Week 1: HTML/CSS Fundamentals

Day 1: Introduction to HTML

- Basic HTML structure

- HTML tags and attributes

Day 2: More HTML

- HTML headings, paragraphs, links

- HTML lists (unordered, ordered, definition)

Day 3: Styling with CSS

- Introduction to CSS

- CSS syntax and selectors

Day 4: CSS Box Model

- Understanding the box model

- CSS margin, padding, border

Day 5: CSS Display and Positioning

- Display property (block, inline, inline-block)

- CSS positioning (static, relative, absolute)

Day 6: Responsive Design Basics

- Introduction to responsive web design

- CSS media queries

Day 7: Project: Build a Simple Web Page

- Apply HTML and CSS concepts learned so far to create a basic web page

**Week 2: Intermediate HTML/CSS**

Day 8: CSS Flexbox

- Introduction to Flexbox

- Flexbox properties and values

-Day 9: CSS Grid

- Introduction to CSS Grid

- Grid properties and values

Day 10: Advanced CSS Selectors

- Pseudo-classes and pseudo-elements

- Attribute selectors

Day 11: CSS Transitions and Animations

- CSS transitions

- CSS keyframe animations

Day 12: Introduction to CSS Pre-processors

- SASS or LESS basics

Day 13: Project: Create a Responsive Layout

- Build a responsive layout using Flexbox/Grid and media queries

Day 14: Project: Build a Portfolio Website

- Apply advanced CSS techniques to create a portfolio website

**Week 3: Introduction to JavaScript**

Day 15: Introduction to JavaScript

- JavaScript basics: variables, data types, operators

Day 16: JavaScript Functions

- Function declaration vs. function expression

- Function parameters and return values

Day 17: JavaScript Control Structures

- Conditional statements (if, else if, else)

- Loops (for, while, do-while)

Day 18: Document Object Model (DOM)

- Introduction to the DOM

- Accessing and manipulating DOM elements

Day 19: Event Handling

- Introduction to event listeners

- Handling user events

Day 20: Project: Create an Interactive Form

- Use JavaScript to validate form inputs and provide feedback

Day 21: Project: Build a Simple JavaScript Game

- Apply JavaScript concepts to create a simple game

Week 4: Intermediate JavaScript

Day 22: Advanced JavaScript Functions

- Arrow functions

- Function hoisting and scope

Day 23: JavaScript Arrays

- Array methods (push, pop, shift, unshift, etc.)

- Iterating through arrays

Day 24: JavaScript Objects

- Object creation and properties

- Object methods

Day 25: JavaScript Classes

- Introduction to ES6 classes

- Class inheritance

Day 26: Asynchronous JavaScript Basics

- Understanding asynchronous programming

- Call-back functions

Day 27: Promises in JavaScript

- Introduction to promises

- Promise chaining

Day 28: Project: Build a To-do List App

- Use JavaScript to create a to-do list application with CRUD operations

Week 5: Advanced JavaScript

Day 29: Introduction to ES6 Features

- Let and const

- Template literals

Day 30: Destructuring and Spread Syntax

- Array and object destructuring

- Spread syntax

Day 31: Higher-Order Functions

- Functions as first-class citizens

- Map, filter, and reduce

Day 32: Closures and Scope

- Understanding closures

- Lexical scope

Day 33: Asynchronous JavaScript with Async/Await

- Introduction to async/await

- Handling asynchronous code

Day 34: JavaScript Modules

- Introduction to ES6 modules

- Import and export statements

Day 35:Project: Build a Weather App

- Use asynchronous JavaScript to fetch weather data from an API and display it

Week 6: Introduction to Python

Day 36: Introduction to Python

- Python syntax and data types

- Variables and operators

Day 37: Python Control Structures

- Conditional statements (if, elif, else)

- Loops (for, while)

