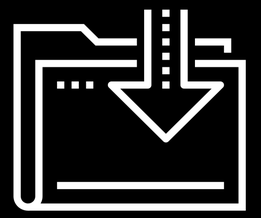
HTML Review & Flask

Data Boot Camp

Office Hours - October 23, 2021



HTML Review

HTML Review

Tags and Attributes

Tags

Used to mark up the start of an HTML element and they are usually enclosed in angle brackets.

Most tags must be opened and closed in order to function.

```
<h1></h1>
<p></p>
```

Attributes

Contain additional pieces of information and are placed inside an opening tag.

```
<img src="mycat.jpg" alt="A photo of my cat.">
```

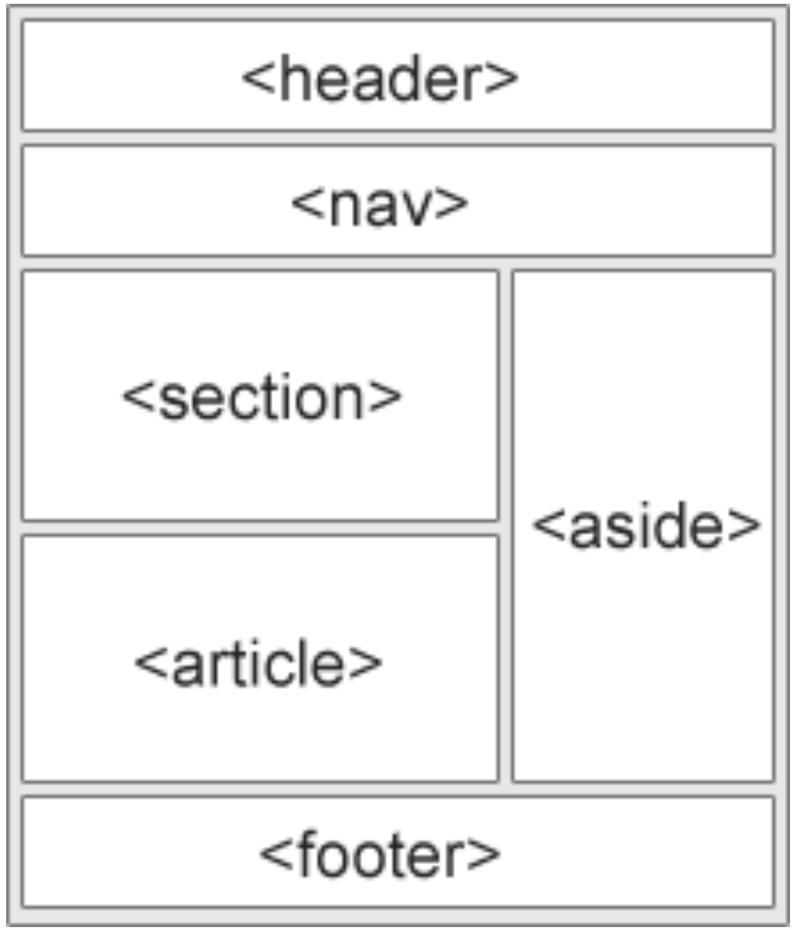
The image source (src) and the alt text (alt) are attributes of the <img> tag.

HTML Review

Semantic Elements

A semantic element clearly describes its meaning to both the browser and the developer.

