, you also need to upload and install the `vsearch` module from Chapter 4. For now, all you need to do is locate the distribution file that you created back then. On our computer, the archive file is called `vsearch-1.0.tar.gz` and it's stored in our `mymodules/vsearch/dist` folder (on *Windows*, the file is likely called `vsearch-1.0.zip`).

NOTE

Recall from Chapter 4 that Python's "setuptools" module creates ZIPs on Windows, and .tar.gz files on everything else.

You don't need to do anything with either archive file right now. Just make a note of where both archive files are on your computer so that they are easy to find when you upload them to *PythonAnywhere*. Feel free to grab a pencil and scribble down each archive file's location here:

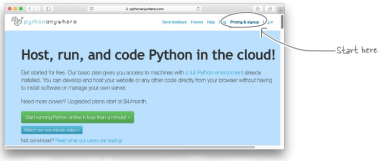
webapp.zip ____________________

vsearch-1.0.tar.gz ____________________

This is "vsearch.zip" instead if you're on Windows.

Step 1: Sign Up for PythonAnywhere

This step couldn't be any easier. Surf over to *pythonanywhere.com (http://www.pythonanywhere.com)*, then click on the **Pricing & signup** link:



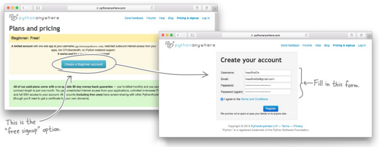
Host, run, and code Python in the cloud!

Get started for free. Our basic plan gives you access to machines with a full Python environment already installed. You can develop and host your website or any other code directly from your browser without having to install software or manage your own server.

Need more power? Upgraded plans start at $4/month.

Not convinced? Read what our users are saying!

Click on the big, blue button to create a *Beginner account*, then fill in the details on the signup form:



Plans and pricing

Beginner: Free!

A limited account with one web app at your-username.pythonanywhere.com, restricted outbound Internet access from your apps, low CPU/bandwidth, no IPython notebook support.

Create your account

Username: headfirst2e

Email: headfirst2e@gmail.com

Password: ••••••••••••

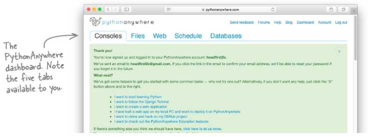
Password (again): ••••••••••••

☑ I agree to the Terms and Conditions

We promise not to spam or pass your details on to anyone else.

Copyright © 2015 PythonAnywhere LLP — Terms — Privacy
"Python" is a registered trademark of the Python Software Foundation.

If all is well, the PythonAnywhere dashboard appears. Note: you are both registered *and* signed in at this point:



Thank you!

You're now signed up and logged in to your PythonAnywhere account: **headfirst2e**.

We've sent an email to **headfirst2e@gmail.com**. If you click the link in the email to confirm your email address, we'll be able to reset your password if you forget it in the future.

What next?

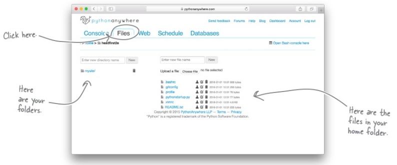
We've got some helpers to get you started with some common tasks — why not try one out? Alternatively, if you don't want any help, just click the "X" button above and to the right.

- I want to start learning Python
- I want to follow the Django Tutorial
- I want to create a web application
- I have built a web app on my local PC and want to deploy it on PythonAnywhere
- I want to clone and hack on my GitHub project
- I want to check out the PythonAnywhere Education features

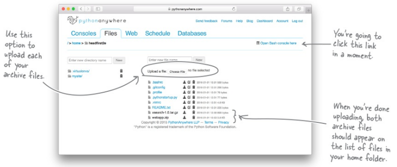
If there's something else you think we should have here, click here to let us know.

Step 2: Upload Your Files to the Cloud

Click on the **Files** tab to view the folders and files available to you:



Use the *Upload a file* option to locate and upload the two archive files from **Step 0**:



/ > home > headfirst2e

.virtualenvs/
mysite/

Upload a file: Choose File no file selected

.bashrc
.gitconfig
.profile
.pythonstartup.py
.vimrc
README.txt
vsearch-1.0.tar.gz
webapp.zip

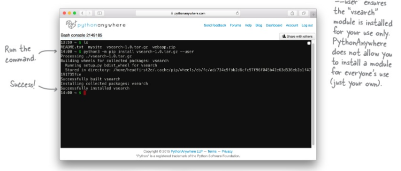
Consoles
Files
Web
Schedule
Databases
Open Bash console here
Use this option to upload each of your archive files.
You're going to click this link in a moment.
When you're done uploading both archive files should appear on the list of files in your home folder.

