Week 3 Study Guide

- What are HTML elements?
  - HTML elements are used to structure a webpage. Some have an opening and closing tag (ex:) and others are self-closing (<img src="url.jpg">). Those that have an opening and closing tag often have text in-between them.
- What are HTML element attributes? What are some examples of HTML attributes and what do they do?
  - HTML element attributes are used to configure elements. Not including them often uses their default values. Some examples include name(identifies an element), href(provides a URL to link to), and required(specifics whether an input is required or not).
- In an HTML element, what does the ***id*** attribute do? The ***class*** attribute?
  - The id attribute is used to identify a unique element. The class attribute is used to categorize an element. These attributes are frequently used to style elements are manipulate them with the DOM.
- What is an inline element? A block element? What are some examples of each?
  - An inline element will sit on a line together with the elements around it and won't break the flow of your page/document. Examples include , <emphasis>, <img>, and .
  - A block element starts on a new line and take up the whole width of their containing element. Examples include , , <table>, headings, and lists.
- How do semantic HTML elements help developers markup content?
  - Using semantic HTML elements helps developers as it truly ensure HTML is being used for presentation and

CSS is being use for style. It also makes it a lot easier to distinguish what is on the web page and the importance of each element. Semantic HTML makes more sense and increases accessibility as it is easier for screen readers to process.

- Usage of specific elements:
  - How are ***div*** elements used? How do they differ from elements like ***article*** or ***p***?
    - A div is a generic block element that can be used when no other HTML element makes semantic sense for the content. They can be used to separate sections of HTML code. Div elements differ from article or p as those elements are specifically used for content that has semantic meaning related to the tag usage.
  - How are ***span*** elements used? How do they differ from elements like ***a*** or ***b***?
    - A span is a generic inline element that can be used when no other HTML element makes semantic sense for the content. They can be used to separate sections of HTML code, such as for styling purposes. Inline elements differ from a or b as those elements are specifically used for content that has semantic meaning related to the tag usage, in this case, making a link or bolding text.
- Creating forms in HTML:
  - In ***form*** elements, what is the ***method*** attribute used to specify? The ***action*** attribute?
    - The method attribute is used to specific how the form data will be utilized after it has been submitted: get or post. The action attribute specifics what page the form will be redirected to on the submission of the form.

    - List the various types of ***input*** tags. How do you set an ***input*** tag's type?
        - Types include: text, textarea, checkbox, radio, password
        - You set an input tag's type like so: input="text"
    - For an ***input*** tag, what does the ***name*** attribute do?
        - The name attribute is used to identify that specific input, so that you may use the value later, such as displaying them once submitted.
- What is the Order of Precedence in CSS?
    - From higher to lower: ID, Class, Element
- When you type a URL into a web browser, what happens?
    - Be able to explain this in your own words, hitting each main concept from the ***How the Web Works*** lecture.
    - When you type a URL into a web browser that DNS takes that URL and translates it into its IP address that it is able to understand. The client makes a request to the server for a particular resource and it may need to call on a database to retrieve information to display. Then this all gets returned to the client and the resource is displayed after rendering multiple files asynchronously, such as image files, css files, html files, and programming files.
- What is the ***request*** object in Flask? What can you use it for?
    - The request object in Flask allows for the response of request made by the user upon completing an action, such as submitting a form. You can use it to retrieve values entered on input fields, so that you can use that information for other purposes like displaying it or storing it in a database.
- In Flask, how do you access form values for a POST request? How about for a GET request?
    - For a POST request: request.form.get('person')

    - For a GET request: request.args.get('person')
- What is a cookie? What is a session? How are the two different?
    - Cookies are a way to store small bits of info on client (browser). Cookies are name/string-value pairs.
    - Flask sessions are a “magic dictionary” They contain info for the current browser, preserve type (lists stay lists, etc), and are encrypted automatically, so users can’t snoop.
    - Cookies are stored on the client for a specific period of time, where as sessions are saved for the duration of a session. Cookies are also limited in what they can store and are only strings, where as with sessions you can store information of various type.
- What is Jinja? What does the function ***render_template*** do?
    - Jinja is a popular template system for Python, used by Flask.
    - The function **render_template** displays a specified HTML file.
- Jinja syntax to know
    - For-loops - {% for item in items%} {% endfor %}
    - If-statements - {% if compliment %} {% endif %}
    - Showing text with Jinja - {{variable}}
    - Setting HTML attributes using Jinja -
    - Template inheritance (using a ***base.html*** to create consistency) - {% block content %} {% endblock %}
- What is a web-based API?
    - An API is an Application Programming Interface. It is meant to be used by developers to get data to build applications.
- What is an API endpoint?
    - An API endpoint is the point of communication between the resource and your application. Endpoints are

(Flask) routes. The endpoint is appended to the base URL of the API.

- What are the core tenets of REST in the context of APIs?
  - A set of standards for API architecture
  - URLs are uniform and meaningful
  - Client and server are independent of each other
  - Can be in different languages!
  - HTTP(S) is the protocol
  - Each request is independent