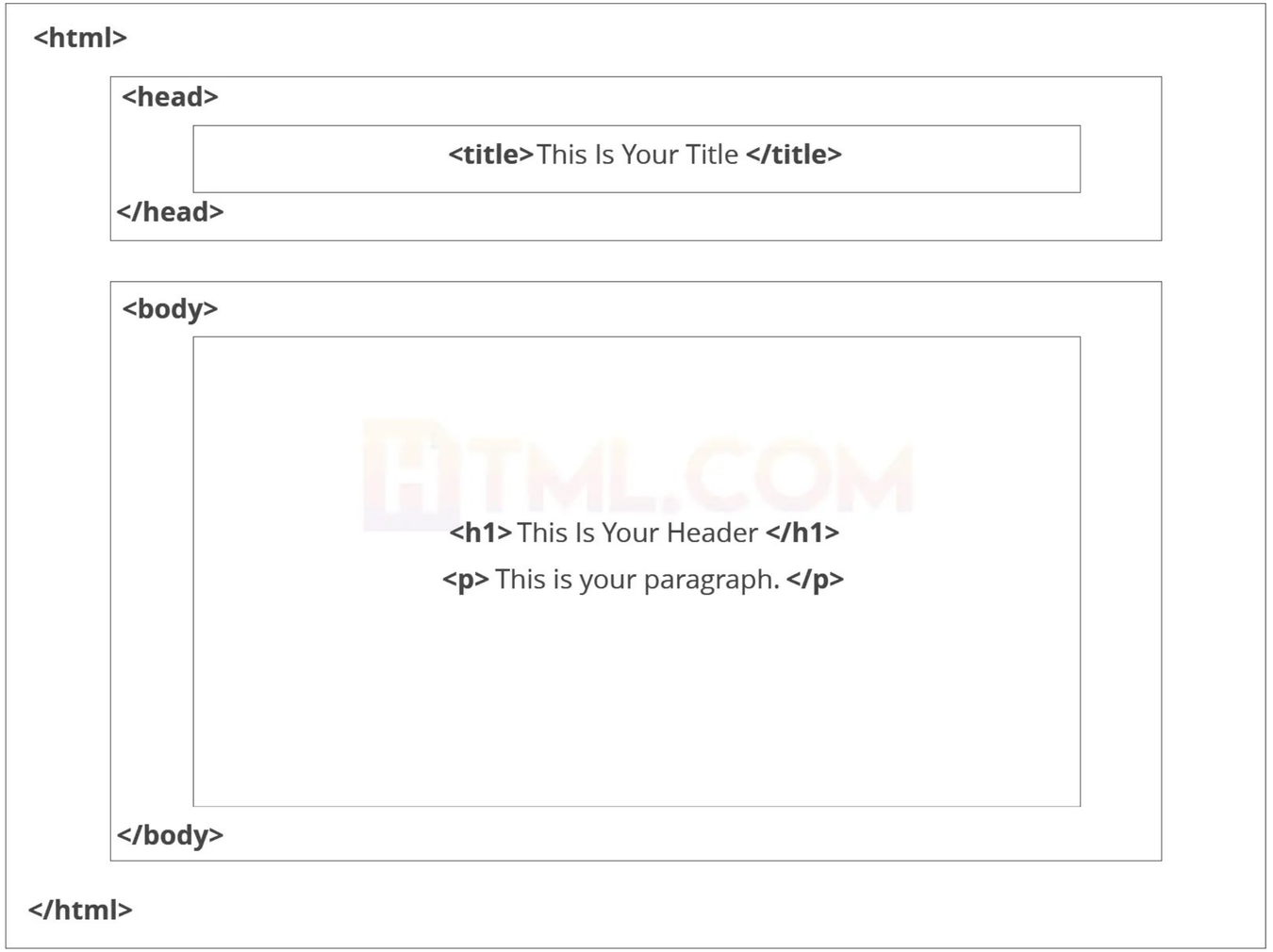
<article>	Defines independent, self-contained content
<aside>	Defines content aside from the page content
<details>	Defines additional details that the user can view or hide
<figcaption>	Defines a caption for a <figure> element
<figure>	Specifies self-contained content, like illustrations, diagrams, photos, etc.
<footer>	Defines a footer for a document or section
<header>	Specifies a header for a document or section
<main>	Specifies the main content of a document
<mark>	Defines marked/highlighted text
<nav>	Defines navigation links
<section>	Defines a section in a document
<summary>	Defines a visible heading for a <details> element
<time>	Defines a date/time



