

Welcome to Software Development!

Hey there! In this 2-week course, you'll learn how software—like apps and websites—works, starting with the **web and internet**. No experience needed—just curiosity! Each week, we'll have one fun class (2.5 hours) to explore coding and what developers do. Let's dive in!

Week 1: The Web, Internet, and First Steps in Coding

What You'll Learn:

- How the internet and web power apps like Twitter/X.
- Their history (from 1960s to now).
- How to write your first code for a webpage.

1. The Internet and Web: The Big Story

- **Internet:** A giant network connecting computers worldwide.
 - **History:**
 - 1960s: ARPANET links research computers.
 - 1990s: Dial-up brings internet to homes.
 - Now: Wi-Fi, cloud, and smart devices make it everywhere!
 - Think of it like roads connecting cities.
- **Web:** Pages (like google.com) you access via browsers.
 - **History:**
 - 1989: Tim Berners-Lee creates the web with HTML.
 - 1995: JavaScript adds clicks and animations.
 - Now: Web runs apps like Gmail and Netflix.
 - Think of it like books in a library, delivered by the internet.
- **Fun Fact:** Early websites were just text! Today's web is interactive thanks to code.

2. How the Web Works

- **Client-Server:** Your browser asks a server for a page (e.g., you → google.com → Google's server).
- **HTTP:** Rules for sending/receiving data (like ordering food).
- **DNS:** Turns "google.com" into an address (like a phonebook).
- **HTML, CSS, JavaScript:**
 - HTML = structure (like a skeleton).
 - CSS = style (like clothes).

- JavaScript = actions (like a brain).
- **Try This:** Right-click a website → “View Page Source” to see its HTML!

3. Your First Code

We'll use **JavaScript** to make a webpage do stuff!

Variables: Store info, like `let name = "Alex";`.

Functions: Reusable actions, like:

```
JavaScript
function sayHello() {
  alert("Hello, World!");
}
```

Example: A button that pops up a message:

html

```
Java

<h1>Welcome!</h1>

<button onclick="sayHello()">Click Me!</button>

<script>

  function sayHello() {

    alert("Hello, World!");

  }

</script>
```

Activity: Change the message to your name!

4. Tools You'll See

- **CodePen:** Online editor to write code.
- **VS Code:** Pro tool for coding.
- **Chrome:** Use “Inspect” to peek at webpages.
- **GitHub:** Where developers save and share code.

5. Homework (Optional)

- Visit a favorite site and check its source code.
 - Play with our CodePen—change the button’s text.
 - Think: What app would you love to build?
-

Week 2: Building a Webpage and Exploring Coding Paths

What You'll Learn:

- Make a working webpage (a to-do list!).
- Discover different types of developers.
- Plan your next steps in coding.

1. Coding a To-Do List

Let’s build something cool with HTML, CSS, and JavaScript!

- **New Stuff:**
 - **DOM:** Change the page (e.g., add text).
 - **Events:** Handle clicks or typing.
 - **Arrays:** Lists, like `let tasks = ["Study", "Code"];`

Example: A to-do list:

html

Unset

```
<h1>To-Do List</h1>
<input id="taskInput" placeholder="Add a task">
<button onclick="addTask()">Add</button>
<ul id="taskList"></ul>

<style>
```

```

    input { padding: 5px; }
    #taskList li {
        list-style: none; /* Removes default bullets */
    }
</style>

<script>
    function addTask() {
        let input = document.getElementById("taskInput");
        let list = document.getElementById("taskList");
        let task = input.value.trim(); /* Trims whitespace from input
*/

        if (task) {
            let li = document.createElement("li");
            li.appendChild(document.createTextNode(task));
            list.appendChild(li);
            input.value = "";
        } else {
            alert("Please enter a task.");
        }
    }
</script>

```

- **Activity:** Try changing the list's color or adding a warning for empty tasks.

2. Types of Developers

Coding has many paths—here's a peek:

- **Front-End:** Builds what you see (e.g., buttons on Twitter/X).
 - Uses: HTML, CSS, JavaScript, React.
- **Back-End:** Handles data (e.g., saving your posts).
 - Uses: Python, Node.js, databases.
- **Mobile:** Makes apps like Instagram.
 - Uses: Swift (iOS), Kotlin (Android).

- **Others:**
 - Data Science: Analyzes data.
 - Game Dev: Builds games like Fortnite.
 - DevOps: Manages servers.
- **Fun Fact:** The web inspired these—JavaScript started it all in 1995!
- **Think:** Which sounds fun to you?

3. Making Good Software

Clean Code: Use clear names (e.g., `taskList`, not `x`). Add comments:

JavaScript

```
// Add task to list

list.innerHTML += `<li>${task}</li>`;
```

Improved by:

- Removing the backticks and template literal syntax, as they are not necessary for this simple string concatenation.
- **Testing:** Check for bugs (e.g., what if you add an empty task?).
- **Teamwork:** GitHub lets developers share code (created in 2005).
- **Try This:** Suggest a test for our to-do app (e.g., “Don’t allow super-long tasks”).

4. What’s Next?

- **Start Here:** Learn more JavaScript or try Python.
- **Projects:** Build a webpage about your hobby or a quiz.
- **Resources:**
 - [freeCodeCamp](#) (free lessons).
 - [MDN Web Docs](#) (web guide).
 - YouTube: Search “JavaScript beginner tutorial.”
- **Tip:** Stuck? Google it! Every coder does.
- **Think:** What’s one small project you’d try?

5. Homework (Optional)

- Build a simple webpage (e.g., list your favorite movies).
- Check out one resource (e.g., freeCodeCamp).
- Share your project idea in class!