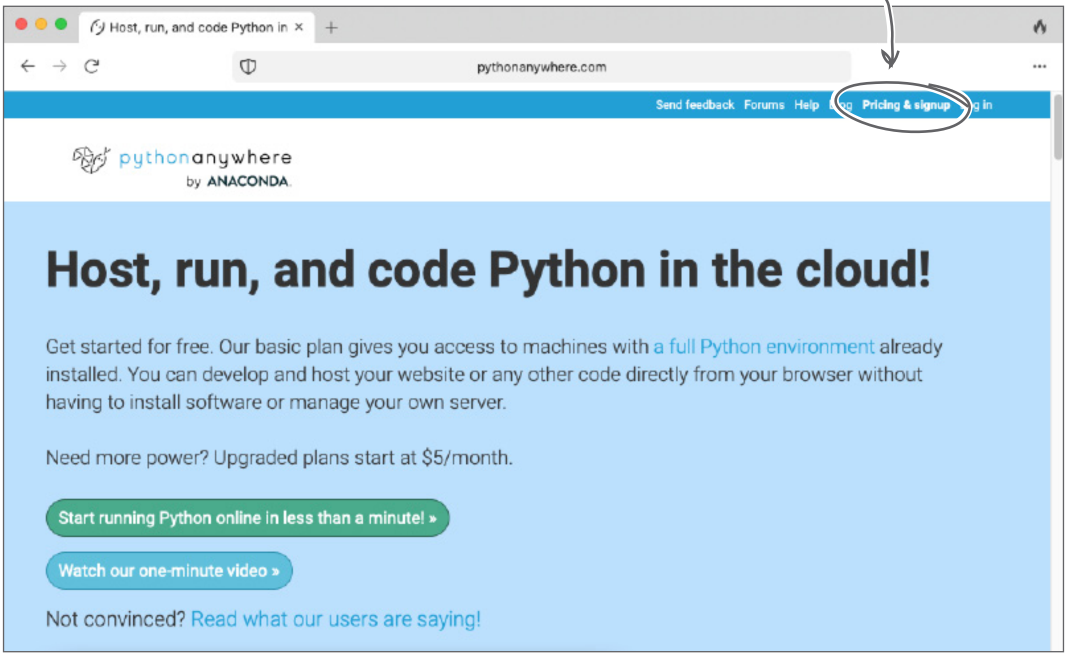
The ten steps to cloud deployment

With a decision made to run on *PythonAnywhere*, let's work through the process of deploying to this Python-focused cloud platform. There are ten steps in total.

Be sure to follow along on as you learn about the process.

1. **Create a Beginner Account on PythonAnywhere.**
Surf on over to *https://pythonanywhere.com* to begin the process of creating a new beginner account.

Start here.

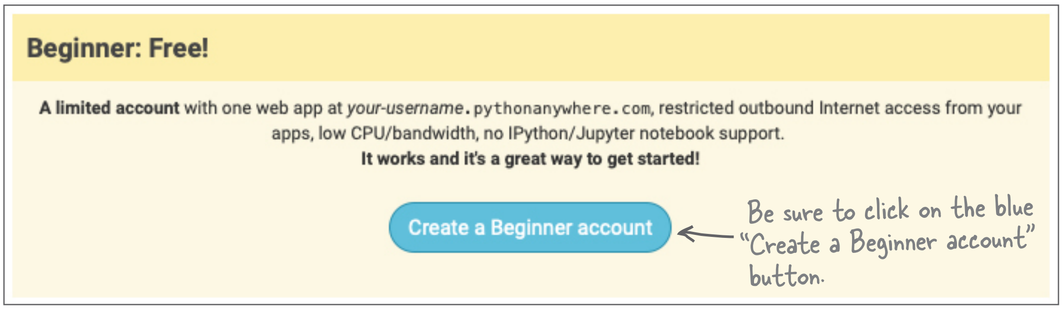

Host, run, and code Python in ×
pythonanywhere.com
Send feedback
Forums
Help
Pricing & signup
pythonanywhere by ANACONDA
Start running Python online in less than a minute! »
Watch our one-minute video »

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 $5/month.

Not convinced? Read what our users are saying!



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/Jupyter notebook support.
It works and it's a great way to get started!

Be sure to click on the blue "Create a Beginner account" button.

