Introduction To Website Development Technologies

HELLO!

I am Trevoir Williams

Software Engineer | Lecturer

Course Objectives

What is HTML

- Publish online documents with headings, text, tables, lists, photos, etc.
- Retrieve online information via hypertext links, at the click of a button.

What is CSS

- CSS is the language for describing the presentation of Web pages, including colors, layout, and fonts.
- CSS makes it easier to maintain sites, share style sheets across pages, and tailor pages to different environments.
- Custom frameworks like Bootstrap, Material, etc.

Course Objectives

What is JavaScript

- JavaScript is the world's most popular programming language.
- JavaScript is the programming language of the Web.
- JavaScript is easy to learn.

What is Bootstrap

- Bootstrap, which is the most popular HTML, CSS, and JavaScript framework for creating responsive, mobile-first websites.

Visual Studio Code

What

A Microsoft open-source text editor that is chock full of plugins and tools to make you as productive as possible when building websites.

Why

- Powerful Web Tools to allow development with ASP.NET, jQuery, Bootstrap, among other popular frameworks.
- Git Integration - Manage your source code in Git repos hosted by any provider, including GitHub.

Other Techhnologies

GitHub

- Source or Version Control is a system that records changes to files over time so that you can recall specific versions later

Netlify

An all-in-one platform for automating modern web projects.

1

Why Learn Web Development?

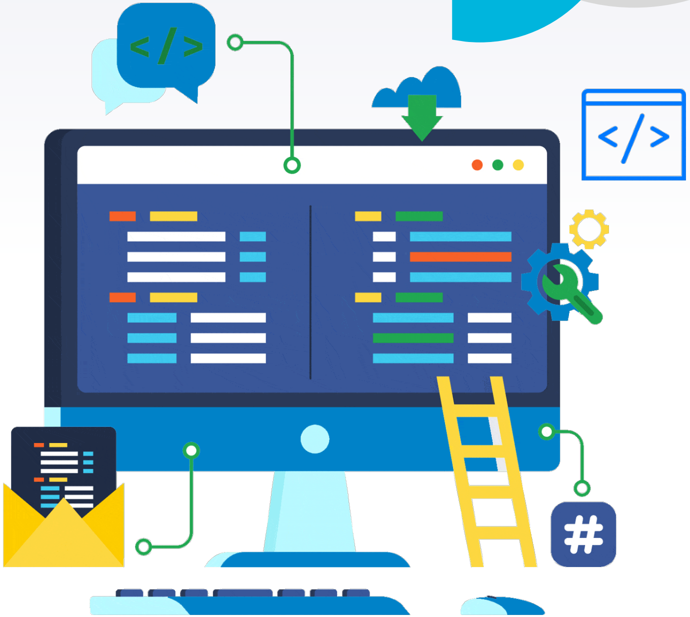
- Involves the creation and maintenance of websites
- Allows you to actively contribute to content on the internet
- Play a crucial role in shaping interactions with digital content

What Are Web Technologies

- Tools that facilitate communication and interaction
- Browsers, HTML, CSS, JavaScript, etc.
- Facilitates global reach

Understand Frontend Technologies

- HTML, CSS, and JavaScript are fundamental languages
- HTML is for the content
- CSS is to style the content
- JavaScript is to add interactivity



- Several job opportunities exist for this skillset
- Easy to use for both personal and professional purposes

Web Development Tools

- Any text editor is valid
  - Notepad, Notepad++, even Microsoft Word
- Files need specific extensions:
  - .html, .css, .js
- An Integrated Development Environment (IDE) is best.
  - Brackets.io, Atom Editor, Sublime, Visual Studio Code

Visual Studio Code

- Lightweight
- Cross-Platform
- Extensible
- Git integration
- Well maintained by Microsoft and the developer community
- Browser based option

