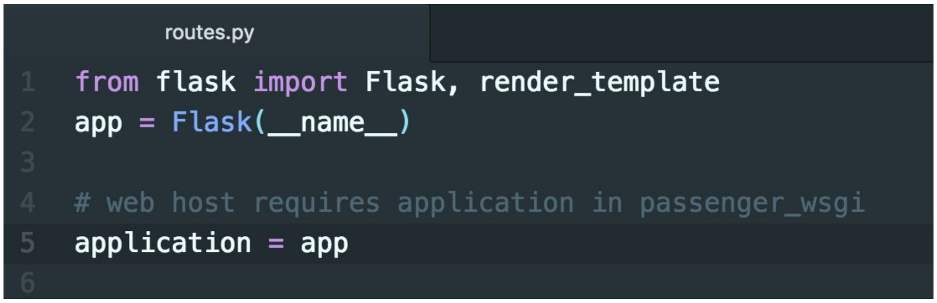
Python at Reclaim Hosting

Before starting this tutorial, make sure your Flask app is fully functioning and complete, with all files in one project folder (it may contain subfolders) on your own computer.

Make the following change in the *.py* file that runs your Flask app:



routes.py

```
from flask import Flask, render_template
app = Flask(__name__)

# web host requires application in passenger_wsgi
application = app

```

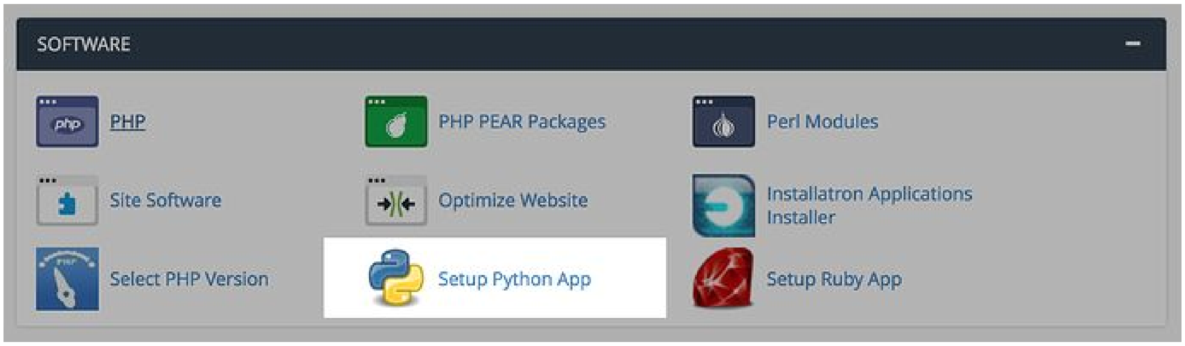
Below the line that says:
`app = Flask(__name__)`

Add this line:
`application = app`

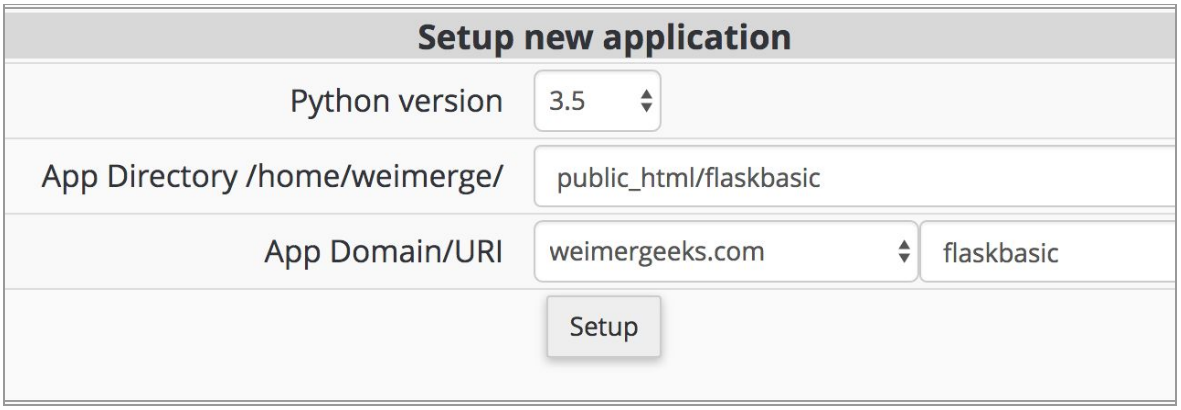
Save the file. Why you must do this is explained on the last page of this document.