You’re now ready to extract and install these two uploaded archive files, and you’ll do that during **Step 3**. To get ready, click the *Open a bash console here* link at the top right of the above page. This opens up a terminal window in your browser window (on *PythonAnywhere*).

Step 3: Extract and Install Your Code

When you click the *Open a bash console here* link, *PythonAnywhere* responds by replacing the *Files* dashboard with a browser-based Linux console (command prompt). You’re going to issue a few commands to extract and install the `vsearch` module as well as your webapp’s code within this console. Begin by installing `vsearch` into Python as a “private module” (i.e., just for your use) using this command (be sure to use `vsearch-1.0.zip` if you’re on *Windows*):

```
python3 -m pip install vsearch-1.0.tar.gz --user
```



"--user" ensures the "vsearch" module is installed for your use only. PythonAnywhere does not allow you to install a module for everyone's use (just your own).
Run the command.
Success!
pythonanywhere
Send feedback Forums Help Blog Dashboard Account Log out
Share with others

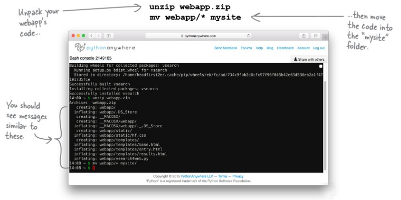
Bash console 2149185

```
13:59 ~ $ ls
README.txt  mysite  vsearch-1.0.tar.gz  webapp.zip
14:00 ~ $ python3 -m pip install vsearch-1.0.tar.gz --user
Processing ./vsearch-1.0.tar.gz
Building wheels for collected packages: vsearch
  Running setup.py bdist_wheel for vsearch
  Stored in directory: /home/headfirst2e/.cache/pip/wheels/eb/fc/ad/734c9fbb2d6cfc97f96f045b42e63d536eb2a1f47
191735fce
Successfully built vsearch
Installing collected packages: vsearch
Successfully installed vsearch
14:00 ~ $ 
```

Copyright © 2015 PythonAnywhere LLP — Terms — Privacy
"Python" is a registered trademark of the Python Software Foundation.

With the `vsearch` module successfully installed, it’s time to turn your attention to your webapp’s code, which has to be installed into the `mysite` folder (which already exists on your *PythonAnywhere* home folder). To do this, you need to issue two commands:

```
unzip webapp.zip
mv webapp/* mysite
```



Unpack your webapp's code...
...then move the code into the "mysite" folder.
You should see messages similar to these.
pythonanywhere
Send feedback Forums Help Blog Dashboard Account Log out
Share with others

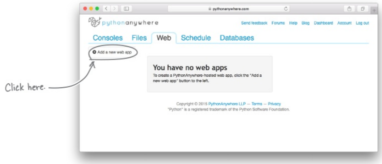
Bash console 2149185

```
Building wheels for collected packages: vsearch
  Running setup.py bdist_wheel for vsearch
  Stored in directory: /home/headfirst2e/.cache/pip/wheels/eb/fc/ad/734c9fbb2d6cfc97f96f045b42e63d536eb2a1f47
191735fce
Successfully built vsearch
Installing collected packages: vsearch
Successfully installed vsearch
14:00 ~ $ unzip webapp.zip
Archive:  webapp.zip
   creating: webapp/
  inflating: webapp/.DS_Store
   creating: __MACOSX/
   creating: __MACOSX/webapp/
  inflating: __MACOSX/webapp/._.DS_Store
   creating: webapp/static/
  inflating: webapp/static/hf.css
   creating: webapp/templates/
  inflating: webapp/templates/base.html
  inflating: webapp/templates/entry.html
  inflating: webapp/templates/results.html
  inflating: webapp/vsearch4web.py
14:08 ~ $ mv webapp/* mysite/
14:08 ~ $ 
```

Copyright © 2015 PythonAnywhere LLP — Terms — Privacy
"Python" is a registered trademark of the Python Software Foundation.

Step 4: Create a Starter Webapp, 1 of 2

With **Step 3** done, return to the *PythonAnywhere* dashboard and select the **Web** tab, where *PythonAnywhere* invites you to create a new starter webapp. You’ll do this, then swap out the starter’s webapp code for your own. Note that each *Beginner account* gets one webapp for free; if you want more, you’ll have to upgrade to a paid account. Luckily—for now—you only need the one, so let’s keep going by clicking *Add a new web app*:

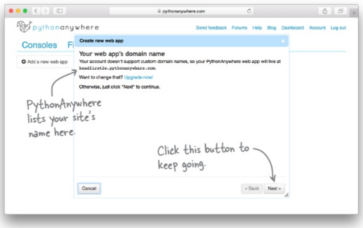


You have no web apps

To create a PythonAnywhere-hosted web app, click the "Add a new web app" button to the left.