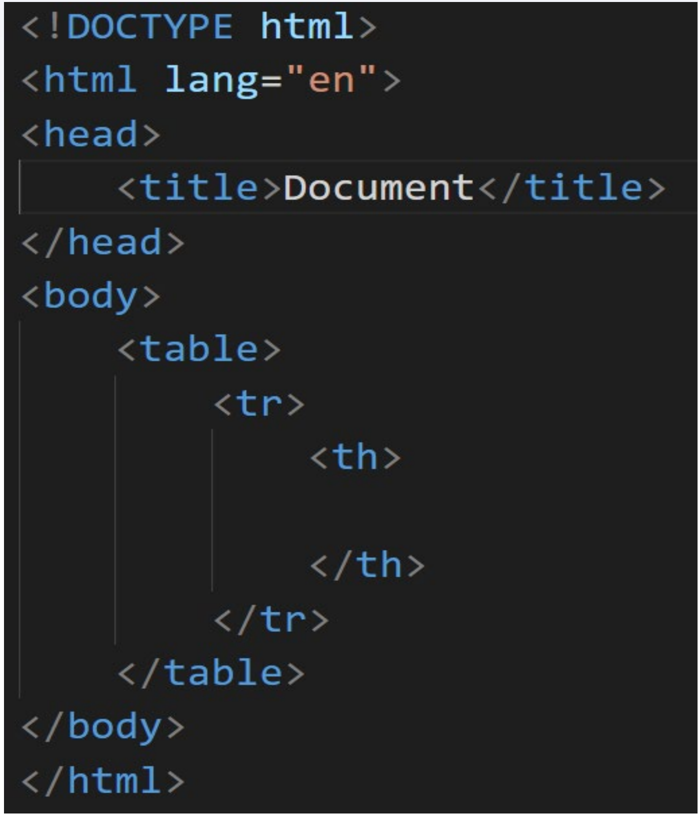
1 Explore HTML

What is HTML

- Short for HyperText Markup Language
- Standard language for authoring web pages
- Defines the structure of web content
  - Images, text, tables, forms, etc.
- Lightweight and easy to learn
- Often assisted by CSS and JavaScript

Basic HTML Syntax

- Uses tags that open and close
  - E.g. <body>…</body>
- Tags add elements to the web page
  - E.g. <table>…</table> adds a table structure



```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Document</title>
</head>
<body>
    <table>
        <tr>
            <th>

            </th>
        </tr>
    </table>
</body>
</html>
```

1

Cascading Style Sheets (CSS)

Cascading Style Sheets

- Used to style and format HTML documents
- Separates presentation from content
- Enhances the visual appearance of web pages

Why use CSS

- Maintainability: Easily change the look of entire website from one file
- Consistency: Apply consistent styles to multiple pages
- Flexibility: Control layout and presentation across different devices
- Accessibility: Improve user experience for people with disabilities

Basic CSS Syntax

-
  ```
  selector {
          property: value;
  }
  ```
- Selector: Targets HTML element to style (e.g., h1, .class-name, #id)
- Property: The CSS property to modify (e.g., color, font-size, margin)
- Value: The value assigned to the property (e.g., blue, 16px, 10px 20px)

1

Next Up
Add CSS to Website

Inline CSS

- style= “property: value; property: value; etc;”
- Easy to use in any tag
- Hard to maintain across several tags
- Has to be repeated everywhere

Internal Style Sheet

-
  ```
  <style>
   selector {
          property: value;
   }
  etc.
  </style>
  ```
- Used at the top of each web page.
- Global to the page, limited to each page
- Hard to maintain

External Style Sheet

- CSS is defined in one file and referenced by each page
- <link rel="stylesheet" href="pathToStylesheet.css" />
- Link is placed in head section of a page
- Global to the website
- Easiest to maintain. Honors DRY principles