A beginner account is all you need

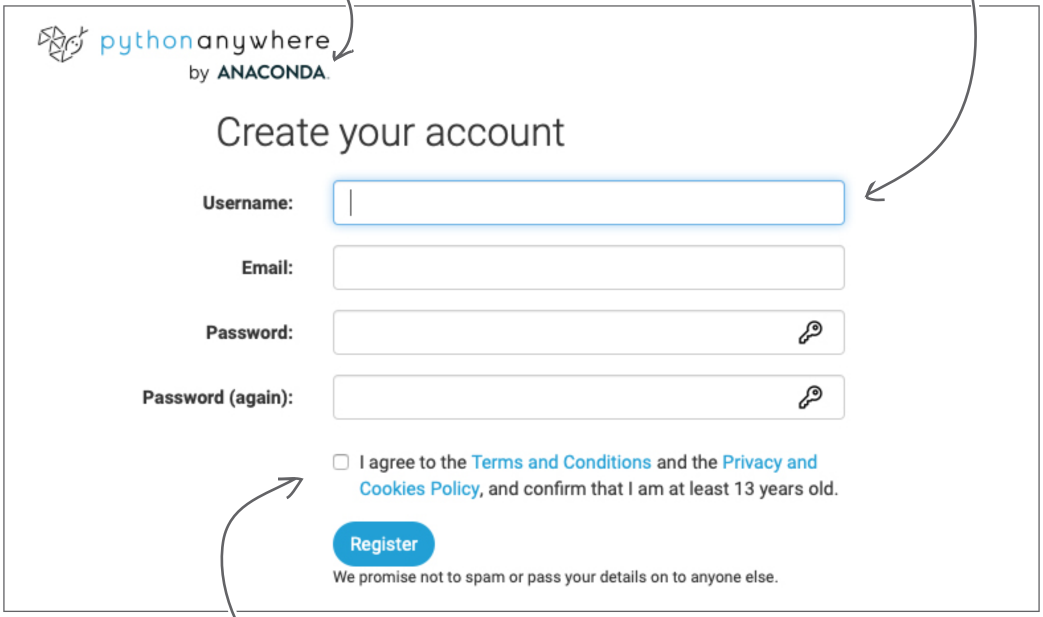
The PythonAnywhere *Beginner* account does have *some* usage restrictions. However, for what you need with the Coach's webapp, you won't be bothered by them any time soon.

Even if you do end up hitting the beginner limits, paid plans start from as little as $5 per month.

2 **Provide all the usual details.**
Go ahead and register your details with PythonAnywhere using the provided sign-up form.

PythonAnywhere is an Anaconda company, one of the biggest names in the Python data science ecosystem.

Sorry, but "headfirstpython" is already taken.



Create your account

Username:	
Email:	
Password:	
Password (again):	

□ I agree to the Terms and Conditions and the Privacy and Cookies Policy, and confirm that I am at least 13 years old.

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

Take a moment to read the T&Cs, then click this little checkbox before registering.

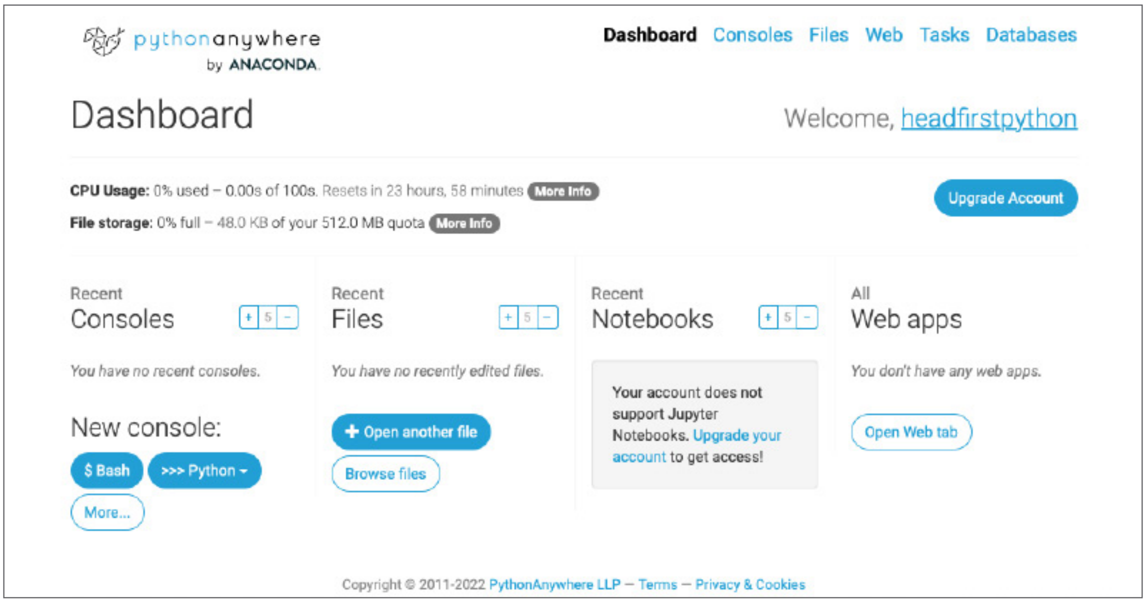
You might think you'll have to wait to confirm your email address before continuing, right? Well, the PythonAnywhere folks work hard at *not* getting in your way…

