Web Development Pick-Me-Ups

# HTML

## Basic

* **HTML** stands for **H**yper**T**ext **M**arkup **L**anguage and is used to create the structure and content of a webpage.
* Most HTML elements contain opening and closing tags with raw text or other HTML tags between them.
* HTML elements can be nested inside other elements. The enclosed element is the child of the enclosing parent element.
* Any visible content should be placed within the opening and closing <body> tags.
* Headings and sub-headings, <h1> to <h6> tags, are used to enlarge text.
* <p>, <span> and <div> tags specify text or blocks.
* The <em> and <strong> tags are used to emphasize text.
* Line breaks are created with the <br> tag.
* Ordered lists (<ol>) are numbered and unordered lists (<ul>) are bulleted.
* Images (<img>) and videos (<video>) can be added by linking to an existing source.
* Syntax: <img src="smiley.gif" alt="Smiley face" height="42" width="42">
* “alt” is used to specify stuff when the image doesn’t load or give a description of the image
* Metadata is information about the page that isn’t displayed directly on the web page. Unlike the information inside of the <body> tag, the metadata in the head is information about the page itself.
* The <!DOCTYPE html> declaration should always be the first line of code in your HTML files. This lets the browser know what version of HTML to expect.
* The <html> element will contain all of your HTML code.
* Information about the web page, like the title, belongs within the <head> of the page.
* You can add a title to your web page by using the <title> element, inside of the head.
* A webpage’s title appears in a browser’s tab.
* Anchor tags (<a>) are used to link to internal pages, external pages or content on the same page.
* You can create sections on a webpage and jump to them using <a> tags and adding ids to the elements you wish to jump to.
* Whitespace between HTML elements helps make code easier to read while not changing how elements appear in the browser.
* Indentation also helps make code easier to read. It makes parent-child relationships visible.
* Comments are written in HTML using the following syntax: <!-- comment -->.

## Tables

* The <table> element creates a table.
* The <tr> element adds rows to a table.
* To add data to a row, you can use the <td> element.
* Table headings clarify the meaning of data. Headings are added with the <th> element.
* Table data can span columns using the colspan attribute.
* Table data can span rows using the rowspan attribute.
* Tables can be split into three main sections: a head, a body, and a footer.
* A table’s head is created with the <thead> element.
* A table’s body is created with the <tbody> element.
* A table’s footer is created with the <tfoot> element.

## Forms

* The purpose of a <form> is to allow users to input information and send it.
* The <form>‘s action attribute determines where the form’s information goes.
* The <form>‘s method attribute determines how the information is sent and processed.
* To add fields for users to input information we use the <input> element and set the type attribute to a field of our choosing:
  + Setting type to "text" creates a single row field for text input.
  + Setting type to "password" creates a single row field that censors text input.
  + Setting type to "number" creates a single row field for number input.
  + Setting type to "range" creates a slider to select from a range of numbers.
  + Setting type to "checkbox" creates a single checkbox which can be paired with other checkboxes.
  + Setting type to "radio" creates a radio button that can be paired with other radio buttons.
  + Setting type to "list" will pair the <input> with a <datalist> element if the id of both are the same.
  + Setting type to "submit" creates a submit button.
  + The “placeholder” attribute is displayed in the field while there is no input and disappears after the user interacts. It can be used instead of the label
* A <select> element is populated with <option> elements and renders a dropdown list selection.
* A <datalist> element is populated with <option> elements and works with an <input> to search through choices.
* A <textarea> element is a text input field that has a customizable area.
* When a <form> is submitted, the name of the fields that accept input and the value of those fields are sent as name=value pairs.

### Validation

* Client-side validations happen in the browser before information is sent to a server.
* Adding the required attribute to an input related element will validate that the input field has information in it.
* Assigning a value to the min attribute of a number input element will validate an acceptable minimum value.
* Assigning a value to the max attribute of a number input element will validate an acceptable maximum value.
* Assigning a value to the minlength attribute of a text input element will validate an acceptable minimum number of characters.
* Assigning a value to the maxlength attribute of a text input element will validate an acceptable maximum number of characters.
* Assigning a regex to pattern matches the input to the provided regex.
* If validations on a <form> do not pass, the user gets a message explaining why and the <form> cannot be submitted.

## Semantic HTML

* Semantic HTML introduces meaning to a page through specific elements that provide context as to what is in between the tags.
* Semantic HTML is a modern standard and makes a website accessible for people who use screen readers to translate the webpage and improves your website’s SEO.
* <header>, <nav> , <main> and <footer> create the basic structure of the webpage.
* <section> defines elements in a document, such as chapters, headings, or any other area of the document with the same theme.
* <article> holds content that makes sense on its own such as articles, blogs, comments, etc.
* <aside> contains information that is related to the main content, but not required in order to understand the dominant information.
* <figure> encapsulates all types of media.
* <figcaption> is used to describe the media in <figure>.
* <video>, <embed>, and <audio> elements are used for media files.

# CSS

CSS, or **Cascading Style Sheets**, is a language that web developers use to style the HTML content on a web page. If you’re interested in modifying colors, font types, font sizes, shadows, images, element positioning, and more, CSS is the tool for the job!

* CSS can change the look of HTML elements. In order to do this, CSS must select HTML elements, then apply styles to them.
* CSS can select HTML elements by tag, class, or ID.
* Multiple CSS classes can be applied to one HTML element.
* Classes can be reusable, while IDs can only be used once.
* IDs are more specific than classes, and classes are more specific than tags. That means IDs will override any styles from a class, and classes will override any styles from a tag selector.
* Multiple selectors can be chained together to select an element. This raises the specificity, but can be necessary.
* Nested elements can be selected by separating selectors with a space.
* The !important flag will override any style, however it should almost never be used, as it is extremely difficult to override.
* Multiple unrelated selectors can receive the same styles by separating the selector names with commas.

## Rule-Sets

* CSS declarations are structured into property and value pairs.
* The font-family property defines the typeface of an element.
* font-size controls the size of text displayed.
* font-weight defines how thin or thick text is displayed.
* The text-align property places text in the left, right, or center of its parent container.
* Text can have two different color attributes: color and background-color. color defines the color of the text, while background-color defines the color behind the text.
* CSS can make an element transparent with the opacity property.
* CSS can also set the background of an element to an image with the background-image property.