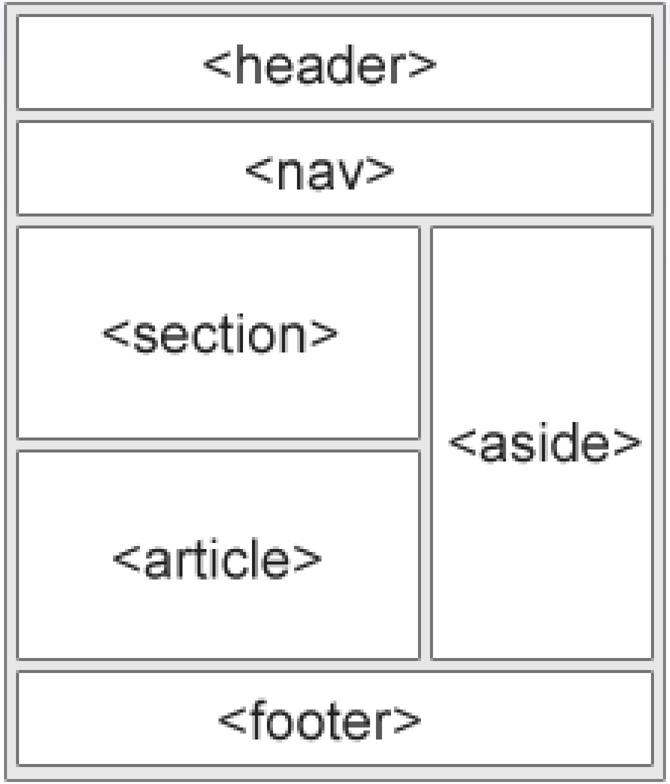
HTML Layout and Semantics

- Semantic elements have meanings
- Non-semantics tags like <div> or <span>
- Semantic elements like <form> <section> <article>
- Traditional layout contains HTML code like:

  ```
  <div id="nav">
  <div class="header">
  <div id="footer">
  ```

  to indicate page sections.
- There are some semantic elements that can be used to define different parts of a web page:

Website Layout



<header>

<nav>

<section>

<aside>

<article>

<footer>

JavaScript

Introduction

- JavaScript is a versatile and powerful programming language that runs on web browsers, enabling dynamic and interactive website experiences.
- Developed in the mid-1990s, JavaScript has become one of web development's cornerstones.

JavaScript empowers developers to:

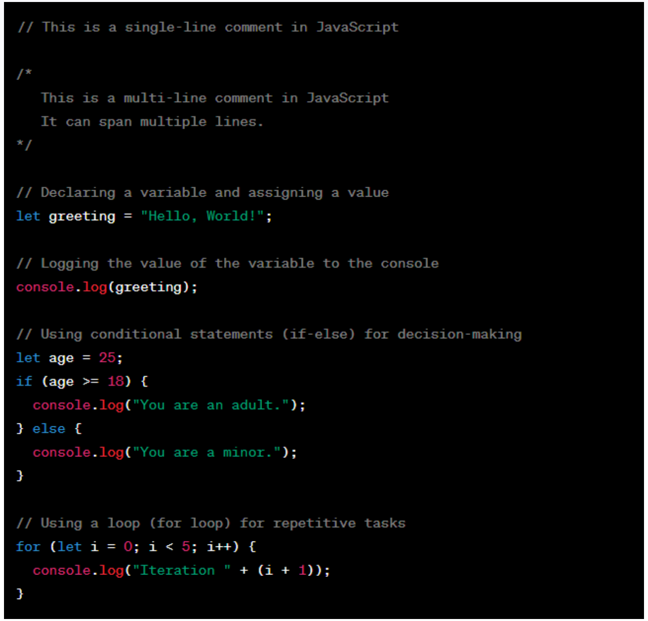
- Add interactivity
- Manipulate content
- Respond to user actions in real-time

Why JavaScript

- Client-Side Interactivity
- Integration with HTML and CSS
- Continuous Improvement
- Wide Adoption
- Versatility
- Asynchronous Programming
- Browser Support
- Open Source Community
- Easy to Learn

