

Lightweight web application framework

Huh? What's the when it's at home?

A collection of libraries and modules that allow us to focus on routes and verbs and actual business logic rather than worrying about protocols, thread management and so on.

Flask vs Django

Easier to get started with Flask - not a *huge* learning curve
Flask is super explicit which improves readability
You make all the decisions with Flask - it's a micro-framework

Hello World

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def hello_world():
    return "<p>Hello World</p>"
```

1. First we imported the Flask class. An instance of this class will be our WSGI application.
2. Next we create an instance of this class. The first argument is the name of the application's module or package. `__name__` is a convenient shortcut for this that is appropriate for most cases. This is needed so that Flask knows where to look for resources such as templates and static files.
3. We then use the `route()` decorator to tell Flask what URL should trigger our function.
4. The function returns the message we want to display in the user's browser. The default content type is HTML, so HTML in the string will be rendered by the browser.

To run the application, run, where ``server`` is the filename:

```
flask --app server run
```

If the file was called `app.py` or `wsgi.py` then you don't need the `--app` filename piece.

If ``flask`` doesn't work in the shell, use **`python -m flask`**

You can also start a debugger service with **`flask --app server.py --debug run`**

Python's Request library has methods that we can use:

```
@app.route('/bbc')
def getBbc():
    return get("https://bbc.co.uk").content
```

This poses problems when the original request wants to fetch supplementary files:

```
@app.route('/<path:path>')
def fullBbc(path):
    return get(f'https://bbc.co.uk/{path}').content
```

One of the main uses though is querying APIs and hiding your credentials:

```
@app.route('/weather/<city>')
def getWeather(city):
    return post(f'https://api.openweathermap.org/data/2.5/weather?q={city}&appid={WEATHER_API_KEY}')
    ).content
```