Copyright © 2015 PythonAnywhere LLP — Terms — Privacy
"Python" is a registered trademark of the Python Software Foundation.

As you are using a free account, your webapp is going to run on the site name shown on the next screen. Click the *Next* button to proceed with *PythonAnywhere’s* suggested site name:



PythonAnywhere lists your site's name here.
Click this button to keep going.
pythonanywhere
Send feedback Forums Help Blog Dashboard Account Log out
Consoles
Fi
Add a new web app
Create new web app
Cancel
« Back
Next »

Your web app's domain name

Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `headfirst2e.pythonanywhere.com`.

Want to change that? Upgrade now!

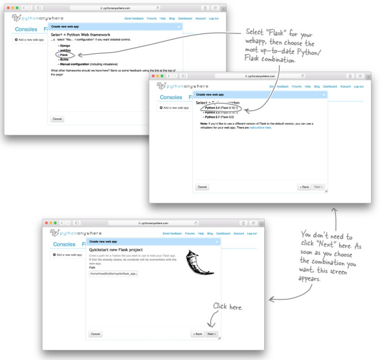
Otherwise, just click "Next" to continue.

Click *Next* to continue with this step.

Step 4: Create a Starter Webapp, 2 of 2

PythonAnywhere supports more than one Python web framework, so the next screen offers you a choice among the many supported systems. Pick Flask, then select the version of Flask and Python you wish to deploy to. As of this writing, Python 3.4 and Flask 0.10.1 are the most up-to-date versions supported by

PythonAnywhere, so go with that combination unless a newer combination is offered (in which case, pick the newer one instead):



Create new web app

Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- Django
- web2py
- Flask
- Bottle
- Manual configuration (including virtualenvs)

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Create new web app

Select a Python version

- Python 3.4 (Flask 0.10.1)
- Python 3.3 (Flask 0.10.1)
- Python 2.7 (Flask 0.9)

Note: If you'd like to use a different version of Flask to the default version, you can use a virtualenv for your web app. There are instructions here.

Create new web app

Quickstart new Flask project

Enter a path for a Python file you wish to use to hold your Flask app. If this file already exists, its contents will be overwritten with the new app.

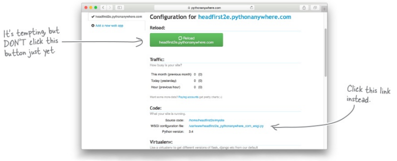
Path: /home/headfirst2e/mysite/flask_app.py

pythonanywhere
Send feedback
Forums
Help
Blog
Dashboard
Account
Log out
Consoles
Add a new web app
Cancel
« Back
Next »
Select "Flask" for your webapp, then choose the most up-to-date Python/Flask combination.
You don't need to click "Next" here. As soon as you choose the combination you want, this screen appears.
Click here.

You're nearly there. The next screen offers to create a quickstart Flask webapp. Go ahead and do that now by accepting the values on this page and clicking on the *Next* button to continue:

Step 5: Configure Your Webapp

With **Step 4** complete, you are presented with the **Web** dashboard. Don't be tempted to click that big, green button just yet—you haven't told *PythonAnywhere* about your code yet, so hold off on running anything for now. Instead, click in the long link to the right of the *WSGI configuration file* label:



Configuration for headfirst2e.pythonanywhere.com

Reload:

Traffic:

How busy is your site?

This month (previous month)	0	(0)
Today (yesterday)	0	(0)
Hour (previous hour)	0	(0)

Want some more data? Paying accounts get pretty charts ;-)

Code:

What your site is running.

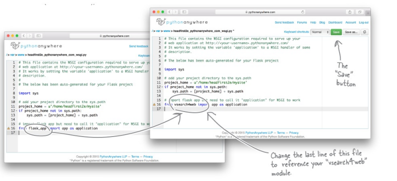
Source code: /home/headfirst2e/mysite
WSGI configuration file: /var/www/headfirst2e_pythonanywhere_com_wsgi.py
Python version: 3.4

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default

headfirst2e.pythonanywhere.com
Add a new web app
Reload headfirst2e.pythonanywhere.com
It's tempting, but DON'T click this button just yet.
Click this link instead.