Basic JavaScript Syntax

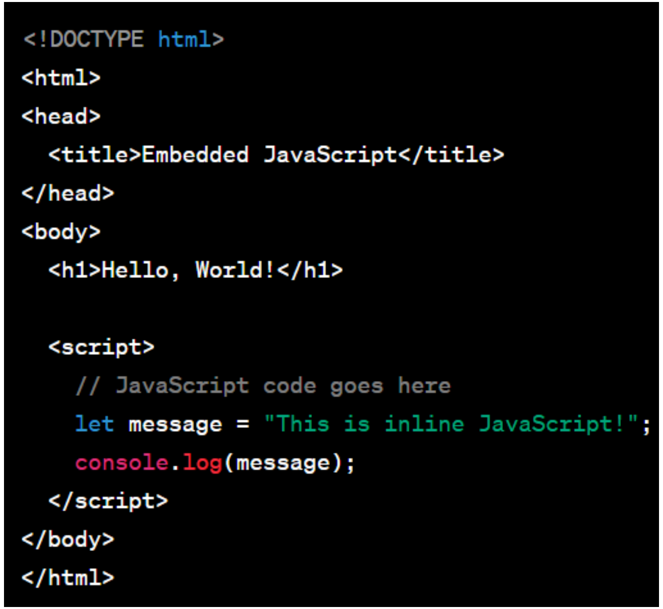
- JavaScript typically consists of statements and expressions.
- JavaScript code can be embedded directly into HTML files using <script> tags or included in separate .js files and linked to HTML pages.



```javascript
// This is a single-line comment in JavaScript

/*
   This is a multi-line comment in JavaScript
   It can span multiple lines.
*/

// Declaring a variable and assigning a value
let greeting = "Hello, World!";

// Logging the value of the variable to the console
console.log(greeting);

// Using conditional statements (if-else) for decision-making
let age = 25;
if (age >= 18) {
  console.log("You are an adult.");
} else {
  console.log("You are a minor.");
}

// Using a loop (for loop) for repetitive tasks
for (let i = 0; i < 5; i++) {
  console.log("Iteration " + (i + 1));
}
```

Embedding JavaScript into HTML

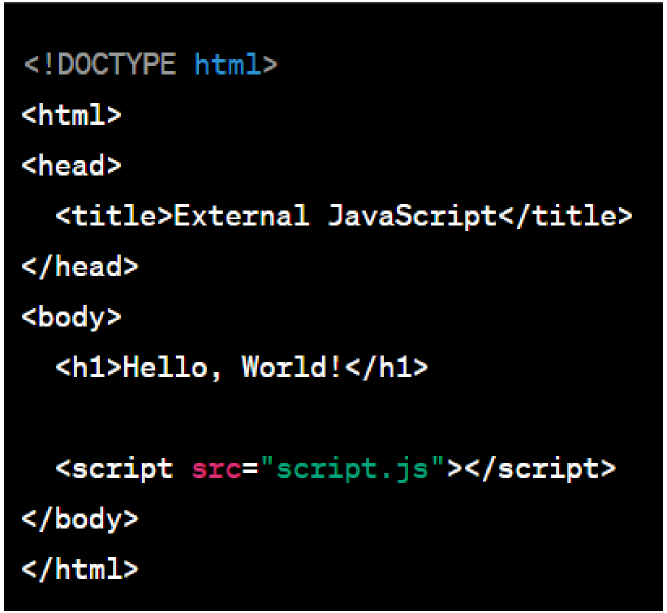
Inline JavaScript



```
<!DOCTYPE html>
<html>
<head>
  <title>Embedded JavaScript</title>
</head>
<body>
  <h1>Hello, World!</h1>

  <script>
    // JavaScript code goes here
    let message = "This is inline JavaScript!";
    console.log(message);
  </script>
</body>
</html>
```