HTML Review

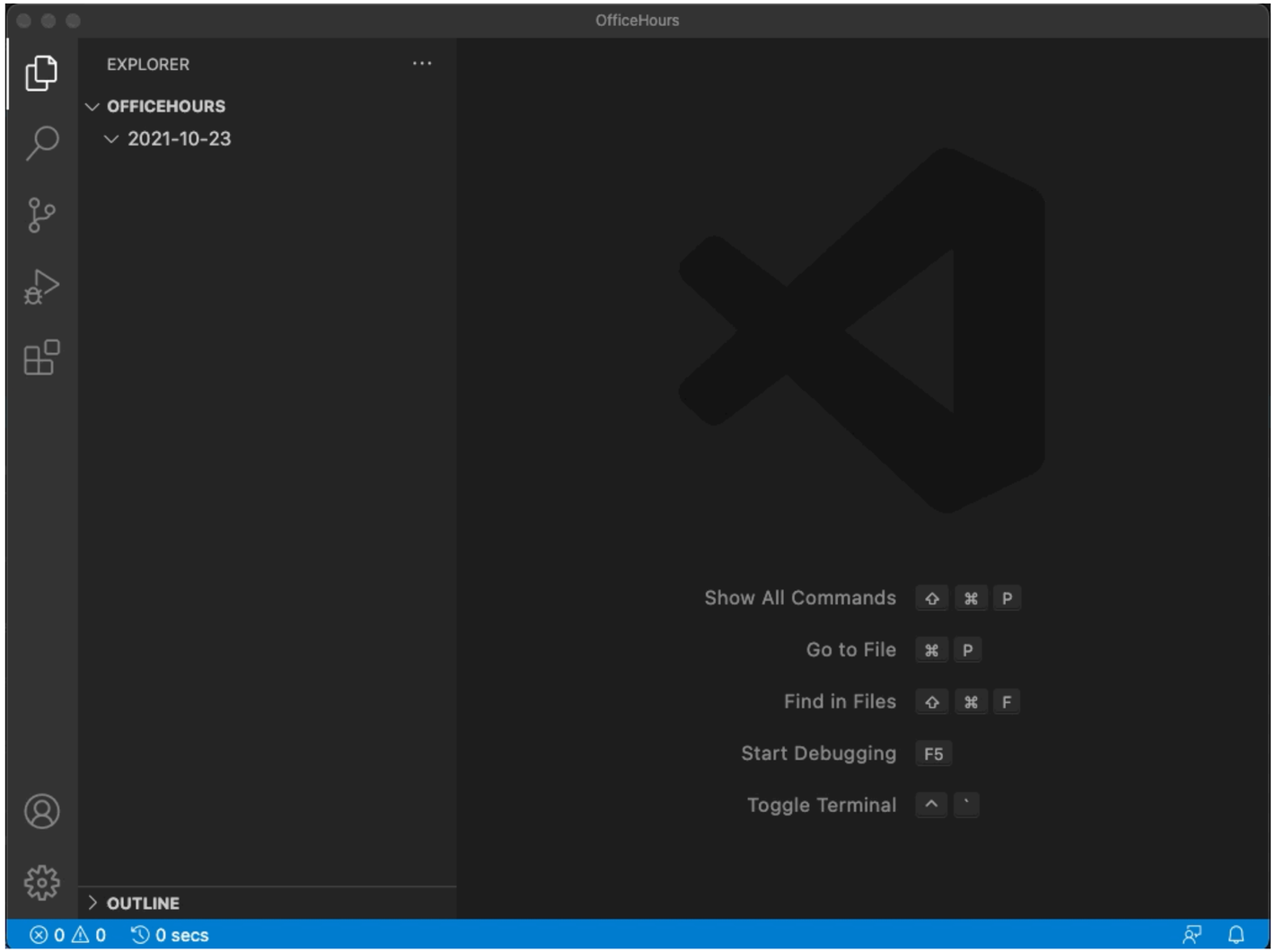
Basic Construction of an HTML Page

```
<!DOCTYPE html>     - Specifies the language
<html>     - Signals HTML code start
  <head>      - Contains metadata for the page
    <title>This Is Your Title</title>
  </head>
  <body>      - Page content
    <h1>This is your Header</h1>
    <p>This is your paragraph.</p>
  </body>
</html>
```



HTML Review

HTML Boilerplate in VSCode



When working with an HTML file you can expand an HTML Boilerplate snippet to quickly get started.

Type **html** anywhere inside an HTML file and select **html:5** from the menu selection.

