Introduction to Flask

Publishing an Artist Classifier
Trained on Song Lyrics

Where we’re going!

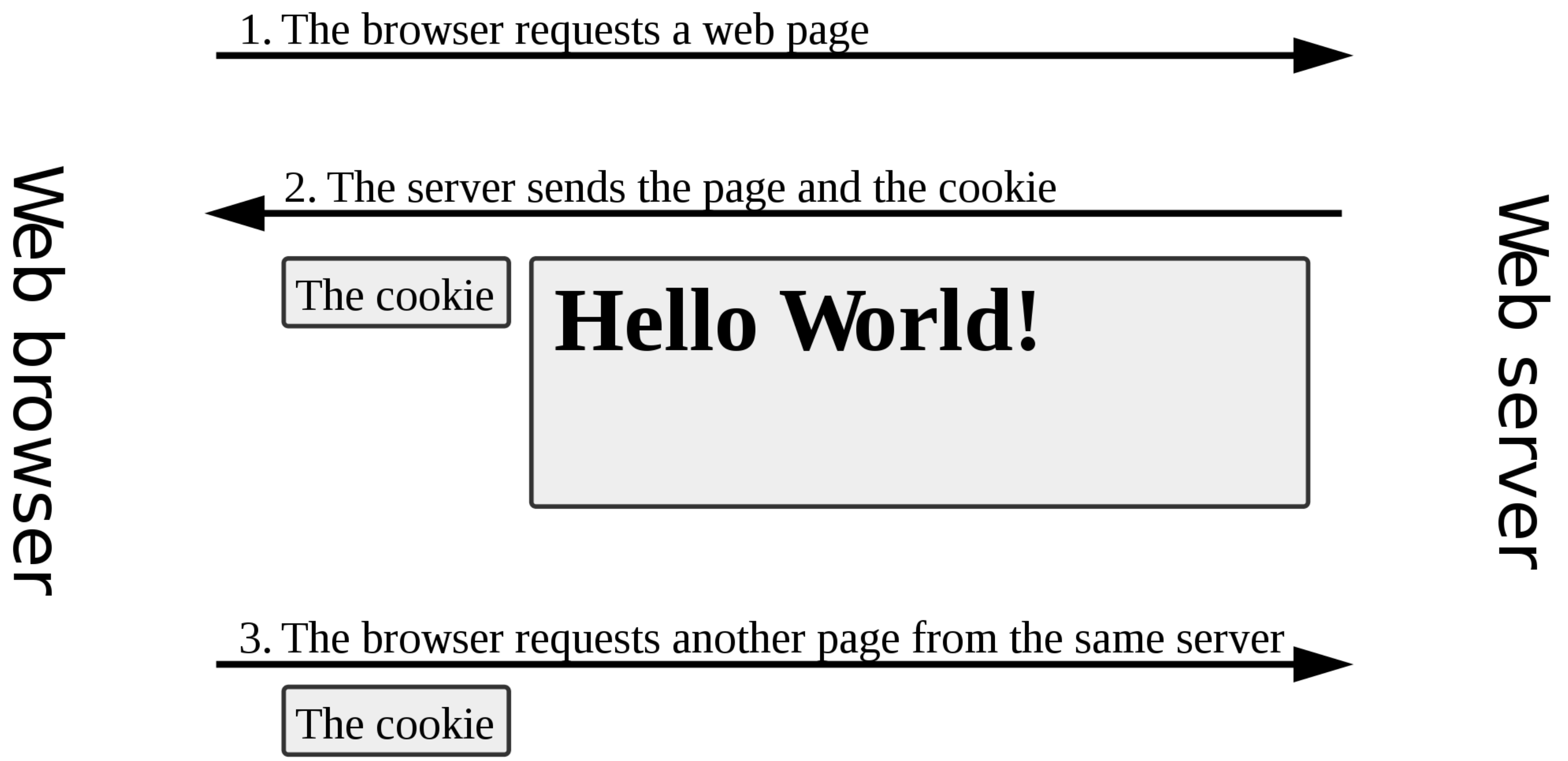
Thomas Bayes

+

=

Web Terminology

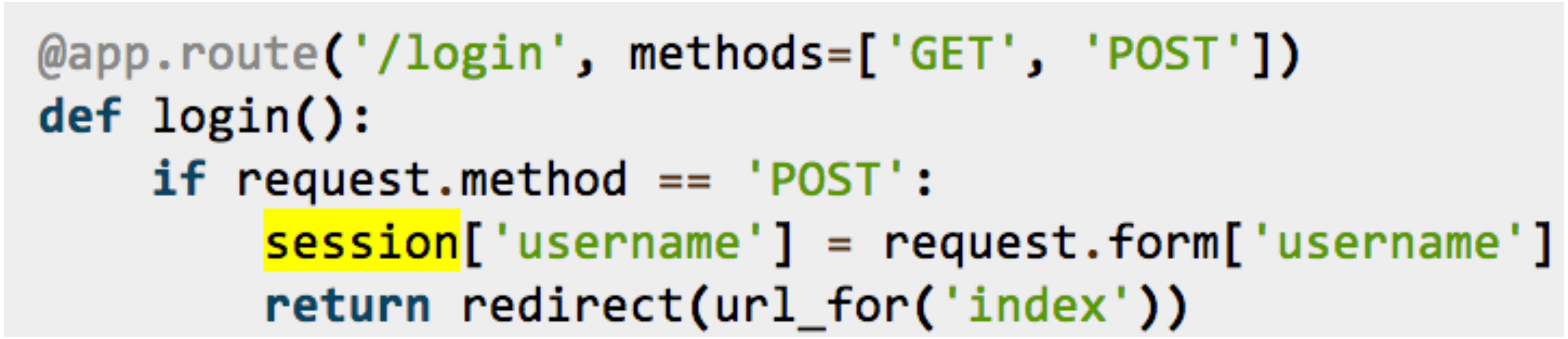
Cookies


1. The browser requests a web page
2. The server sends the page and the cookie
The cookie
Hello World!
3. The browser requests another page from the same server
The cookie
Web browser
Web server

Sessions

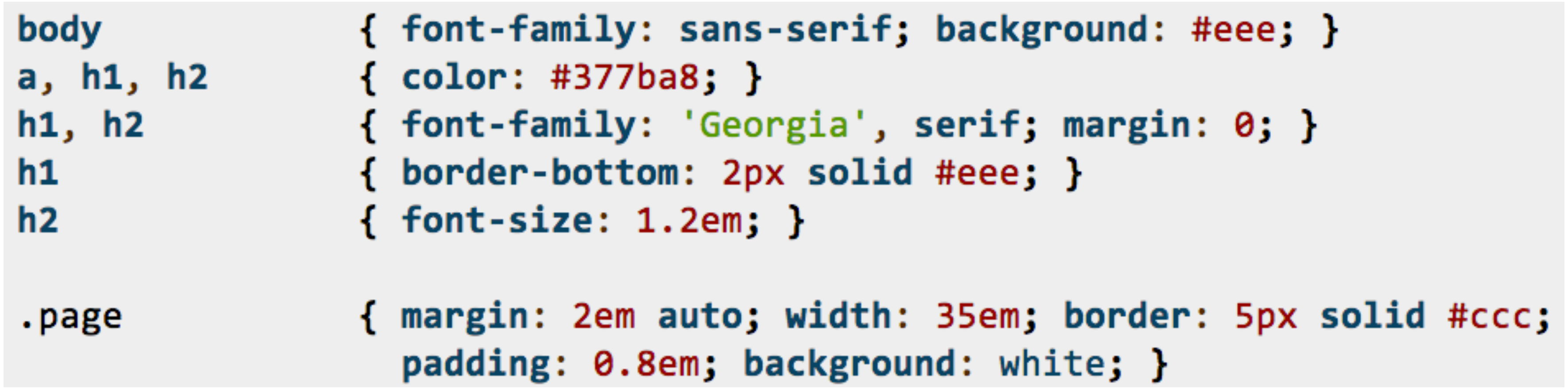
Flask provides a *session* object

- built on top of cookies, cryptographically secure
- critical data only stored on the server



```
@app.route('/login', methods=['GET', 'POST'])
def login():
    if request.method == 'POST':
        session['username'] = request.form['username']
        return redirect(url_for('index'))
```

Cascading Style Sheets



```
body            { font-family: sans-serif; background: #eee; }
a, h1, h2       { color: #377ba8; }
h1, h2          { font-family: 'Georgia', serif; margin: 0; }
h1              { border-bottom: 2px solid #eee; }
h2              { font-size: 1.2em; }

.page           { margin: 2em auto; width: 35em; border: 5px solid #ccc;
                  padding: 0.8em; background: white; }
```

Flask Terminology

Templates

```
{% extends "layout.html" %}
{% block body %}
  <ul>
  {% for user in users %}
