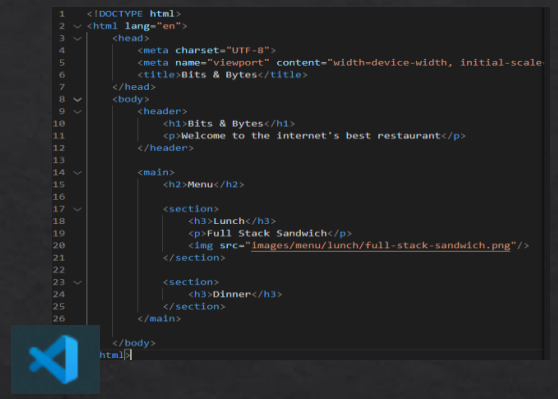
Module 3
Introduction - HTML CSS



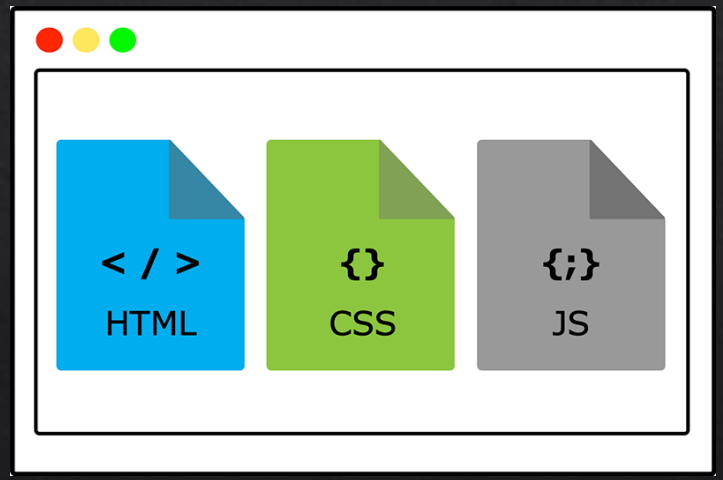
Session Objectives:

- Understand the basics of HTML
    - HTML document structure
    - HTML tags & attributes
- Understand the following basic HTML tags:
    - Headings (h1, h2, etc)
    - Paragraphs
    - Links
    - Images
- Utilize the different elements that go into building a form
- Understand the action attribute of the form tag
- Describe what Semantic HTML is
- Demonstrate a basic understanding of what CSS is and how to create style declarations
- Understand how fonts work in the browser and some of the different properties available for styling them
- Understand how colors work and some of the different properties for working with them

Module 3
Introduction - HTML CSS

HTML, **CSS**, and **JavaScript** are the building blocks of the web. Each of these languages play their own significant roles in building the web:

- **HTML**: Provides the basic structure or markup of a document.
- **CSS**: Provides formatting of the document to control presentation and layout.
- **JavaScript**: Provides behavior to the document.



Module 3
Introduction - HTML CSS

Introduction HTML

HTML stands for **H**yper **T**ext **M**arkup **L**anguage. HTML is a markup language, which means that you take the raw content you want to display and structure it using HTML tags. A web browser needs to be told how to display and structure your content.

Some of the advantages of doing web development with HTML and CSS are that you can start working with the languages pretty quickly, and they have a fast feedback loop.

Module 3
Introduction - HTML CSS

HTML Tags

If you look in the code block, you'll notice that almost every HTML tag has a start and end tag. You saw this earlier with the <head> tag. There was an opening tag, <head>, and a closing tag, </head>. In between, there was a <title> tag. and inside the <title> tag was the text that set the page title:

```html
<head>
  <!-- metadata goes here -->
  <title>My Page Title</title>
</head>
```

There are also tags that don't have any content between them. If that's the case, you don't need a closing tag. These types of tags are self-closing tags:

```html
<!-- line break -->
<br />

<!-- horizontal rule -->
<hr />
```

Module 3
Introduction - HTML CSS

HTML Attributes

Elements in HTML have **attributes**. These are additional values that configure the elements or adjust their behavior in various ways to meet the criteria the users want. There are two types of attributes: global attributes, which apply to all HTML elements, and then attributes that apply to one or more elements.

Attributes are defined on HTML tags using the attribute name and value. You already saw an example of an attribute in the <html> element:

```html
<html lang="en">
  <meta charset="UTF-8" />
</html>
```

Module 3
Introduction - HTML CSS

Forms

HTML forms are one of the most important parts of a website. They allow you to collect information from your visitors. This could be something as simple as a contact form, mailing list signup form, or blog comment, all the way to an entire e-commerce checkout process.

All forms start with the same structure and use different elements to capture different types of data from your visitors.

The form tag

The HTML <form> element represents a document section containing interactive controls for submitting information:

```html
<form>
  <!-- form interactive controls -->
</form>
```

Module 3
Introduction - HTML CSS

Forms

The form element has two attributes:

- **action**: The URL that processes the form submission.
- **method**: The HTTP method to submit the form with. Possible values:
  - post: The POST method; form data sent as the request body.
  - get: The GET method; form data appended to the action URL with a ? separator.

