Here's a suggested **study schedule** for mastering the basics of **HTML** and **CSS**. This plan assumes you're dedicating about 1 to 2 hours per day, and the schedule spans 3 weeks. It covers the fundamentals of both HTML and CSS.

**Week 1: Introduction to HTML and Basic Structure**

**Day 1: Introduction to HTML**

* Overview of HTML and its role in web development
* Basic structure of an HTML document (<html>, <head>, <body>)
* The purpose of the <title> tag and <meta> tags

**Day 2: Working with Text Elements**

* Headings (<h1>, <h2>, etc.) and paragraphs (<p>)
* Emphasizing text (<strong>, <em>)
* Using line breaks (<br>) and horizontal rules (<hr>)

**Day 3: Lists and Links**

* Unordered (<ul>) and ordered lists (<ol>)
* List items (<li>)
* Creating hyperlinks (<a>) and attributes like href, target

**Day 4: Working with Images and Multimedia**

* Embedding images (<img>), setting attributes (src, alt, width, height)
* Audio (<audio>) and video (<video>) tags
* Setting controls for media elements

**Day 5: HTML Forms**

* Introduction to forms (<form>)
* Form elements: text fields (<input>), text areas (<textarea>), and buttons (<button>)
* Adding labels and placeholders to form fields
* Different form types: submit, radio, checkbox, etc.

**Day 6: HTML Tables**

* Creating tables with <table>, <tr>, <th>, <td>
* Using table captions (<caption>)
* Table attributes like colspan and rowspan

**Day 7: Recap and Practice**

* Review the first week's HTML concepts
* Practice by creating a simple webpage using text, links, images, and forms

**Week 2: Intermediate HTML and Introduction to CSS**

**Day 8: HTML Semantics**

* Importance of semantic HTML
* Semantic tags: <header>, <footer>, <article>, <section>, <nav>
* Benefits for accessibility and SEO

**Day 9: Introduction to CSS**

* What is CSS and why is it important?
* Basic CSS syntax: selectors, properties, and values
* Inline, internal, and external CSS
* Adding CSS to HTML using <style> and external stylesheets

**Day 10: CSS Selectors and Properties**

* CSS selectors: element, class (.class), and ID (#id)
* CSS properties: color, font, background, margin, padding, border
* Grouping, combining, and chaining selectors

**Day 11: Box Model in CSS**

* Introduction to the box model: content, padding, border, margin
* Understanding box-sizing property
* Adjusting width and height with the box model

**Day 12: CSS Positioning and Layout**

* Positioning elements: static, relative, absolute, fixed, sticky
* CSS Flexbox layout (introduction)
* Aligning items using justify-content and align-items

**Day 13: Styling Text and Fonts**

* CSS properties for text: font-family, font-size, line-height, text-align
* Adding web fonts using @font-face and Google Fonts
* Styling links (:link, :visited, :hover, :active)

**Day 14: Recap and Practice**

* Review the CSS topics covered in Week 2
* Practice creating a simple layout with Flexbox and basic styles

**Week 3: Advanced HTML and CSS Techniques**

**Day 15: CSS Backgrounds and Borders**

* Setting background colors and images (background-color, background-image)
* Using background-position, background-repeat, background-size
* Border properties: border, border-radius, box-shadow

**Day 16: CSS Transitions and Animations**

* Introduction to CSS transitions (transition-property, transition-duration)
* Creating simple animations with @keyframes
* Animating properties like opacity, transform, and color

**Day 17: Advanced Flexbox Layouts**

* Advanced usage of Flexbox: flex-grow, flex-shrink, flex-basis
* Creating complex layouts with Flexbox
* Aligning and centering elements using Flexbox

**Day 18: CSS Grid Layout**

* Introduction to CSS Grid: grid-template-columns, grid-template-rows
* Defining grid items: grid-column, grid-row
* Creating complex grid-based layouts

**Day 19: Responsive Web Design**

* Introduction to media queries for responsive design
* Adapting layouts for different screen sizes (mobile-first approach)
* Using @media to target specific device widths

**Day 20: CSS Variables and Custom Properties**

* Introduction to CSS variables (--variable-name)
* Using variables for colors, font sizes, margins, etc.
* Benefits of CSS custom properties for maintainable code

**Day 21: Recap and Final Project**

* Recap of HTML and CSS concepts learned
* Build a final project: Design a simple personal website or portfolio page using HTML and CSS
* Make the page responsive with Flexbox, Grid, and media queries

**Final Tips:**

* **Practice** is key to mastering HTML and CSS. Try building different layouts and styles as you progress.
* Create small projects after each major concept to reinforce your learning.
* **Don’t rush**: Spend more time on understanding how CSS works with the layout, positioning, and styling.
* Use **developer tools** in browsers (like Chrome DevTools) to inspect and manipulate your HTML and CSS in real-time.