Use your normal method to *copy* or upload that project folder into the *public_html* directory on the server at Reclaim. You may rename the project folder if you like. Make sure the folder name has NO SPACES and NO PUNCTUATION (other than an underscore) and NO UPPERCASE LETTERS. (My folder is named *flaskbasic.*)

1. Log in at **Reclaim Hosting** (assuming that is where your domain is hosted).
2. Open **cPanel**.
3. Click this ("Setup Python App"):



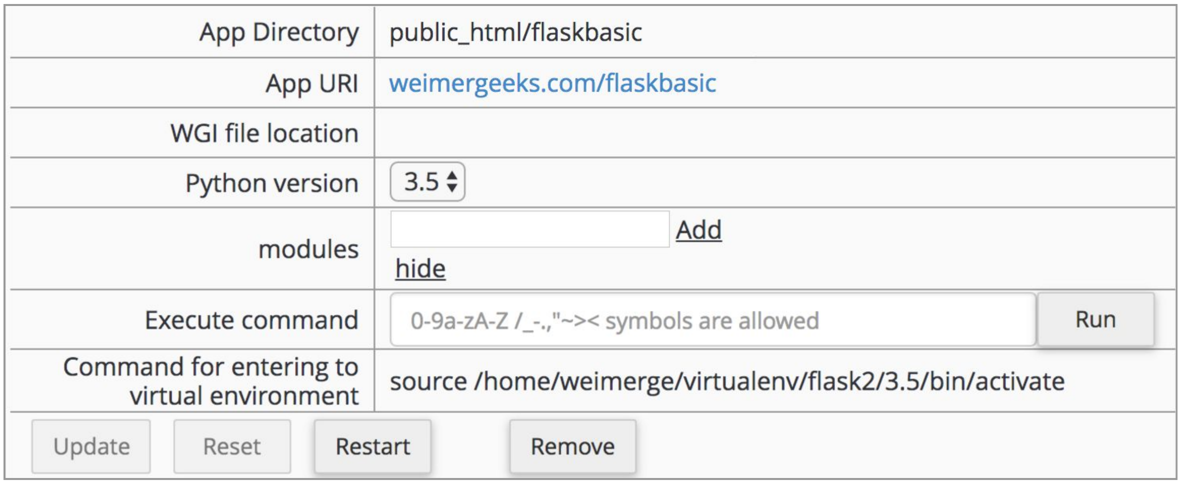
4. Choose **Python 3.5** and fill in the directory name *and* URI (all lowercase, no punctuation and no spaces) for the folder containing your app (my folder is named *flaskbasic*):



Setup new application

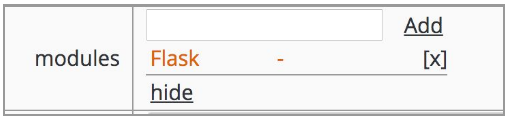
Python version	3.5	
App Directory /home/weimerge/	public_html/flaskbasic	
App Domain/URI	weimergeeks.com	flaskbasic

5. Click **Setup**. Afterward, a new panel appears below this one.
6. In the panel *below*, open the "modules" area (click **Show** to get this view):



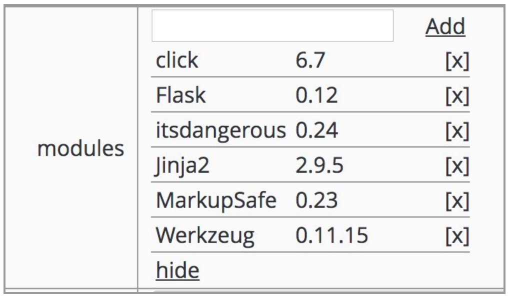
App Directory	public_html/flaskbasic
App URI	weimergeeks.com/flaskbasic
WGI file location	
Python version	3.5
modules	
Execute command	0-9a-zA-Z /_-.,"~>< symbols are allowed
Command for entering to virtual environment	source /home/weimerge/virtualenv/flask2/3.5/bin/activate

7. In the field to the right of "modules," type or paste Flask.
8. Click **Add**.
9. If it looks like this, click the **Update** button *below* this.