Imagine that you had a login form that asked the user for their username and password. When they submit the form, their information is sent to your server using the POST method. An example login form might look like this:

```html
<form action="http://localhost:3000/api/users/login" method="POST">
  <!-- elements to capture username and password here -->
</form>
```

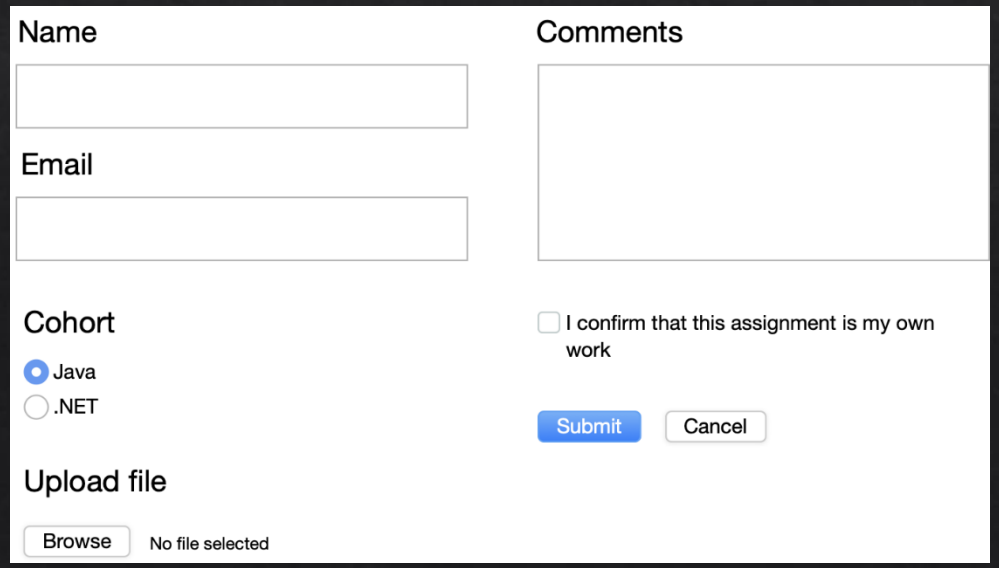
Module 3
Introduction - HTML CSS

Forms

Form Elements

When building a form, it's important to understand what tools are available to you. There are many elements and variations of those elements that let you add interactive controls to your forms. In this section, you'll learn about the different form elements, when you might want to use them, and see some example code.

This example form has some of the interactive controls you'll learn about below.



Module 3
Introduction - HTML CSS

Forms

Form Elements - Input

The <input> element is one of the most powerful elements in HTML because of the sheer number of combinations of input types and attributes. When you define an input element, you start by setting the **name** attribute:

```html
<input name="firstName" />
```

When a form is submitted, the input element is sent in the form of a key/value pair. The name of the element is what you define in the name attribute, and the value is what the user typed in the input control. You need to define the names of your elements so that you can handle them when processing them on the server.

Module 3
Introduction - HTML CSS

Forms

Form Elements - Input

The next attribute you need to set is the **type**. How an <input> element works varies considerably depending on the value of its type attribute. If this attribute isn't specified, the default type adopted is text:

```html
<input name="firstName" type="text" />
```

Module 3
Introduction - HTML CSS

Forms

Form Elements - Input

Input types
The following is a list of common types and what they provide:

Type	Description
button	A push-button with no default behavior displaying the value of the value attribute, empty by default.
text	The default value. A single-line text field. Line-breaks are automatically removed from the input value.
checkbox	A checkbox allowing single values to be selected/deselected.
radio	A radio button, allowing a single value to be selected out of multiple choices with the same name value.
date	A control for entering a date (year, month, and day, with no time). Opens a date picker or numeric wheels for year, month, day when active in supporting browsers.
email	A field for editing an email address. Looks like a text input, but has validation parameters and relevant keyboard in supporting browsers and devices with dynamic keyboards.
color	A control for specifying a color; opening a color picker when active in supporting browsers.
file	A control that lets the user select a file. Uses the accept attribute to define the types of files that the control can select.
number	A control for entering a number. Displays a spinner and adds default validation when supported. Displays a numeric keypad in some devices with dynamic keypads.
submit	A button that submits the form.

Module 3
Introduction - HTML CSS

Forms

Form Elements - Input

Input attributes
Because <input> has so many types, there is a long list of supported attributes based on type. Here are a few of the more common attributes you'll come across when working on input elements.

Attribute	Description
name	Name of the input form control. Submitted with the form as part of a name/value pair.
type	Type of input form control.
value	Current value of the form control. Submitted with the form as part of a name/value pair.
disabled	Whether the form control is disabled.
minlength	Minimum length (number of characters) of value.
maxlength	Maximum length (number of characters) of value placeholder.
required	Boolean. A value is required or must be checked for the form to be submittable.