There's nothing stopping you from starting...

Although PythonAnywhere just sent an email to confirm your email address, they are trusting sorts and are more than happy to let you proceed (with the strict understanding that you'll confirm your email soon).

PythonAnywhere's dashboard appears.

There's a lot going on here. For now, concentrate on these links, which lead to each of the dashboard's tabs.



CPU Usage: 0% used – 0.00s of 100s. Resets in 23 hours, 58 minutes

File storage: 0% full – 48.0 KB of your 512.0 MB quota

You have no recent consoles.

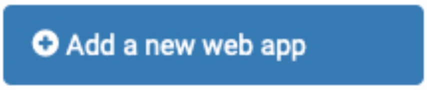
You have no recently edited files.

Your account does not support Jupyter Notebooks. Upgrade your account to get access!

You don't have any web apps.

Copyright © 2011-2022 PythonAnywhere LLP – Terms – Privacy & Cookies

3 **Click on the Web tab, then create a new webapp.**
Beginner accounts on PythonAnywhere can have one webapp active at any one time. You have yet to create any, so let's fix that.



Go for it! This button is just waiting to be clicked...

When in doubt, stick with the defaults

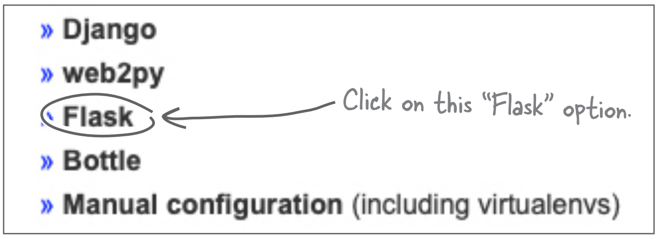
Upon clicking that big blue button, you'll see a message informing you your account can't employ a custom domain name (without upgrading). That's no biggie. By default, your webapp runs on a domain name created by prefixing your newly registered PythonAnywhere username with the *.pythonanywhere.com* postfix. This combination (for us) results in the following web address:

headfirstpython.pythonanywhere.com

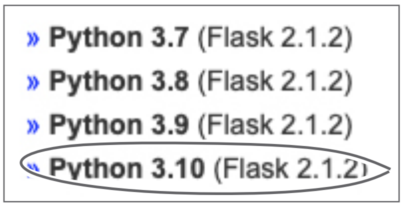
Go ahead and click on the **Next** button.

Go on, click it (you know you want to).

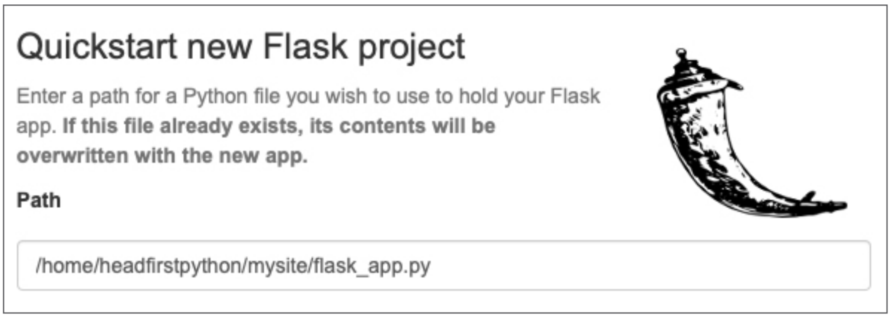
4 **Select "Flask" to create a default, placeholder webapp.**
A list of choices of web framework appears. Click on the "Flask" option, then select the most recent Python release from the list that follows.