modules	Flask	-

10. *After it updates,* you'll see this:



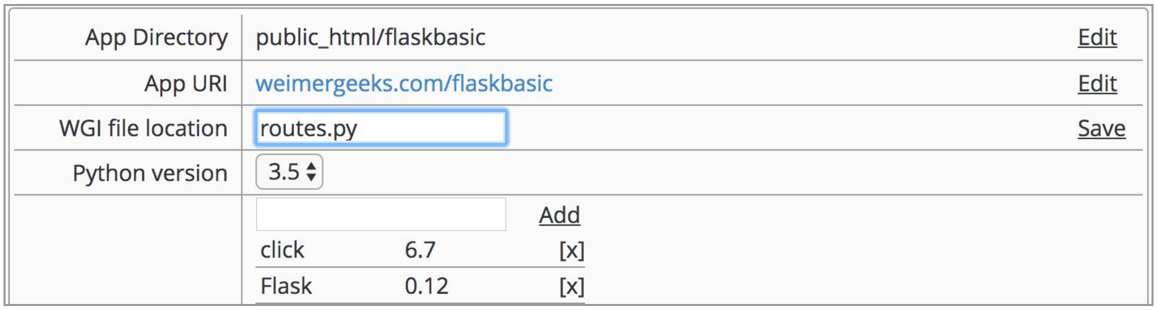
modules	click	6.7
	Flask	0.12
	itsdangerous	0.24
	Jinja2	2.9.5
	MarkupSafe	0.23
	Werkzeug	0.11.15

Those are dependencies for Flask — it needs them all to run. They were installed along with Flask automatically.

If you need *additional* dependencies for your app, such as Flask-WTF and Flask-Bootstrap, **add them** in the same way.

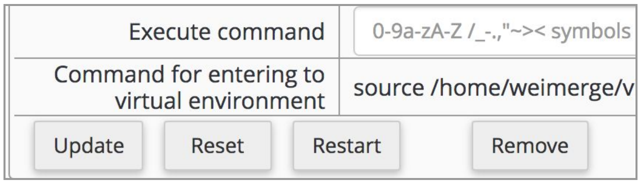
11. Did you mess up? You can instantly *delete* an app with the **Remove** button below. Warning: It does NOT ask if you are sure! Good news: It will not affect your uploaded files in the *flaskbasic* folder. It just removes the added bits from Reclaim.

12. Now you need to type in the **complete path** to your Python script file (the app file), in the field labeled "WGI file location" — and then **Save**.
THIS IS NEW: A complete path resembles the line below. Figure out yours.
`/home/weimerge/public_html/flaskbasic/routes.py:application`
^ Type that where it says `routes.py` below.



App Directory	public_html/flaskbasic		Edit
App URI	weimergeeks.com/flaskbasic		Edit
WGI file location	routes.py		Save
Python version	3.5		
		Add	
	click	6.7	[x]
	Flask	0.12	[x]