    <li><a href="{{ user.url }}">{{ user.username }}</a></li>
  {% endfor %}
  </ul>
{% endblock %}
```

Template engine (Jinja2) inserts dynamic content before rendering the page

Contexts

Application Context

- The initial state prior to servicing requests
- Safe to load configuration, etc.

Request Context

- The functional/online state of your application
- Application context code is not re-run

Message Flashing

- User feedback is important!

- Flask provides a flash(“msg”) routine
  - The messages are made pretty by CSS

Database Terminology

Schema

```
CREATE TABLE entries (
    id integer primary key autoincrement,
    lyrics text not null,
    artist text not null
);
```

Python Terminology

Pickle

pickle.dump( ) → object *serialization*

- object is converted to a series of strings (*ish*)
- strings are written to disk for later use

pickle.load( ) → object *deserialization*

- object, read from disk, is recreated in memory

Decorators

(What's that funny @ doing?)

```
def mydecorator(func):
    def addone(x):
        return func(x) + 1
    return addone
```

```
@mydecorator
def foo(x):
    return x * 2

>>> print foo(3)
?
```

Decorators

```
def mydecorator(func):
    def addone(x):
        return func(x) + 1
    return addone
```

```
@mydecorator
def foo(x):
    return x * 2

>>> print foo(3)
7
```

Code-along Time!