Python | Flask

Session 3

- Template Inheritance
- Cookies
- Example

by Mohammad Amin H.B. Tehrani

www.maktabsharif.ir

Template Inheritance

Template Inheritance

Intro

The most powerful part of Jinja is template inheritance. Template inheritance allows you to build a base “skeleton” template that contains all the common elements of your site and defines blocks that child templates can override.

Sounds complicated but is very basic. It’s easiest to understand it by starting with an example.

Template Inheritance

Base Template

This template, which we'll call base.html, defines a simple HTML skeleton document that you might use for a simple two-column page. It's the job of "child" templates to fill the empty blocks with content:

```
<!DOCTYPE html>
<html lang="en">
<head>
    {% block head %}
        <link rel="stylesheet" href="style.css" />
        <title>{% block title %}{% endblock %} - My Webpage</title>
    {% endblock %}
</head>
<body>
<div id="content">{% block content %}{% endblock %}</div>
<div id="footer">
    {% block footer %}
        &copy; Copyright 2008 by <a href="http://domain.invalid/">you</a>.
    {% endblock %}
</div>
</body>
</html>
```

In this example, the {% block %} tags define four blocks that child templates can fill in. All the block tag does is tell the template engine that a child template may override those placeholders in the template.

block tags can be inside other blocks such as if, but they will always be executed regardless of if the if block is actually rendered.

Template Inheritance

Child Template

A child template might look like this:

```
{% extends "base.html" %}
{% block title %}Index{% endblock %}
{% block head %}
   {{ super() }}
   <style type="text/css">
       .important { color: #336699; }
   </style>
{% endblock %}
{% block content %}
   <h1>Index</h1>
   <p class="important">
       Welcome to my awesome homepage.
   </p>
{% endblock %}
```

The {% extends %} tag is the key here. It tells the template engine that this template “extends” another template. When the template system evaluates this template, it first locates the parent. The extends tag should be the first tag in the template. Everything before it is printed out normally and may cause confusion.

Template Inheritance

Super Blocks

It’s possible to render the contents of the parent block by calling super(). This gives back the results of the parent block:

```
{% block sidebar %}
   <h3>Table Of Contents</h3>
   ...
   {{ super() }}
{% endblock %}
```

If you want to print a block multiple times, you can, however, use the special self variable and call the block with that name:

```
<title>{% block title %}{% endblock %}</title>
<h1>{{ self.title() }}</h1>
{% block body %}{% endblock %}
```

Template Inheritance

Include

The include tag is useful to include a template and return the rendered contents of that file into the current namespace:

```
{% include 'header.html' %}
Body
{% include 'footer.html' %}
```

This behavior can be changed explicitly: by adding **with context** or **without context** to the include directive, the current context can be passed to the template and caching is disabled automatically.

```
{% set x = 100 %}
{% include 'header.html' with context %}
{% include 'footer.html' without context %}
```

Cookies

Cookies

Intro

HTTP is Stateless: "does not require the HTTP server to retain information or status about each user for the duration of multiple requests."

HTTP cookies (also called web cookies, Internet cookies, browser cookies, or simply cookies) are small blocks of data created by a web server while a user is browsing a website and placed on the user's computer or other device by the user’s web browser.

Cookies

Set-Cookies

You should use Flask Response object, then **.set_cookie(..)** method, to set specific cookie to response.
Also you can set:

1. **max_age:** should be a number of seconds.
2. **expires:** should be a `datetime` object or UNIX timestamp.
3. **secure:** If ``True``, the cookie will only be available via HTTPS.
4. **httponly:** Disallow JavaScript access to the cookie.
5. ...

```
@app.route('/cookie-example', methods=['POST'])
def cookies_post():
    resp = Response("Setting 'TEST' cookie...")
    resp.set_cookie('TEST', 'Akbar', expires=datetime.utcnow() +
                                             timedelta(minutes=1))
    return resp
```

Cookies

Get-Cookies

You can get request cookies using **request.cookies** -> **MultiDict**

```
@app.route('/cookie-example', methods=['GET'])
def cookies_get():
    cookies = request.cookies
    print(cookies)
    resp = Response("'TEST' cookie value: " + cookies.get('TEST', "Not set!"))
    return resp
```