Embed JavaScript code directly into the HTML file using the <script> tag

External JavaScript File

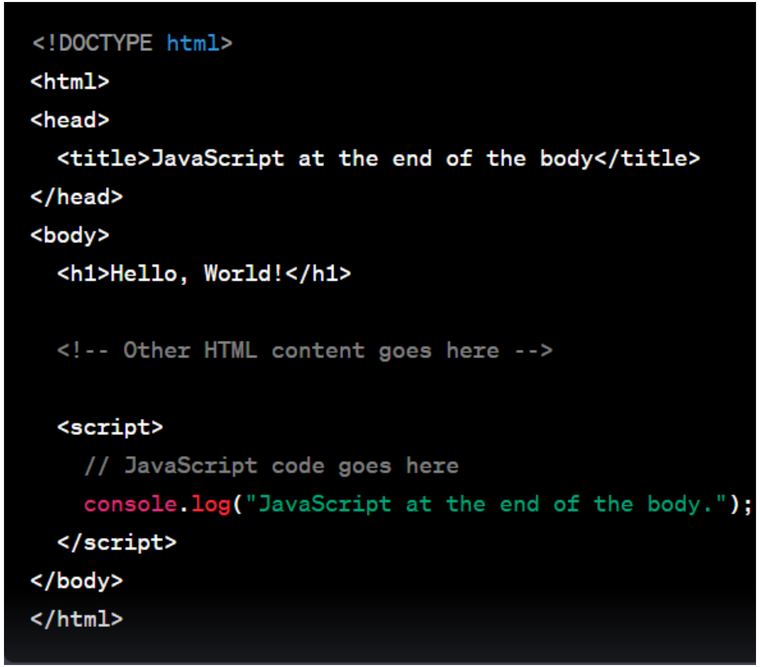


```
<!DOCTYPE html>
<html>
<head>
  <title>External JavaScript</title>
</head>
<body>
  <h1>Hello, World!</h1>

  <script src="script.js"></script>
</body>
</html>
```

Reference an eternal **.js** file in the **src** attribute of the <script> tag.

Embedding JavaScript into HTML

Place script tag at the end of the body



```
<!DOCTYPE html>
<html>
<head>
  <title>JavaScript at the end of the body</title>
</head>
<body>
  <h1>Hello, World!</h1>

  <!-- Other HTML content goes here -->

  <script>
    // JavaScript code goes here
    console.log("JavaScript at the end of the body.");
  </script>
</body>
</html>
```

For better performance and to ensure that the JavaScript code executes after the HTML content is loaded by ., placing the <script> tag at the end of the <body> element.

Introduction to jQuery

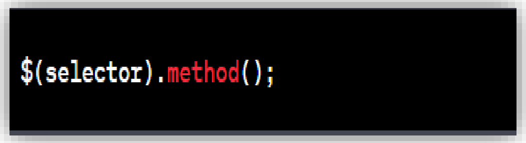
jQuery is a JavaScript library that offers a great deal of flexibility and power to web developers.

Some advantages of jQuery are:

- Easy to use and understand, even for novice designers
- It can standardize and simplify interactions between HTML elements and JavaScript Code
- Provides a wide range of plugins available for various needs
- Create dynamic and attractive effects that are like flash, but more SEO- friendly
- Lightweight and works in all browsers

Basic jQuery Syntax

jQuery syntax follows a pattern of selecting HTML elements and performing actions on them. It typically involves a dollar sign $ followed by a set of parentheses ().



```
$(selector).method();
```

Inside the parentheses, you pass a CSS-style selector to target specific elements, and then you chain one or more jQuery methods to perform actions on those elements

Break down the basic jQuery syntax components

- **$ (Dollar Sign):** The dollar sign is an alias for the jQuery function, which is the entry point to use jQuery. It allows you to access and manipulate the selected HTML elements.

- **selector**: The selector is a CSS-style expression that identifies the HTML elements you want to target. It could be a tag name, class name, ID, or any other valid CSS selector.

- **method**: The method is a jQuery function that performs an action on the selected elements. It can be a method to manipulate the elements, handle events, make AJAX requests, or perform animations, among other things.

Embedding jQuery into HTML

Embedding jQuery into an HTML document, requires the jQuery library using a script tag

Download the jQuery library and host it locally or use a content delivery network (CDN) to access the jQuery library online.

