

# **HTML TEMPLATES IN FLASK**

**USING JINJA 2**

# SEPARATION OF CONCERNS



SCALABILITY AND  
MAINTAINABILITY



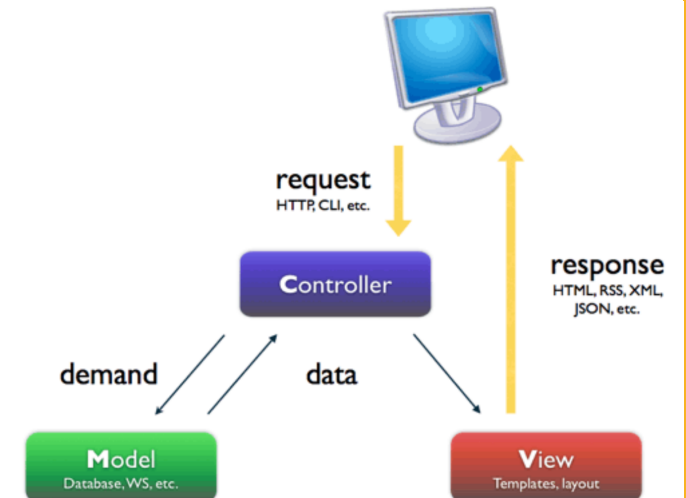
SEPARATE HTML, CSS,  
AND JAVASCRIPT FILES



REQUEST HANDLING  
CODE VS. HTML  
DOCUMENTS



MVC



# JINJA 2 TEMPLATES



It's HTML, with placeholders

```
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8">
  <title>Hello</title>
</head>

<body>
  <h1> Hello [[name]]! </h1>
</body>

</html>
```



Jinja2 → Flask default and most commonly used template system

Your website will likely rely on some static files such as...

## STATIC FILES

Jinja is a template system for HTML only



CSS documents



JavaScript source



image files

*These should be saved in a "static" directory to be automatically accessible.*

# TEMPLATE EXAMPLE

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route("/", defaults={"name": "Stranger"}, methods=["GET"])
@app.route("/<string:name>", methods=["GET"])
def hello(name):
    return render_template("hello.html", name=name)
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <title> hello </title>
</head>
<body>
    <h1> Hello {{name}}! </h1>
</body>
</html>
```

# TODAY: WORKSHOP

- Create multi-age HTML templates with Jinja 2 ...?
  - Limited to URL input only (GET method) ...we'll build forms next week so we can use POST
  - Use Bootstrap features
  - Use template inheritance ("*server-side includes*")
    - {block} and {extend}