Cookies

Delete-Cookies

```
@app.route('/cookie-example', methods=['DELETE'])
def cookies_delete():
    resp = Response("Deleting 'TEST' cookie...")
    resp.delete_cookie('TEST')
    return resp
```

Cookies

make_response() utility function

Sometimes it is necessary to set additional headers in a view. Because views do not have to return response objects but can return a value that is converted into a response object by Flask itself, it becomes tricky to add headers to it. This function can be called instead of using a return and you will get a response object which you can use to attach headers.

```
from flask import render_template, make_response, ...

@app.route('/')
def index():
   html_str = render_template('index.html')  # type = str
   resp = make_response(html_str)  # type = Flask.Response class
   resp.set_cookie('Test', 'Akbar')
   return resp
```

Example

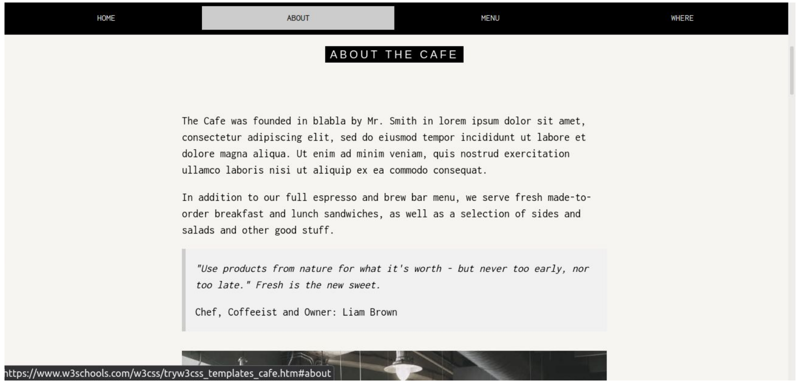
Cookies

Practice: Cafe page

Create a web page for your cafe:

- Use **Jinja Template inheritance** to reuse navbar division in all pages.
- Use **Cookies**, to save client orders into the cart.

(W3 Template: Cafe template)



ABOUT THE CAFE

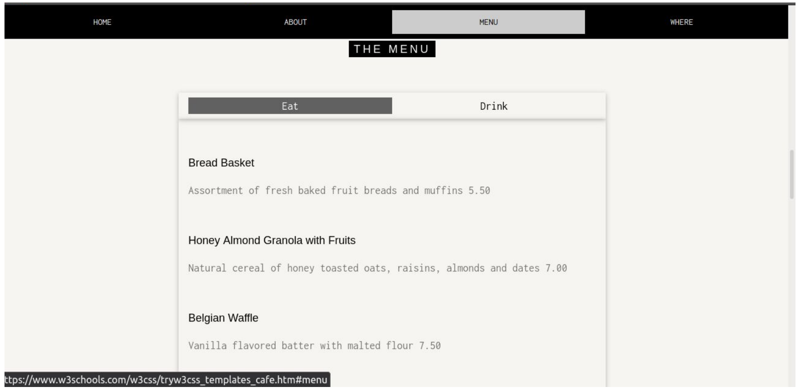
The Cafe was founded in blabla by Mr. Smith in lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

In addition to our full espresso and brew bar menu, we serve fresh made-to-order breakfast and lunch sandwiches, as well as a selection of sides and salads and other good stuff.

"Use products from nature for what it's worth - but never too early, nor too late." Fresh is the new sweet.

Chef, Coffeeist and Owner: Liam Brown

ttps://www.w3schools.com/w3css/tryw3css_templates_cafe.htm#about



THE MENU

Bread Basket

Assortment of fresh baked fruit breads and muffins 5.50

Honey Almond Granola with Fruits

Natural cereal of honey toasted oats, raisins, almonds and dates 7.00

Belgian Waffle

Vanilla flavored batter with malted flour 7.50

ttps://www.w3schools.com/w3css/tryw3css_templates_cafe.htm#menu