» Django
» web2py
[✓ » Flask]
» Bottle
» Manual configuration (including virtualenvs)

Click on this "Flask" option.


» Python 3.7 (Flask 2.1.2)
» Python 3.8 (Flask 2.1.2)
» Python 3.9 (Flask 2.1.2)
[✓ » Python 3.10 (Flask 2.1.2)]

Select the most recent Python release (and don't worry if you see releases later than 3.10).


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

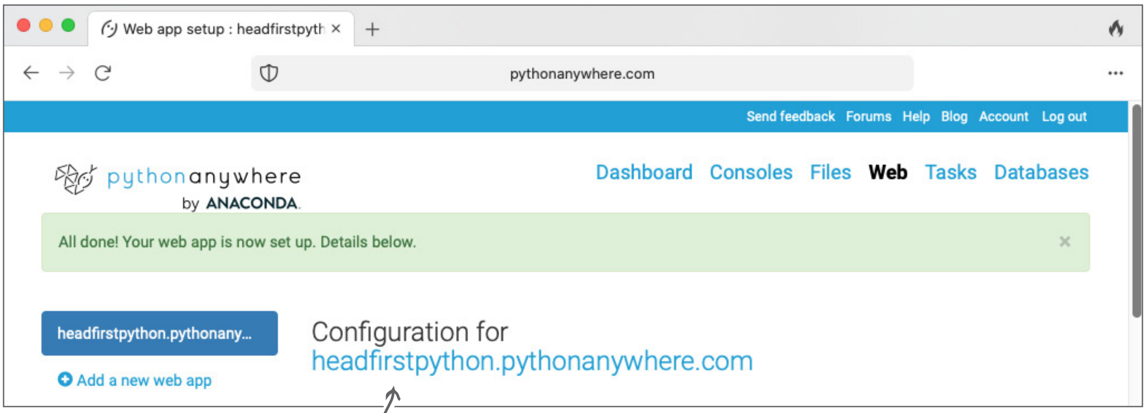
/home/headfirstpython/mysite/flask_app.py

When you're presented with this dialog, change nothing. Accept the defaults by clicking on the "Next" button.

The placeholder webapp doesn't do much

You're presented with the web setup confirmation page for your new webapp, which provides a bunch of configuration options. For now, let's not adjust any of these. Instead, click on the blue-highlighted URL for your new webapp to see it in action:

The top of your new webapp's configuration page

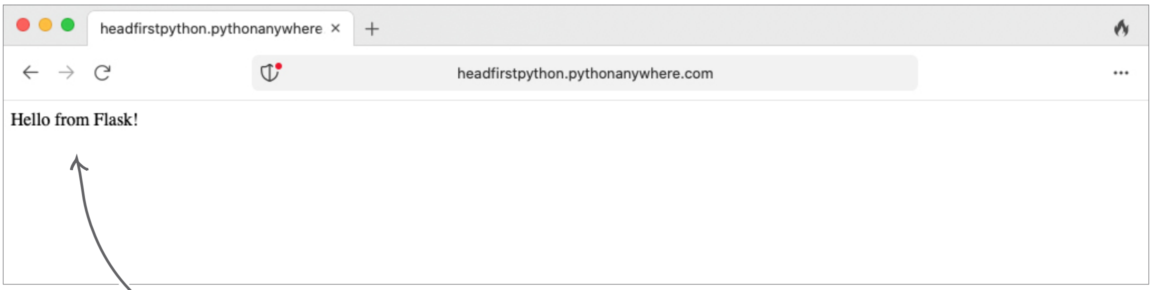


All done! Your web app is now set up. Details below.

Configuration for headfirstpython.pythonanywhere.com

Click here...