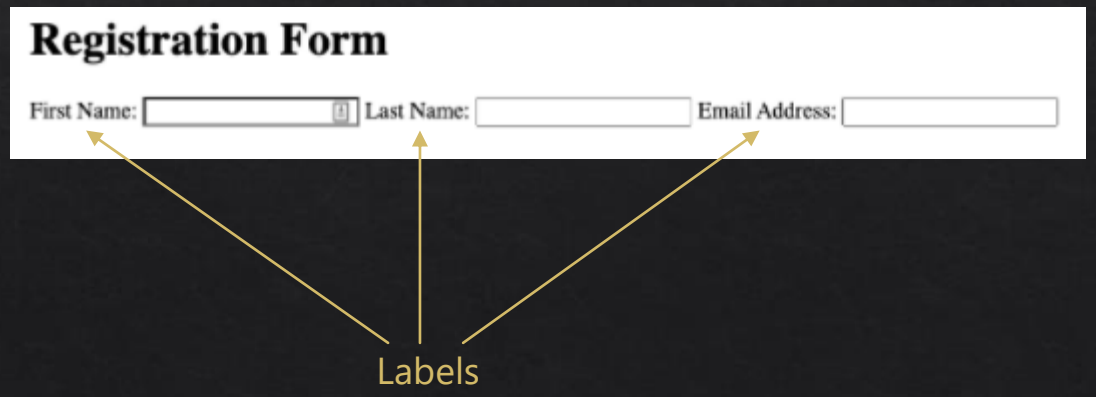
Module 3
Introduction - HTML CSS

Forms

Form Elements - Label

The HTML <label> element represents a caption for an item in a user interface. For example, you need to collect the first name, last name, and email address in a user registration form.

You start with your form element, create input controls, and give each of them unique names so you know which piece of data is which when the form is submitted to your server:



Module 3
Introduction - HTML CSS

Forms

Form Elements - Textarea

You can use the input control to ask the user for some text, but what happens when you need paragraphs of text? The HTML <textarea> element represents a multi-line, plain-text editing control.

It's useful when you want to allow users to enter a sizeable amount of free-form text—for example, a comment on a review or feedback form. In this example, it's used to allow users to enter information about themselves:

```
<label for="about">Tell us about yourself:</label><br />
<textarea id="about" name="about" rows="8" cols="60"></textarea>
```

The rows and cols attributes are one way to specify the exact size of the textarea. Setting these is a good idea for consistency, as browser defaults can differ.

Module 3
Introduction - HTML CSS

Forms

Form Elements - Select

The HTML <select> element represents a control that provides a menu of options:

```html
<form>
  <label for="department">Department:</label>
  <select id="department" name="department">
    <option value="">--Please choose an option--</option>
    <option value="sales">Sales</option>
    <option value="marketing">Marketing</option>
    <option value="IT">IT</option>
  </select>
</form>
```

Each menu option is defined by an <option> element nested inside the <select> element. Each <option> element can have a value attribute containing the data value to submit to the server when that option is selected.

If no value attribute is included, the value defaults to the text contained inside the element. You can include a selected attribute on an <option> element to make it selected by default when the page loads for the first time.

By default, the <select> element only allows for a single choice. If you want to allow for multiple selections, you can use the multiple attribute.

Module 3
Introduction - HTML CSS

Semantic HTML

The word semantic means, "of or relating to meaning in language." If you're using semantic HTML, you're using elements that have meaning to them.

Sematic HTML is a key component to a much larger and ambitious project, the Semantic Web. Originally proposed by Sir Tim Berners-Lee, the creator of the World Wide Web, the Semantic Web allows for more intelligent searches and knitting together of knowledge than is possible with simple word matches. This is because semantic HTML tags provide additional context or meaning to the words beyond the fact they match.

Another advantage of using semantic HTML is that your website becomes more accessible. Accessibility is the practice of making your websites usable by as many people as possible.

Many people think of accessibility when it comes to the visually impaired who visit your website with a screen reader. Making websites more accessible also includes those using mobile devices or those with slower internet connections.

There are over a dozen Semantic HTML elements. The following are the most commonly used:

- `<header>`
- `<footer>`
- `<nav>`
- `<article>`
- `<section>`
- `<aside>`
- `<main>`

Module 3
Introduction - HTML CSS

Introduction to CSS

As you saw in the previous section, HTML on its own isn't very impressive. HTML allows you to structure your documents, but Cascading Style Sheets (CSS) brings your web pages to life with layout and style. CSS is a style sheet language used for describing the presentation of a document written in a markup language like HTML.

Cascading Style Sheets (CSS) brings your web pages to life!

Module 3
Introduction - HTML CSS

New Tools!