Download and Host jQuery Locally

Place the downloaded jQuery file (usually named jquery-x.y.z.min.js, where x.y.z represents the version number) in your project directory. Link the jQuery file in your HTML file using the <script> tag just before the closing </body> tag.

Use jQuery via CDN (Content Delivery Network):

Instead of downloading the jQuery file, you can use a publicly available CDN to load jQuery into your HTML file. This can reduce the load on your server and benefit from potential caching by widely-used CDNs.

Bootstrap

Introduction

- Bootstrap is a popular open-source front-end framework used for building responsive and visually appealing websites and web applications

Key features of Bootstrap:

- Responsive Design
- CSS Grid System
- Pre-styled Components
- Customizable Themes
- JavaScript Plugins
- Browser Compatibility
- Extensive Documentation

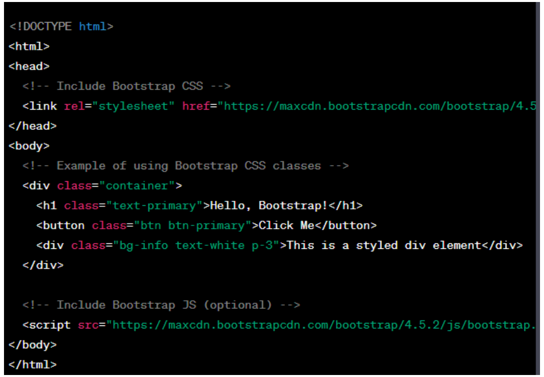
Why BootStrap

- Rapid Development
- Responsive Design
- Consistency and Uniformity
- Cross-Browser Compatibility
- Customization Options
- Community and Support
- Integration with JavaScript Libraries
- Continued Development and Improvement

Basic Bootstrap Syntax

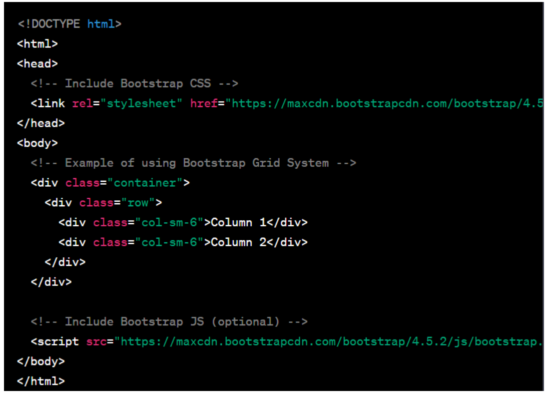
Bootstrap is based on HTML, CSS, and JavaScript, so its syntax involves using predefined CSS classes and JavaScript components provided by the framework. Here are some basic Bootstrap syntax examples

Using Bootstrap CSS Classes



```
<!DOCTYPE html>
<html>
<head>
  <!-- Include Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5
</head>
<body>
  <!-- Example of using Bootstrap CSS classes -->
  <div class="container">
    <h1 class="text-primary">Hello, Bootstrap!</h1>
    <button class="btn btn-primary">Click Me</button>
    <div class="bg-info text-white p-3">This is a styled div element</div>
  </div>

  <!-- Include Bootstrap JS (optional) -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.
</body>
</html>
```

Using Bootstrap grid system

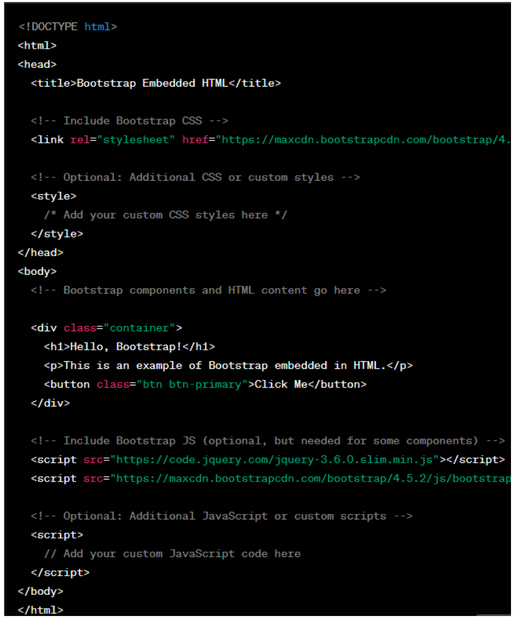


```
<!DOCTYPE html>
<html>
<head>
  <!-- Include Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5
</head>
<body>
  <!-- Example of using Bootstrap Grid System -->
  <div class="container">
    <div class="row">
      <div class="col-sm-6">Column 1</div>
      <div class="col-sm-6">Column 2</div>
    </div>
  </div>

  <!-- Include Bootstrap JS (optional) -->
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.
</body>
</html>
```