... to run your new webapp.


headfirstpython.pythonanywhere
headfirstpython.pythonanywhere.com

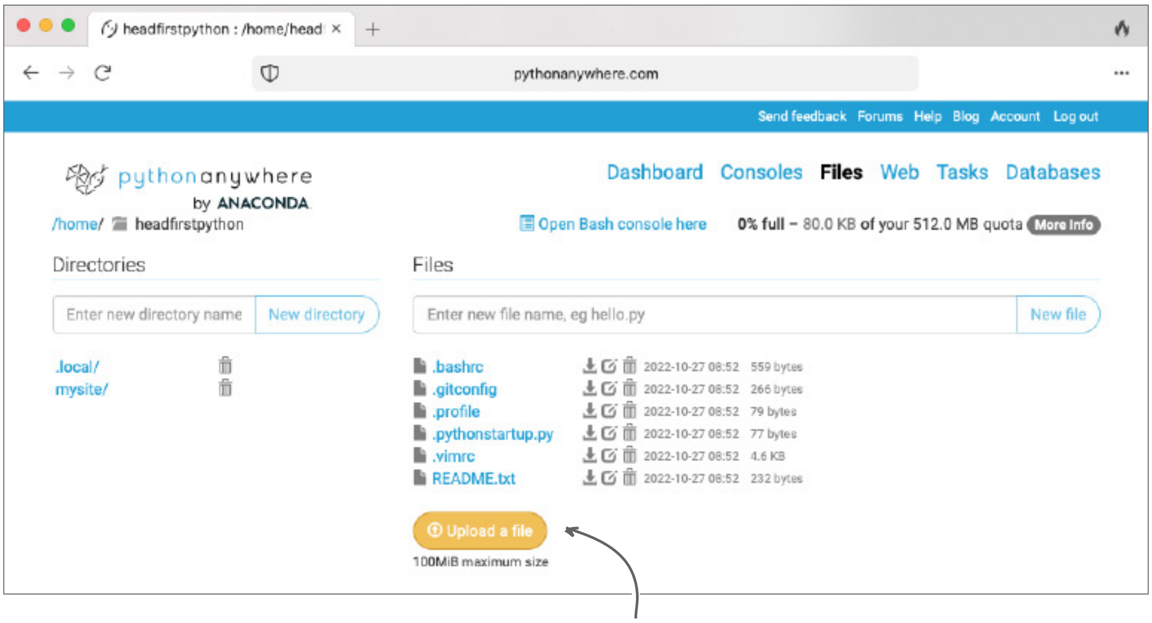
Hello from Flask!

We know... this is almost too exciting.

When you're done basking in the glory of the default Flask webapp, press your browser's **Back** button to return to the configuration page.

Deploying your code to PythonAnywhere

Having pressed your browser's **Back** button, you are returned to the PythonAnywhere **Web** tab. Move to the **Files** tab by clicking on "Files." Your browser should display a page not unlike this:



headfirstpython : /home/head ×

pythonanywhere.com

Send feedback Forums Help Blog Account Log out

pythonanywhere by ANACONDA

Dashboard Consoles **Files** Web Tasks Databases

/home/ headfirstpython

Open Bash console here 0% full – 80.0 KB of your 512.0 MB quota More Info

Directories

Enter new directory name | New directory

.local/
mysite/

Files

Enter new file name, eg hello.py | New file

.bashrc	2022-10-27 08:52	559 bytes
.gitconfig	2022-10-27 08:52	266 bytes
.profile	2022-10-27 08:52	79 bytes
.pythonstartup.py	2022-10-27 08:52	77 bytes
.vimrc	2022-10-27 08:52	4.6 KB
README.txt	2022-10-27 08:52	232 bytes

Upload a file

100MiB maximum size

You might be tempted to click on this button, but don't do so until you've completed Step #5.

5 **Prepare the Coach's webapp code for deployment.**
You have a copy of the Coach's webapp (code, templates, CSS, et al.) in your *webapp* folder on your computer. Use your operating system's compression technology to ZIP everything in your *webapp* folder into a file called *webapp.zip*.

6 **Use the "Upload a file" button to copy *webapp.zip* to the cloud.**

Watch it!

