Static Websites

- Static websites are built using only HTML, CSS, and sometimes simple JavaScript. They cannot handle user input or dynamically update content. Static websites are fast to load but lack interactivity, meaning users can't submit information (like forms) or see personalized data.

1. HTML

- HyperText Markup Language (HTML) is the basic building block of web development. It structures the content of the webpage using elements like headings, paragraphs, images, and links. HTML defines the skeleton of the website.

2. CSS

- Cascading Style Sheets (CSS) is used to style the HTML elements. It controls the layout, colors, fonts, and overall design of a website, making it visually appealing.

3. Bootstrap

- Bootstrap is a popular CSS framework that helps build responsive websites faster. It provides pre-designed components like buttons, forms, navigation bars, and grid systems to ensure your site looks great on all devices (mobile, tablet, desktop).

4. Git

- Git is a version control system that tracks changes in your code. It allows multiple developers to work on a project simultaneously and ensures changes are safely stored. Git is especially useful for collaborating on projects and keeping a history of code changes.

Dynamic websites are web pages that can display different content and interact with users in real-time. Unlike static websites, which have fixed content, dynamic websites can change their content based on user input, preferences, or other factors.

Key Features of Dynamic Websites:

1. User Interaction: They allow users to submit data (e.g., login, forms, comments), and the content is updated based on the user's input.

2. Database Integration: Dynamic websites are often connected to databases, allowing them to display real-time data (e.g., product listings, user profiles).

3. Content Management: Websites like blogs or e-commerce platforms rely on dynamic content that can be easily updated through content management systems (CMS).

4. Personalization: They can provide personalized experiences, such as showing tailored product recommendations or user-specific dashboards.

5. Backend Processing: Server-side languages (e.g., PHP, Node.js, Python) process data and generate dynamic responses.

Examples of Dynamic Websites:

- Social media platforms (e.g., Facebook, Twitter)
- E-commerce websites (e.g., Amazon, eBay)
- News portals (e.g., BBC, CNN)

Dynamic websites provide richer and more engaging user experiences, making them ideal for applications where content needs to change frequently or based on user interaction.

5. JavaScript (JS)

- JavaScript is a programming language that runs in the browser, making websites interactive. For example, JS is used to create slideshows, validate forms, and respond to user actions (like clicking a button).

- It is frontend-focused when running in the browser.

6. React

- React is a JavaScript library for building user interfaces, especially single-page applications (SPAs). It allows developers to build reusable components, making the development process more efficient and scalable. React is designed for creating dynamic web applications where content changes frequently without refreshing the page.

7. Node.js

- Node.js is a runtime environment that allows JavaScript to run on the server, outside of the browser. With Node.js, developers can use JavaScript for backend development, making it possible to build full-stack applications (both frontend and backend) with a single language.

- Node.js uses the V8 engine (same as Chrome) to execute JavaScript code.

8. SQL

- Structured Query Language (SQL) is used for managing databases. It allows you to retrieve, insert, update, and delete data from databases, making it essential for dynamic websites that handle user data (e.g., login systems, product catalogs, etc.).

9. AWS (Amazon Web Services)

- AWS is a cloud computing platform that offers a wide range of services such as storage, computing power, and databases. It’s often used for hosting websites, storing large amounts of data, and scaling applications efficiently.

JavaScript Engines

- JavaScript engines are responsible for executing JS code in different environments:
  - V8 engine – used by Chrome and Node.js.
  - SpiderMonkey – used by Firefox.
  - JavaScriptCore – used by Safari and iOS devices.
  - Chakra – used by Microsoft's Edge browser.

The Origins of JavaScript

- JavaScript was invented by Brendan Eich in 1995, originally called Mocha before being renamed JavaScript. It was created to make web pages dynamic and interactive.

ECMAScript (ES) is the standard for JavaScript, defining the language's rules and features. JavaScript follows this specification, ensuring consistency across different platforms like browsers and servers.

- ECMAScript 1 (1997): The first official version that standardized JavaScript.
- ECMAScript 3 (1999): Introduced key features like regular expressions and error handling.
- ECMAScript 5 (2009): Added strict mode, JSON support, and improved array methods.
- ECMAScript 6 (2015): A major update with new features like `let`, `const`, arrow functions, classes, and modules. This version marked a turning point in modern JavaScript development.
- Annual Updates: After ES6, new ECMAScript versions are released yearly (e.g., ES7, ES8, etc.), adding features like async/await, optional chaining, and more.

ECMAScript continues to evolve, making JavaScript more powerful and efficient for developers.

Key Differences Between Java and JavaScript

- Java is a statically typed language, meaning variables must be declared with their data type (e.g., `int x = 24;`).

- JavaScript, on the other hand, is a dynamically typed (or loosely typed) language. This means you can declare variables without specifying their type (e.g., `let y = 24;`).

Compiled vs Interpreted

- Java uses a compiler, which converts the code into bytecode that the Java Virtual Machine (JVM) can run.

- JavaScript uses an interpreter, which reads and executes code line by line in the browser or runtime environment (like Node.js).

Dynamically Typed Language (JavaScript)

- In JavaScript, you can assign different types of values to the same variable. For example:

javascript

```
let x = 24;  // number
x = "Hello";  // now x is a string
```

- This flexibility makes JavaScript dynamically typed or loosely coupled, meaning you don't need to define the variable type when writing code.

Summary

Static websites are built with technologies like HTML, CSS, and Bootstrap. These sites can't handle user input effectively. JavaScript and frameworks like React help create dynamic, interactive websites. Node.js allows JavaScript to run outside the browser, enabling full-stack development. Understanding how JavaScript engines work and the key differences between Java and JavaScript will help you build more efficient and responsive web applications.