Embedding Bootstrap into HTML

Embedding Bootstrap into an HTML document, requires the necessary Bootstrap CSS and JavaScript files

Download the Bootstrap files and host them locally or use content delivery network (CDN) links to access the Bootstrap files online.



```
<!DOCTYPE html>
<html>
<head>
  <title>Bootstrap Embedded HTML</title>

  <!-- Include Bootstrap CSS -->
  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/4.

  <!-- Optional: Additional CSS or custom styles -->
  <style>
    /* Add your custom CSS styles here */
  </style>
</head>
<body>
  <!-- Bootstrap components and HTML content go here -->

  <div class="container">
    <h1>Hello, Bootstrap!</h1>
    <p>This is an example of Bootstrap embedded in HTML.</p>
    <button class="btn btn-primary">Click Me</button>
  </div>

  <!-- Include Bootstrap JS (optional, but needed for some components) -->
  <script src="https://code.jquery.com/jquery-3.6.0.slim.min.js"></script>
  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap

  <!-- Optional: Additional JavaScript or custom scripts -->
  <script>
    // Add your custom JavaScript code here
  </script>
</body>
</html>
```

Here's an example of how to embed Bootstrap in an HTML document

Bootstrap Alternatives

- 1. Tailwind CSS (Free)
- 2. Bulma(Free)
- 3. Materialize (Paid)
- 4. Foundation by Zurb (Paid)
- 5. Pure CSS (Free)
- 6. Element(Paid)
- 7. Skeleton(Free)
- 8. Metro UI(Freemium)

Web Hosting

Introduction

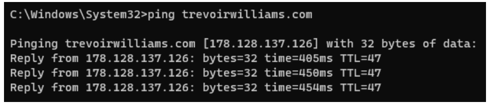
- Hosting refers to making a website, application, or any digital content accessible to users over the Internet.

- When you host a website or an application, you store the necessary files, data, and resources on a connected server.

Let’s review some concepts

Domain Name System (DNS) Resolution

the process of converting human-readable domain names, like "www.example.com," into numerical IP addresses that computers use to locate and communicate with each other on the internet.



```
C:\Windows\System32>ping trevoirwilliams.com

Pinging trevoirwilliams.com [178.128.137.126] with 32 bytes of data:
Reply from 178.128.137.126: bytes=32 time=405ms TTL=47
Reply from 178.128.137.126: bytes=32 time=450ms TTL=47
Reply from 178.128.137.126: bytes=32 time=454ms TTL=47
```

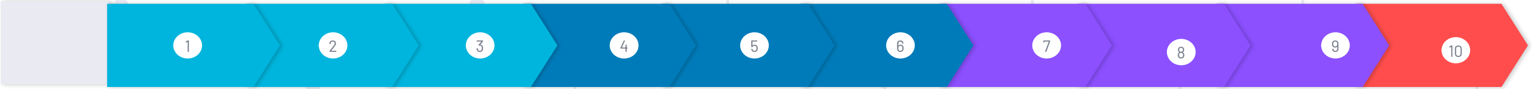
Web Server

It processes incoming requests from clients (usually web browsers) and sends back the requested web pages, files, or resources to be displayed on the client's device.

SSL (Secure Sockets Layer)

SSL (Secure Sockets Layer) encryption, and its successor TLS (Transport Layer Security), are cryptographic protocols designed to secure the communication between a client (such as a web browser) and a server (such as a web server) over the internet.

