

EN 1: Engineering in the Kitchen

Steven Bell

27 November 2023



Logistics

Continue working on project 4

Plan to have data collected by Wednesday
Figure + writeup in OneDrive by Friday

There will be streamlined options for the final project

Plan for "technical interviews" during the last week
(in person or via zoom)

Objectives

By the end of class today, you should be able to:

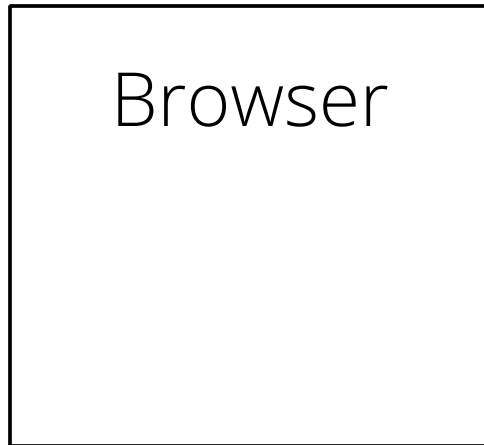
- Describe the functionality of a web server

- Write Python/microdot code to serve web pages from your ESP32

The old web

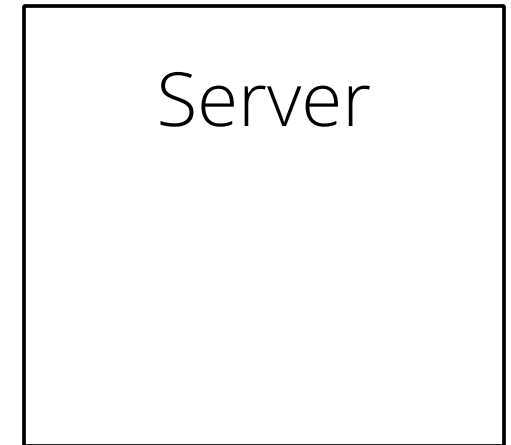
Servers have files

<http://www.ece.tufts.edu/en/1EK/esp32.html>



Can I have file?

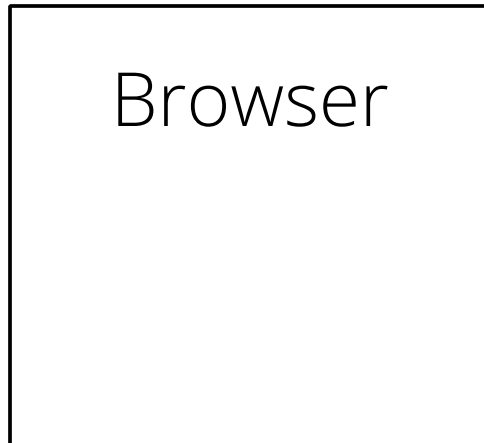
Sure, here it is!



The new web

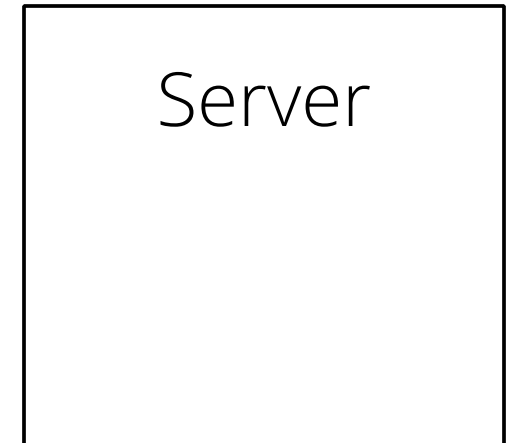
Servers have "endpoints"

<https://vhdlweb.com/problem/count5>



Can I have file?

Sure, here it is!



Microdot

Is a ultra-lightweight web server framework

Flask is similar but more full-featured for "real servers"

```
app = Microdot()
```

```
@app.route('/')
```

```
def index(request):
```

```
    return "Look, this is my web server! Isn't it cool?!"
```

```
app.run(port=80, debug=True)
```

Running Microdot

Copy `microdot.py` and `simpletemplate.py` to your ESP32

Download `webstarter.py` and open it in Thonny

Connect your ESP32 to WiFi (just like previous classes)

Run `webstarter.py`

Open a web browser and type in the ESP32's IP address

Challenge: add another endpoint (or two!)

Challenge: Include a count on the page of how many times it has been accessed (this was super cool in the late 90s)