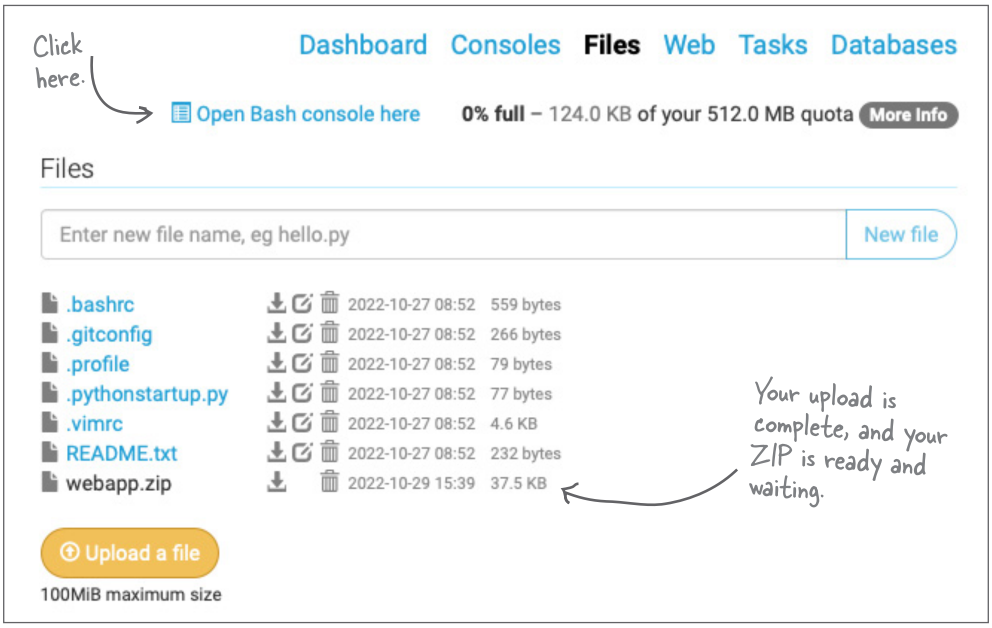
Care is needed when first uploading your code to PythonAnywhere.

Don't be tempted to upload your webapp's files individually, which is possible, but prone to error (because sure-as-shootin' you'll forget something). ZIP is your friend.

Extract your code in the console

With your code uploaded, you can open a terminal on the PythonAnywhere cloud to *unzip webapp.zip*, which is Step 7.

7 **Unzip your just-uploaded ZIP file.**
Do you see that "Open Bash console here" link underneath the word "Dashboard"? Go ahead and click on that link to open a browser-based terminal running on the PythonAnywhere cloud.



Click here.

0% full – 124.0 KB of your 512.0 MB quota

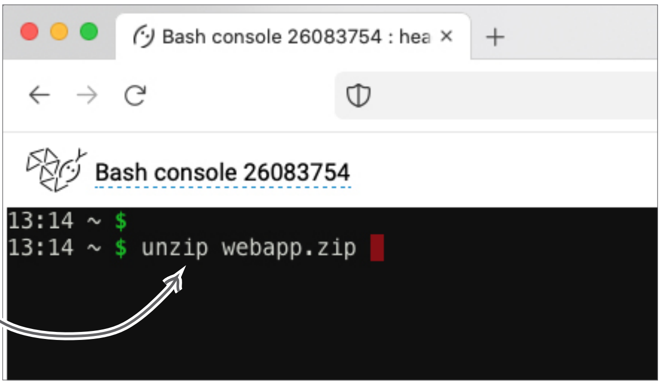
Files

Enter new file name, eg hello.py

.bashrc	2022-10-27 08:52	559 bytes
.gitconfig	2022-10-27 08:52	266 bytes
.profile	2022-10-27 08:52	79 bytes
.pythonstartup.py	2022-10-27 08:52	77 bytes
.vimrc	2022-10-27 08:52	4.6 KB
README.txt	2022-10-27 08:52	232 bytes
webapp.zip	2022-10-29 15:39	37.5 KB

100MiB maximum size

Your upload is complete, and your ZIP is ready and waiting.



```
13:14 ~ $
13:14 ~ $ unzip webapp.zip
```

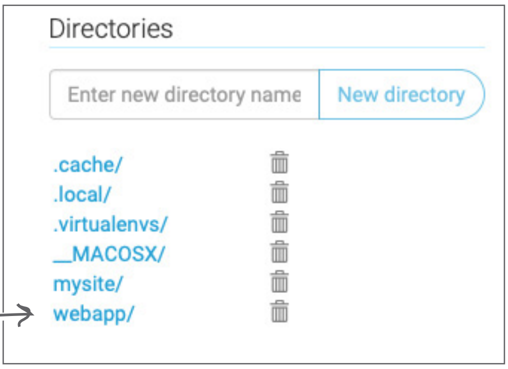
The terminal runs within a version of Linux running the bash shell. Type this command at the prompt, then press the Enter key to start the extraction process.

Configure the Web tab to point to your code

The `unzip` command (at the bottom of the previous page) unpacks your code and creates a folder called *webapp* on your PythonAnywhere top-level folder.

Once the command completes, type `exit` at the console prompt to terminate the terminal, then click on the PythonAnywhere "snake logo" to return to your dashboard.