Day 38: Python Functions

- Function definition and parameters

- Return values

Day 39: Data Structures: Lists and Tuples

- Creating and manipulating lists

- Tuple basics

Day 40: Data Structures: Dictionaries and Sets

- Dictionary basics

- Set operations

Day 41: File Handling in Python

- Reading from and writing to files

- File modes

Day 42: Project: Build a Simple Python CLI Application

- Create a command-line application using Python

Week 7: Intermediate Python

Day 43: Exception Handling

- Handling errors with try-except blocks

- Finally block

Day 44: Working with Functions in Python

- Function arguments (default arguments, keyword arguments)

- Variable scope

Day 45: Python Modules and Packages

- Introduction to modules

- Creating and importing packages

Day 46: Object-Oriented Programming (OOP) Basics

- Classes and objects in Python

- Constructor and instance variables

Day 47: Inheritance and Polymorphism

- Inheriting from a superclass

- Method overriding

Day 48: Advanced OOP Concepts

- Class methods and static methods

- Class inheritance vs. composition

Day 49: Project: Build a Simple Python Web Scraper

- Use Python to scrape data from a website

Week 8: Advanced Python

Day 50: Working with Dates and Times in Python

- Date and time objects

- Date formatting

Day 51: Regular Expressions

- Introduction to regular expressions

- Pattern matching

Day 52: Generators and Iterators

- Creating and using generators

- Iterator protocol

Day 53: Decorators

- Function decorators in Python

- Using decorators for logging, caching, etc.

Day 54: Context Managers

- Using the `with` statement

- Creating custom context managers

Day 55: Python Testing

- Introduction to unit testing

- Using the unit test framework

Day 56: Project: Build a RESTful API with Python

- Create a simple RESTful API using Python and Flask/Django

Week 9: Web Development with Flask/Django

Day 57: Introduction to Flask/Django

- Overview of Flask/Django frameworks

- Setting up a development environment

Day 58: Routing in Flask/Django

- Creating routes and views

- URL mapping

Day 59: Templates in Flask/Django

- Rendering HTML templates

- Template inheritance

- \*\*Day 60:\*\* Working with Forms

- Form handling in Flask/Django

- Form validation

- \*\*Day 61:\*\* Database Integration

- Connecting Flask/Django to a database

- ORM basics

- \*\*Day 62:\*\* User Authentication

- Implementing user authentication and authorization

- Password hashing

- \*\*Day 63:\*\* Project: Build a Blogging Platform

- Create a simple blogging platform using Flask/Django

### Week 10: Advanced Web Development

- \*\*Day 64:\*\* RESTful API Design

- Principles of RESTful API design

- API endpoints and methods

- \*\*Day 65:\*\* Consuming APIs

- Making API requests in Flask/Django

- Handling API responses

- \*\*Day 66:\*\* Authentication and Authorization in APIs

- Implementing token-based authentication

- Authorization with JWT

- \*\*Day 67:\*\* WebSocket Communication

- Introduction to WebSockets

- Implementing real-time communication

- \*\*Day 68:\*\* Testing Flask/Django Applications

- Unit testing Flask/Django applications

- Integration testing

- \*\*Day 69:\*\* Deployment

- Deploying Flask/Django applications to a server

- Setting up production environment

- \*\*Day 70:\*\* Project: Build a Social Media Platform

- Create a social media platform with RESTful API using Flask/Django

### Week 11: Project Week

- \*\*Day 71:\*\* Capstone Project Planning

- Define project scope and requirements

- Create a project timeline

- \*\*Day 72-89:\*\* Capstone Project Development

- Work on implementing the capstone project

- \*\*Day 90:\*\* Capstone Project Presentation

- Present your capstone project to peers or mentors

- Reflect on your coding challenge journey

This breakdown should guide you through the remaining weeks of your 90-day coding challenge, focusing on JavaScript and Python topics, as well as practical web development using Flask/Django. Adjust the schedule and topics as needed based on your learning pace and interests. Good luck with your coding journey!