13. Afterward, be sure to click **Update**:

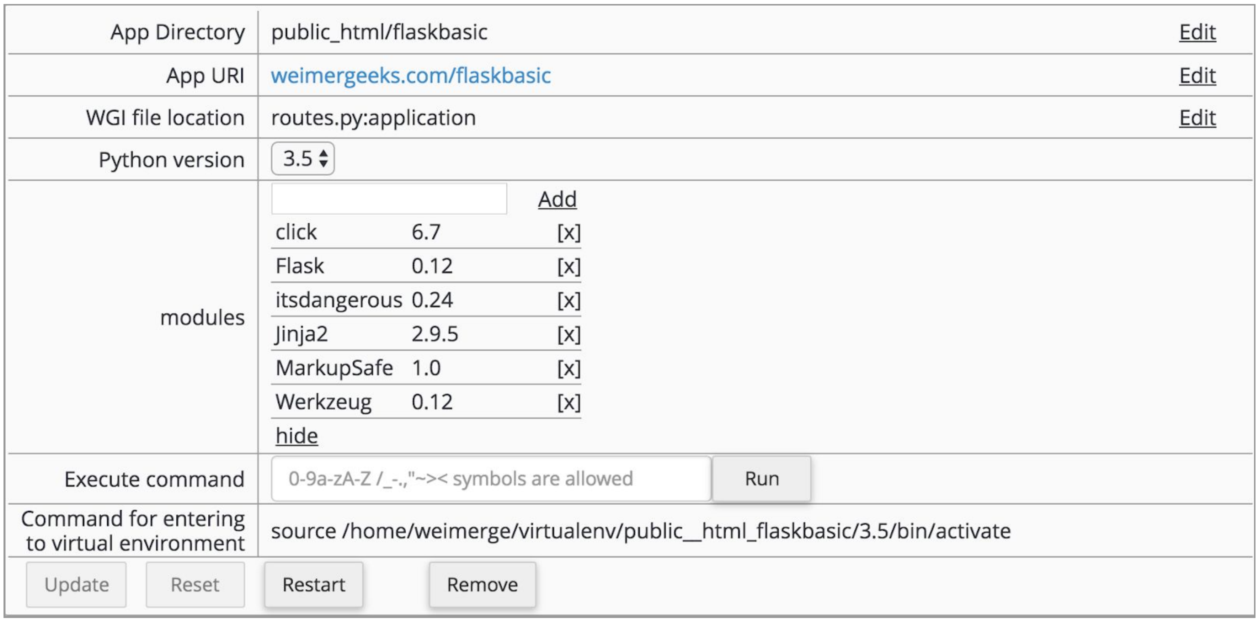


Execute command	0-9a-zA-Z /_-.,"~>< symbols
Command for entering to virtual environment	source /home/weimerge/v

This **rewrites** the file *passenger_wsgi.py* — which was inserted into the folder *flaskbasic* automatically when you updated before. If you ever need to change the filename of the Python file that runs your app, you can do steps 12 and 13 again.

You don't need anything in the field labeled "Execute command."

14. You should be able to see the Flask app functioning now in your browser. Go to *http://yourdomain.com/flaskbasic* (using your own folder name, of course). If it's not working yet, try clicking the **Restart** button in the panel shown above.



App Directory	public_html/flaskbasic	Edit
App URI	weimergeeks.com/flaskbasic	Edit
WGI file location	routes.py:application	Edit
Python version	3.5	
modules	click 6.7 [x] Flask 0.12 [x] itsdangerous 0.24 [x] Jinja2 2.9.5 [x] MarkupSafe 1.0 [x] Werkzeug 0.12 [x]	
Execute command	0-9a-zA-Z /_-.,"~>< symbols are allowed	
Command for entering to virtual environment	source /home/weimerge/virtualenv/public__html_flaskbasic/3.5/bin/activate	

Above: Final app configuration for a very basic Flask app.

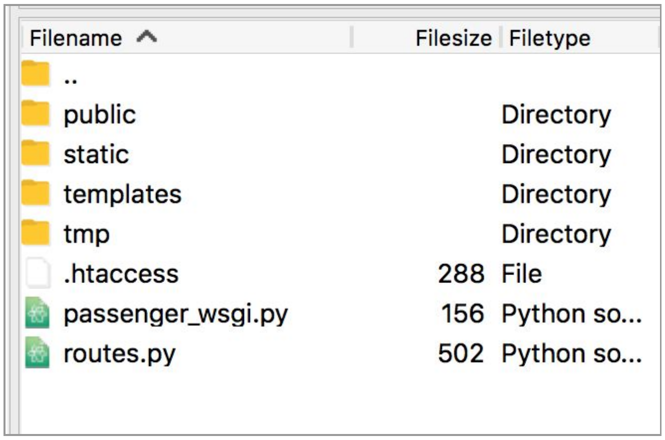
Other helpful facts

If you replace/update files in your app, you might need to **restart** it. In **cPanel,** open "Setup Python App" to get to the access panel for your app.

You can create *more than one app* using "Setup Python App." Reclaim does not appear to limit how many Python apps we can have.

Directory structure

Here's my *flaskbasic* directory on Reclaim, viewed in FileZilla, after the above steps:



Filename	Filesize	Filetype
..		
public		Directory
static		Directory
templates		Directory
tmp		Directory
.htaccess	288	File
passenger_wsgi.py	156	Python so...
routes.py	502	Python so...

These files and folders are part of **my Flask app,** which I uploaded:

/static
/templates
routes.py

These files and folders were **added by** the “Setup Python App” process:

/public
/tmp
.htaccess
passenger_wsgi.py

DO NOT delete or modify *.htaccess* or *passenger_wsgi.py*!