The "snake logo." When you click on this logo, you are always returned to your PythonAnywhere dashboard.

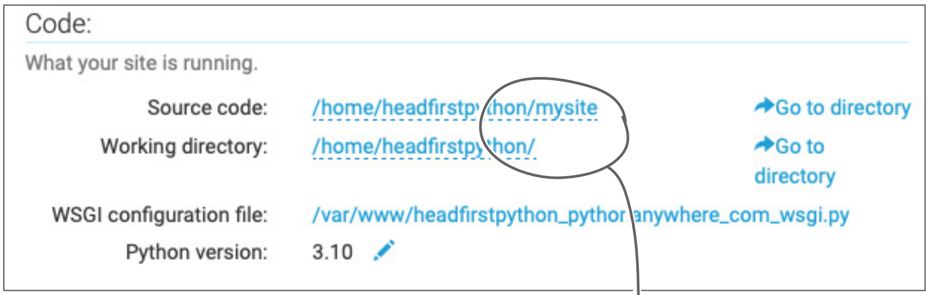


.cache/
.local/
.virtualenvs/
__MACOSX/
mysite/
webapp/

Clicking on the "Files" tab confirms that a folder called "webapp" now exists.

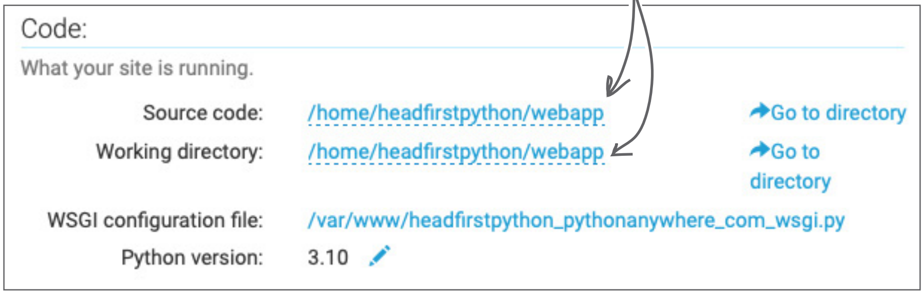
8 **Configure your webapp's folders.**
Click on your dashboard's **Web** tab, then scroll down until you see the **Code** settings. Click on the "Source code" and "Working directory" entries to adjust both to refer to your *webapp* folder.

Before...



Source code: /home/headfirstpython/mysite
Working directory: /home/headfirstpython/
WSGI configuration file: /var/www/headfirstpython_pythonanywhere_com_wsgi.py
Python version: 3.10

After...



Source code: /home/headfirstpython/webapp
Working directory: /home/headfirstpython/webapp
WSGI configuration file: /var/www/headfirstpython_pythonanywhere_com_wsgi.py
Python version: 3.10

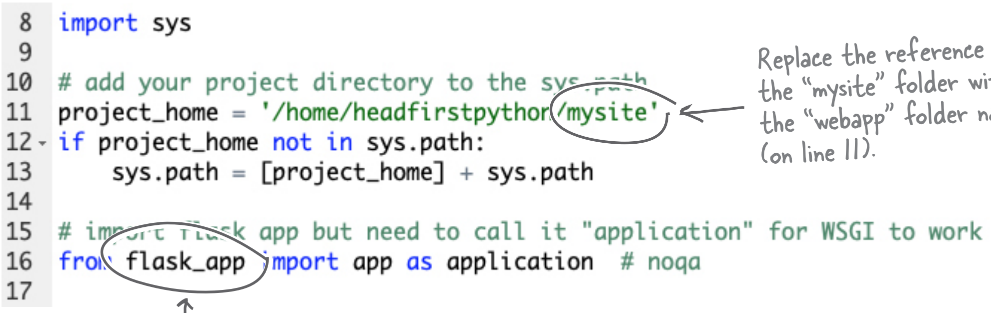
Edit your webapp's WSGI file

The third link in the **Code** section of your webapp's setup tab refers to your webapp's WSGI (*Web Server Gateway Interface*) configuration.

On PythonAnywhere, the WSGI configuration file tells the server where to find your code as well as how to start it. The default webapp puts its code in the *mysite* folder and uses the *flask_app.py* file as its starting point, and these settings are referred to in the WGSI file. Both need to change in order to run the Coach's webapp.

