

Flask Python (2 – Starting out)

Flask (set up)

Now that we've had a little introduction to HTML, Python and Bootstrap it's time to get into flasks.

So far, we have developed python mainly using an online development environment such as repl.it. When we needed more complex functionality (such as sprites in PyGame) we put a development environment on our own machines such as IDLE or PyCharm.

We haven't yet looked at database and they will come along in due course. But to put everything together and run servers. We can do this locally but to allow the outside world to see it it is probably best to use an online platform so we can have everything in one place and available. The other advantage is that the management of the servers and the ability to expand is easier dealt with by hosted solutions rather than doing all this ourselves.

So, we are going to use pythonanywhere. When you sign up below please use a sensible username as the link is visible to the world. In class we will look at whether we can have the educational package.

Step 1: Sign up for an account at <https://pythonanywhere.com> and here is your dashboard. The main links at the top left are the ones we will be needing. The Database option may or may not be required depending on the flavour of DB you want!

The screenshot shows the PythonAnywhere dashboard. At the top, there's a navigation bar with links: Dashboard, Consoles, Files, Web, Tasks, and Databases. A warning message states: "Warning: You have not confirmed your email address yet. This means that you will not be able to reset your password if you lose it. If you cannot find your confirmation email anymore, send yourself a new one here." Below this, the dashboard title "Dashboard" is on the left and "Welcome, dracstorey" is on the right. A status bar shows "CPU Usage: 2% used - 2.95s of 100s. Resets in 20 hours, 5 minutes" and "File storage: 0% full - 352.0 KB of your 512.0 MB quota". There's an "Upgrade Account" button. The main content area is divided into four panels: "Recent Consoles" (empty, with a "New console:" section below it showing "\$ Bash", ">>> Python", and "More..."), "Recent Files" (showing "/home/dracstorey/mysite/flask_app.py" with "Open another file" and "Browse files" buttons), "Recent Notebooks" (showing a message: "Your account does not support Jupyter Notebooks. Upgrade your account to get access!"), and "All Web apps" (showing "dracstorey.pythonanywhere.com" with an "Open Web tab" button).

Step 2: Click on **Web** and then **Add a new web app**, then select the **Flask** framework and **Python 3.8**. Open the configuration link for username.pythonanywhere.com in a new tab (right click) and note that you have a web page which has a message from flask. So, where's the code that did this?

Step 3: Go back to the dashboard and open the **Files** link. On the left you will see the directory *mysite*. Go into *mysite* and then click on the python file *flask_app.py*. You will see the code listed below:

```
# A very simple Flask Hello World app for you to get started
with...
```

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello from Flask!'
```

What this is doing is importing the flask module and using a decorator (@) to point to the root directory (*mysite*) and any result (`return`) from the function will be rendered in the browser.

Step 4: You can update the return value by adding html tags as well. Try putting the message in `<h2>` tags and using inline css to put the message in green.

Step 5: Now add the following code:

```
@app.route('/test')
def new_message():
    return 'Another message'
```

Step 6: Save the file and then reload the page. *Reloading is important as the process is always running, so you have to do this every time you make a change to the python code (but not the templates!).*

Step 7: In your browser type: <https://username.pythonanywhere.com/test>

This will now display the message “Another message”.