Clicking that long link loads your newly created Flask webapp's configuration file into *PythonAnywhere's* web-based text editor. At the end of Chapter 5, we told you that *PythonAnywhere* imports your webapp code before invoking `app.run()` for you. This is the file that supports that behavior. However, it needs to be told to reference *your* code, not the code in the starter app, so you need to edit the last line of this file (as shown below), and then click *Save*:



/> var > www > headfirst2e_pythonanywhere_com_wsgi.py

```
# This file contains the WSGI configuration required to serve up your
# web application at http://<your-username>.pythonanywhere.com/
# It works by setting the variable 'application' to a WSGI handler of some
# description.
#
# The below has been auto-generated for your Flask project

import sys

# add your project directory to the sys.path
project_home = u'/home/headfirst2e/mysite'
if project_home not in sys.path:
    sys.path = [project_home] + sys.path

# import flask app but need to call it "application" for WSGI to work
from flask_app import app as application
```

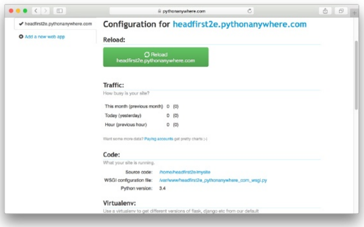
/> var > www > headfirst2e_pythonanywhere_com_wsgi.py *

```
# This file contains the WSGI configuration required to serve up your
# web application at http://<your-username>.pythonanywhere.com/
# It works by setting the variable 'application' to a WSGI handler of some
# description.
#
# The below has been auto-generated for your Flask project

import sys

# add your project directory to the sys.path
project_home = u'/home/headfirst2e/mysite'
if project_home not in sys.path:
    sys.path = [project_home] + sys.path

# import flask app but need to call it "application" for WSGI to work
from vsearch4web import app as application
```

pythonanywhere
Send feedback
Forums
Help
Blog
Dashboard
Account
Log out
Keyboard shortcuts
Normal
Save
Save as...
Copyright © 2015 PythonAnywhere LLP — Terms — Privacy
"Python" is a registered trademark of the Python Software Foundation.
The "Save" button
Change the last line of this file to reference your "vsearch4web" module.

Step 6: Take Your Cloud-Based Webapp for a Spin!

Be sure to save your changed configuration file, then return to the **Web** tab on the dashboard. It is now time to click on that big, tempting, green button. Go for it!



Configuration for headfirst2e.pythonanywhere.com

Reload:

Traffic:

How busy is your site?

This month (previous month)	0	(0)
Today (yesterday)	0	(0)
Hour (previous hour)	0	(0)

Want some more data? Paying accounts get pretty charts ;-)

Code:

What your site is running.

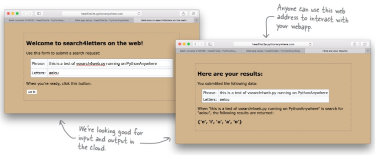
Source code: /home/headfirst2e/mysite
WSGI configuration file: /var/www/headfirst2e_pythonanywhere_com_wsgi.py
Python version: 3.4

Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default

headfirst2e.pythonanywhere.com
Add a new web app
Reload headfirst2e.pythonanywhere.com

After but a brief moment, your webapp appears in your browser, and it works exactly as it did when you ran it locally, only now anybody with an Internet connection and a web browser can use it too:


headfirst2e.pythonanywhere.com
Do it!
headfirst2e.pythonanywhere.com

Welcome to search4letters on the web!

Use this form to submit a search request:

Phrase:	this is a test of vsearch4web.py running on PythonAnywhere
Letters:	aeiou

When you're ready, click this button:

Here are your results:

You submitted the following data:

Phrase:	this is a test of vsearch4web.py running on PythonAnywhere
Letters:	aeiou

When "this is a test of vsearch4web.py running on PythonAnywhere" is search for "aeiou", the following results are returned:

{'e', 'i', 'u', 'a', 'o'}

Anyone can use this web address to interact with your webapp.

We're looking good for input and output in the cloud.

And with that, you're done. The webapp you developed in Chapter 5 has been deployed to *PythonAnywhere's* cloud (in less than 10 minutes). There's lots more to *PythonAnywhere* than what's shown in this short appendix, so feel free to explore and experiment. At some point, remember to return the *PythonAnywhere* dashboard and log out. Note that, despite your logging out, your webapp keeps running in the cloud until you tell it otherwise. That's pretty cool, isn't it?

PREV
A. Installation: Installing Python

NEXT
C. Top Ten Things we Didn't Cover: There's Always More to Learn