9 **Edit the WSGI file to refer to the Coach's webapp code.**
Click on the WSGI link to load the configuration code into PythonAnywhere's text editor. Apply the two changes as shown below.

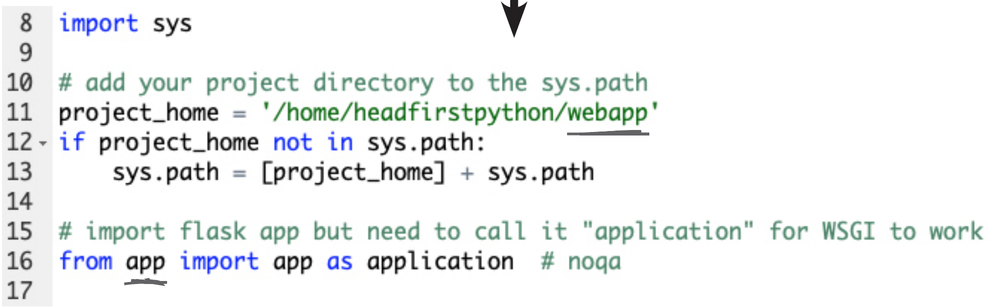
Before...



```
8   import sys
9
10  # add your project directory to the sys.path
11  project_home = '/home/headfirstpython/mysite'
12  if project_home not in sys.path:
13      sys.path = [project_home] + sys.path
14
15  # import flask app but need to call it "application" for WSGI to work
16  from flask_app import app as application  # noqa
17
```

Replace the reference to the "mysite" folder with the "webapp" folder name (on line 11).

As the Coach's code resides in the "app.py" file, you need to change the reference to "flask_app" to be "app" (on line 16).



```
8   import sys
9
10  # add your project directory to the sys.path
11  project_home = '/home/headfirstpython/webapp'
12  if project_home not in sys.path:
13      sys.path = [project_home] + sys.path
14
15  # import flask app but need to call it "application" for WSGI to work
16  from app import app as application  # noqa
17
```

After...

With these two small edits applied, click on that big green **Save** button, then click on the snake logo to return to your dashboard, then return to the **Web** tab once more.

Save

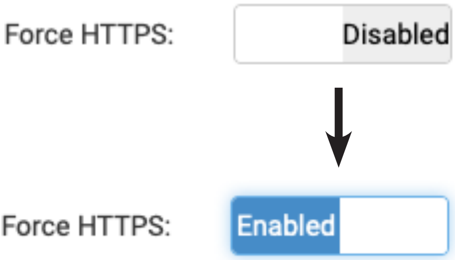
Geek Note

The term "WSGI" is pronounced "whiskey" by most Pythonistas, as opposed to spelling out the four individual letters. Nobody says "W…S…G…I." The whiskey pronunciation can lead to some confusion, especially when newbies attend Python conferences or meetups for the first time. There's many a poor soul that's been left wondering if they're in the right place…. as everyone in the *Web Track* seems to be talking about hard liquor.

You're almost there. You've created a PythonAnywhere account, created a default Flask webapp, uploaded the Coach's code, and reconfigured your hosted webapp's configuration. Two quick clicks stand between you and your first test run of your PythonAnywhere-hosted webapp.

10 **Switch on HTTPS, then reload your running webapp.**
The good folk at PythonAnywhere provide HTTPS support at no extra cost, so be sure to switch it on *before* reloading your webapp code.

Before...



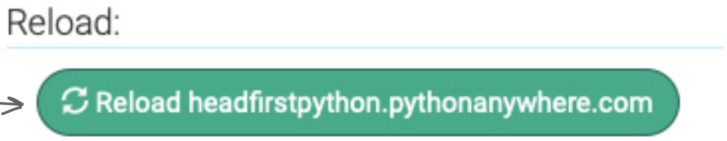
Force HTTPS: Disabled

Force HTTPS: Enabled

After...

Click #1: scroll to near the bottom of the web configuration page, then click on the word "Disabled" to ensure HTTPS is used by default.

Click #2: scroll to the *top* of your web configuration page, then click that big green "reload" button to swap the default webapp with the Coach's code. Your cloud-hosted webapp is restarted.



Reload:

That's the 10 steps done, which depending on how quickly you read/type, may have only consumed 10 minutes. All that remains is to take your cloud-hosted version of the Coach's webapp for a spin. *Drum roll, please*… can you feel the *excitement*?