How does Web hosting work



Domain Name System (DNS) Resolution
Request to a DNS server to translate the domain name into an IP address.

Request to the Web Server
Once the browser obtains the IP address, it sends a request to the web server associated with that IP address

Server Processing
The web server receives the request and processes it

File Retrieval and Rendering
The web server retrieves the requested files from its storage and sends them back to the user's browser over the internet

Content Delivery
To ensure efficient and fast content delivery, many web hosting providers use Content Delivery Networks (CDNs)

Interactivity and Dynamic Content
If the website has interactive elements or requires dynamic the web server processes the user's interactions or retrieves the required data from databases or other sources.

Response to User Interaction
As users interact with the website, their actions generate requests to the web server

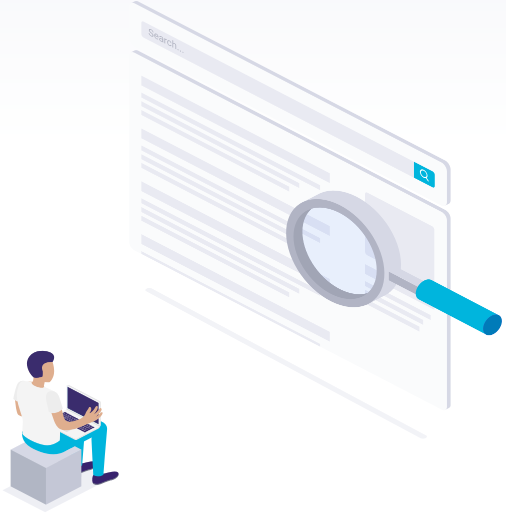
Database Integration
If the website includes a backend database (the web server communicates with the database to retrieve, update, or store data as needed

SSL Encryption (If Enabled)
Secure Socket Layer (SSL) or Transport Layer Security (TLS) encryption may be used to secure data transmission between the user's browser and the web server

Response and Rendering
The browser receives the requested resources from the server and renders the webpage for the user to see and interact with.

Web Hosting options

- Shared Hosting
- Dedicated Hosting
- Virtual Private Server (VPS) Hosting
- Cloud Hosting



Web Hosting Providers

- GoDaddy
- BlueHost
- HostGator
- Netlify
- Microsoft Azure
- Amazon Web Services (AWS)
- Many others...

Introduction

➢Netlify is a cloud computing platform that offers a variety of services and tools designed to simplify the process of building, deploying, and managing modern web applications and websites. It's particularly focused on providing a seamless and efficient workflow for web developers.

Key features of Netlify are:

➢Static Site Hosting
➢Continuous Deployment
➢Build Automation
➢Serverless
➢Global Scaling

Why Netlify

- Easy Deployment Workflow
- Static Site Hosting
- Continuous Deployment
- Serverless Functions
- Custom Domains and SSL
- Developer-Friendly
- Branch Deployments
- Scalability
- Analytics and Insights

1

Course Recap

What did we learn?

Visual Studio Code

- Coding environment
- Keyboard shortcuts
- Source Control Integration

HTML

- HTML Syntax
- Best Practices
- How to author web pages

What did we learn?

CSS

- Inline styling, Internal and External Stylesheets
- Best Practices
- Bootstrap framework

JavaScript + jQuery

- JavaScript Syntax
- How to embed JavaScript in HTML
- How to use jQuery

What did we learn?

GitHub

- Git and commits
- Manage code versions

Netlify

- Manual Deployment
- CI/CD Pipeline
- Custom Domains

Next Steps

- Practice!!
- Do personal projects
- Take on small professional projects
- Explore more frameworks (Bootstrap and alternatives)
- Explore more JavaScript
- Check out additional courses
  - Database Development
  - ASP.NET Core Development
  - Azure DevOps
- Practice even more!!!!!
  - Practice Makes Permanent

See You Soon

Trevoir Williams

Software Engineer | Lecturer | Author