By the end of this schedule, you should have a solid understanding of HTML and CSS, along with the ability to create basic and responsive web pages!

Here's a suggested **study schedule** for mastering the basics of **JavaScript**. This plan assumes you're dedicating about 1 to 2 hours per day, and can be adjusted based on your pace. The schedule will spread the topics across 3 weeks.

**Week 1: Fundamentals and Basic Concepts**

**Day 1: Introduction to JavaScript**

* Overview of JavaScript and its role in web development
* How JavaScript works in the browser
* Difference between JavaScript and other programming languages

**Day 2: Setting Up JavaScript**

* How to add JavaScript to HTML (inline, internal, external)
* Introduction to Developer Tools for JavaScript debugging

**Day 3: Basic Syntax and Data Types**

* JavaScript syntax overview
* Working with variables (var, let, const)
* Primitive data types (String, Number, Boolean, Undefined, Null)

**Day 4: Operators and Type Conversion**

* Arithmetic, assignment, comparison, and logical operators
* Type conversion and coercion in JavaScript

**Day 5: Control Flow (Part 1)**

* Conditional statements: if, else, else if
* Using the switch statement for multiple conditions

**Day 6: Control Flow (Part 2)**

* Working with loops: for, while, do...while
* for...in and for...of loops

**Day 7: Recap and Practice**

* Review the first week's concepts
* Practice by writing small programs to apply the learned concepts

**Week 2: Intermediate Concepts**

**Day 8: Functions (Part 1)**

* Introduction to functions and defining functions in JavaScript
* Function declarations and function expressions

**Day 9: Functions (Part 2)**

* Parameters and arguments in functions
* Understanding function scope (Global vs Local)
* Returning values from functions

**Day 10: Arrays (Part 1)**

* Introduction to arrays and how to create them
* Accessing and modifying array elements

**Day 11: Arrays (Part 2)**

* Common array methods: push, pop, shift, unshift, map, filter, etc.
* Iterating over arrays

**Day 12: Objects (Part 1)**

* Introduction to objects in JavaScript
* Creating and accessing object properties

**Day 13: Objects (Part 2)**

* Adding and modifying properties in objects
* Creating methods inside objects
* Iterating over objects

**Day 14: Recap and Practice**

* Review arrays and objects concepts
* Practice by building a small project (e.g., a contact list or a to-do app)

**Week 3: Advanced Concepts and Modern JavaScript**

**Day 15: JavaScript Events (Part 1)**

* Introduction to events in JavaScript
* Using addEventListener to handle events
* Common events like click, keypress, etc.

**Day 16: JavaScript Events (Part 2)**

* Event propagation: Bubbling and Capturing
* Handling events on the DOM

**Day 17: DOM Manipulation (Part 1)**

* Introduction to the Document Object Model (DOM)
* Selecting elements using getElementById, querySelector, etc.
* Modifying text and styles of HTML elements

**Day 18: DOM Manipulation (Part 2)**

* Adding and removing elements dynamically
* Modifying attributes and classes

**Day 19: Error Handling in JavaScript**

* Introduction to errors in JavaScript
* Using try, catch, and finally
* Throwing errors with throw

**Day 20: Asynchronous JavaScript (Part 1)**

* Introduction to asynchronous programming
* Using callbacks to handle asynchronous operations

**Day 21: Asynchronous JavaScript (Part 2)**

* Introduction to Promises
* Working with then, catch, finally
* Using async and await (ES6+)
* Recap and practice

**Week 4 (Optional Extension): Modern JavaScript and Advanced Topics**

**Day 22: ES6 Features (Part 1)**

* Introduction to ES6+ features: let, const
* Arrow functions and template literals

**Day 23: ES6 Features (Part 2)**

* Destructuring assignment, Spread, and Rest operators
* Modules and imports/exports

**Day 24: JavaScript Classes**

* Introduction to classes in JavaScript
* Creating and instantiating classes
* Constructor functions and methods

**Day 25: JavaScript Classes and Inheritance**

* Inheritance and super keyword
* Getter and Setter methods
* Static methods in classes

**Day 26: Working with APIs (Part 1)**

* Introduction to APIs and HTTP requests
* Using the Fetch API to make GET requests

**Day 27: Working with APIs (Part 2)**

* Handling JSON data from APIs
* Making POST requests and handling responses

**Day 28: Recap and Final Project**

* Recap of all the concepts learned
* Build a final project (e.g., a weather app using API requests or a simple game with JavaScript)

**Final Tips:**

* Dedicate time for practice and experiments each day. Working on small projects can solidify your understanding.
* Take breaks and don’t rush through the material. The key is consistent practice and review.
* If you're already familiar with some topics, feel free to move faster through them.

By the end of this schedule, you should have a solid understanding of JavaScript basics and be able to work on basic projects with it!