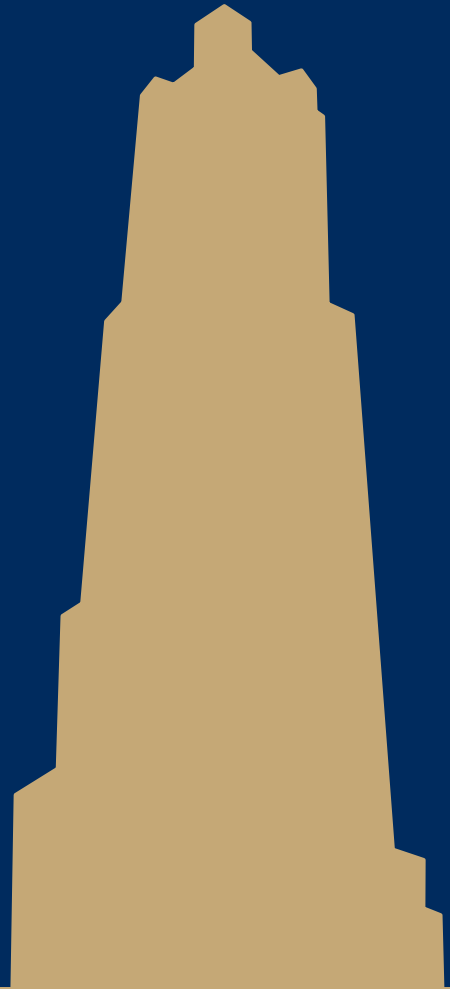


CS/COE 1520

pitt.edu/~ach54/cs1520

Using Templates to Generate Views in Flask



Jinja2 templating language

- Jinja templates are simply text files
 - Include Jinja tags that will be expanded when the template is rendered
 - Here, we'll focus on using Jinja to produce HTML, however, it can be used to generate any kind of text document
- Flask configures the Jinja template engine for use within web apps automatically

Basic Jinja tags

- `{{ ... }}`
 - Expression tag, contents are evaluated and place in the text
- `{% ... %}`
 - Statement tag, used to define Jinja constructs and issue flow control statements
- `{# ... #}`
 - Comment

Simple Jinja tag example

```
<!doctype html>
<html>
  <head>
    <title>Hello from Flask</title>
  </head>
  <body>
    {% if name %}
      <h1>Hello {{ name }}!</h1>
    {% else %}
      <h1>Hello, World!</h1>
    {% endif %}
  </body>
</html>
```

Using Jinja templates within Flask

- The `render_template(template_name, arguments ...)` function is used to grab a template file, and pass data to be used in generating a view of the page (e.g., `name`)
 - By *convention*, Flask will look for templates in a directory simply called "templates"
 - In addition to `arguments`, the `session` and `request` Flask vars can be referenced in templates
 - And, additionally, others we have yet to discuss

Control structures

- Operate similarly to their Python variants
 - `{% if cond %} {% elif cond %} {% else %} {% endif %}`
 - Only render a part of the template if some condition is met
 - E.g., display logout link if a user is logged in
 - `{% for i in seq %} {% endfor %}`
 - Render some part of the template multiple times for each item in a given sequence
 - E.g., create a div for each of a user's blog posts

Template Inheritance

- Define blocks of a template that can be overridden in subtemplates
 - `{% block name %}`
 - `{% endblock %}`
- Establish inheritance through the `{% extends base %}` tag

Message flashing

- In our examples, it would be handy to let the user know they've been successfully logged in
 - Or give notice that there was an error in their attempt...
- Message flashing is built to handle this type of feedback!
- `flash(msg)` adds `msg` to `session["_flashes"]`
 - Can be retrieved with `get_flashed_messages()`
 - Accessible within templates!