Not suitable for commercial projects

As this fine tutorial points out, “There are better ways to deploy your Flask application.” But as the author also notes, “However, great tools come with a price tag, and you might be already paying for **a shared hosting account.** So let's use that yo.” We have a shared hosting account at Reclaim, so that’s what we’re using. Yo.

Don’t include virtualenv folder

If you have a *virtualenv* or *env* folder inside your Flask app project, you should NOT upload that to your server. During the setup process shown on pages 1–3, a virtual environment is created for you on your server.

Some things that worked fine on your computer ...

… May not work at all on the live server. No doubt there are lots of different reasons for this, and if you run into a problem, you will just have to Google the heck out of it until you solve it.

An important tip: If you start changing the code, test it on your own computer first before you put it on the server. Running in debugging mode can give you some valuable clues.

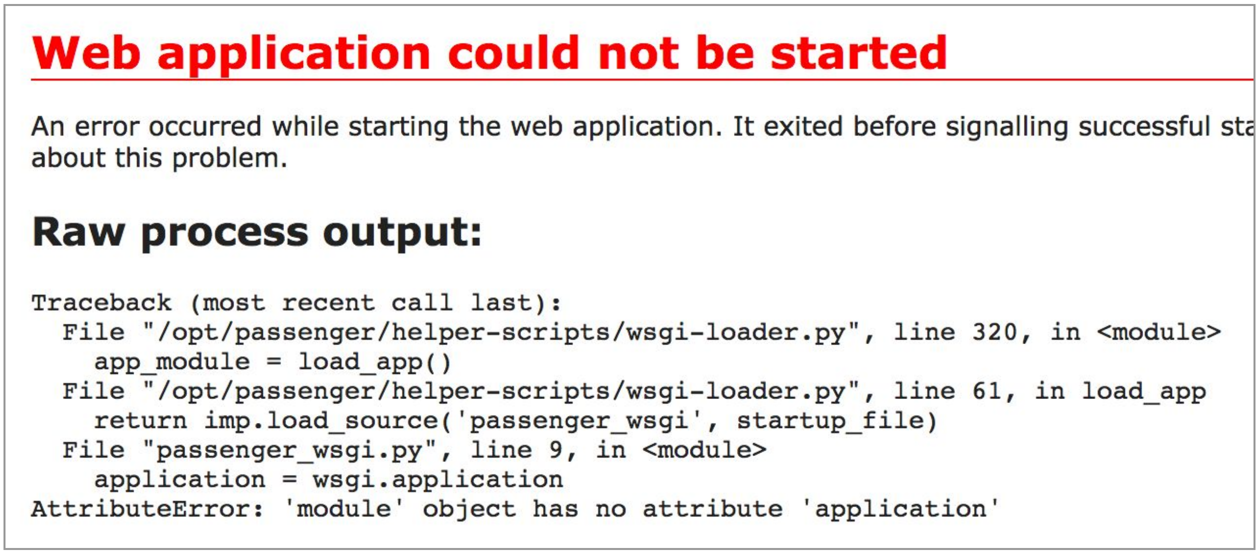
Why we add a line to the *.py* file

On page 1 of this document, I told you to add this line to the *.py* file that runs your Flask app:
`application = app`

I learned that from this fine tutorial. Without it, WSGI can't run your app. (What is WSGI?)

In the file *passenger_wsgi.py,* there is code that references *application*. You should not touch the code in *passenger_wsgi.py* — all you need to do is **add that one line** in your Python Flask script, *after* the line that says:
`app = Flask(__name__)`

If you forget to do this, you'll see this in the browser:



Web application could not be started

An error occurred while starting the web application. It exited before signalling successful sta
about this problem.

Raw process output:

```
Traceback (most recent call last):
  File "/opt/passenger/helper-scripts/wsgi-loader.py", line 320, in <module>
    app_module = load_app()
  File "/opt/passenger/helper-scripts/wsgi-loader.py", line 61, in load_app
    return imp.load_source('passenger_wsgi', startup_file)
  File "passenger_wsgi.py", line 9, in <module>
    application = wsgi.application
AttributeError: 'module' object has no attribute 'application'
```

Bitlink: http://bit.ly/flask-deploy
Updated March 22, 2018