HTML Review

HTML Boilerplate in VSCode

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
</head>
<body>

</body>
</html>
```

HTML Review

HTML Boilerplate in VSCode

<html lang="en">
The lang attribute inside the opening <html> tag sets the language for the page. It is important to include it for accessibility reasons, because it tells screen readers how to properly pronounce the text.

<meta charset="UTF-8">
UTF-8 is the standard character encoding you should use in your web pages. This will normally be the first <meta> tag shown in the <head> element.

<meta http-equiv="X-UA-Compatible" content="IE=edge">
This <meta> tag specifies the document mode for Internet Explorer. IE=edge is the highest supported mode.

<meta name="viewport" content="width=device-width, initial-scale=1.0">
This tag renders the width of the page to the width of the device's screen size. If you have a mobile device that is 600 px wide, then the browser window will also be 600px wide. The initial-scale controls the zoom level. The value of 1.0 for the initial-scale prevents the default zoom by browsers.

HTML Review

Accessibility

Accessible Rich Internet Applications **(ARIA)**

A set of attributes that define ways to make web content and web applications (especially those developed with JavaScript) more accessible to people with disabilities.

Web Content Accessibility Guidelines **(WCAG)**

A single shared standard for web content accessibility that meets the needs of individuals, organizations, and governments internationally.

Flask

Flask

Static Files

CSS and Javascript files for your Flask application can be served from a folder called static.

The files have to be stored on the filesystem as static/*.*

```
<link rel="stylesheet" href="{{ url_for('static', filename='style.css') }}">
```

Flask

Base Layout

Each page in the application will have the same basic layout around a different body.

Instead of writing the entire HTML structure in each template, each template will *extend* a base template and override specific sections.

Inside your base template use blocks to indicate where content will be replaced by other html files.

```
{% block content %}{% endblock %}
```

Tell Jinja that a template should replace the blocks from the base template.

```
{% extends 'base.html' %}
```

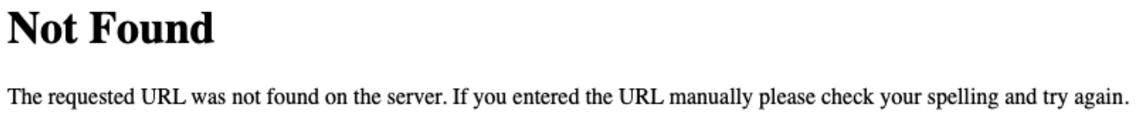
A useful pattern used here is to place `{% block title %}` inside `{% block header %}`. This will set the title block and then output the value of it into the header block, so that both the window and page share the same title without writing it twice.

Flask

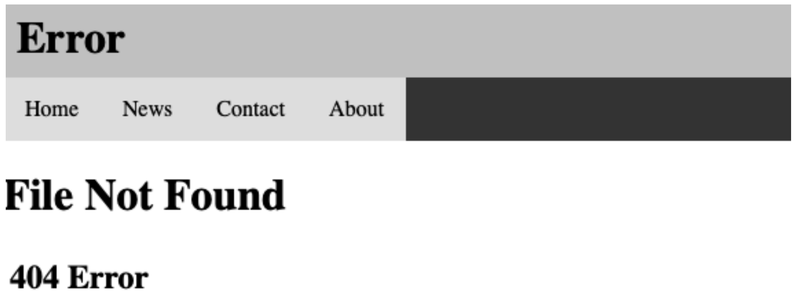
Error Handling

Customize your error pages and keep your base layout when displaying errors.

Before



After




```python
from flask import render_template

@app.errorhandler(404)
def page_not_found(e):
    # note that we set the 404 status explicitly
    return render_template('404.html'), 404
```

Resource Links

HTML
https://www.freecodecamp.org/news/basic-html5-template-boilerplate-code-example/
https://www.freecodecamp.org/news/semantic-html5-elements/

Accessibility
https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA#videos
https://www.w3.org/WAI/standards-guidelines/wcag/

Flask
https://flask.palletsprojects.com/en/2.0.x/tutorial/templates/
https://flask.palletsprojects.com/en/2.0.x/errorhandling/#custom-error-pages

Questions?