Detailed 3-Week Preparation Plan

April 24, 2025

Week 1: Detailed Python Foundation & Initial Exposure

Monday - Tuesday: Core Python Fundamentals

Morning (3-4 hours):

- **Introduction to Python:** What is Python, its uses in web development, setting up your development environment.
- **Basic Syntax:** Variables, data types (int, float, str, bool, list, tuple, dict, set), operators, basic I/O.
- **Control Flow:** Conditional statements (‘if‘, ‘elif‘, ‘else‘), looping (‘for‘, ‘while‘, ‘break‘, ‘continue‘, ‘pass‘).

Afternoon (3-4 hours):

- **Data Structures:** Lists, tuples, dictionaries, sets (creation, access, methods).
- **Practice:** Basic coding challenges utilizing fundamental concepts.

Evening (1-2 hours):

- **Introduction to Flask:** What is Flask? Setting up a project. ”Hello, World!” app. Running the development server.

Wednesday: Functions and Modules

Morning (3-4 hours):

- **Functions:** Defining, arguments, return values, scope, docstrings.

Afternoon (3-4 hours):

- **Modules and Packages:** Importing, standard library, basic use of common modules.
- **Practice:** Writing functions and organizing code into modules.

Evening (1-2 hours):

- **Basic Flask Routing:** Understanding routes and URL patterns. Returning simple text and HTML responses.

Thursday: Object-Oriented Programming (Basics)

Morning (3-4 hours):

- **Introduction to OOP:** Classes, objects, attributes, methods.

- **Defining Classes:** Creating classes, the ‘$_{i}nit_{‘method,instanceattributes.}$ **Afternoon (3-4 hours):**
  - **Instance Methods:** Defining and calling methods, the ‘self‘ parameter.
  - **Practice:** Creating simple classes and implementing basic behaviors.

Evening (1-2 hours):

- **Basic HTML Review:** Structure of an HTML document.

Friday: Error Handling and Basic JavaScript

Morning (3-4 hours):

- **Error Handling:** ‘try‘, ‘except‘ blocks, exception types, raising exceptions.

Afternoon (3-4 hours):

- **Basic JavaScript:** Variables, data types, operators, basic control flow, ‘console.log()‘.
- **Linking JavaScript to HTML:** Using the ‘¡script¿‘ tag.

Evening (1-2 hours):

- **Flask Routing with HTML Responses:** Serving simple HTML files from Flask routes.

Saturday: Working with Data Structures and Flask Templates (Intro)

Morning (3-4 hours):

- **Practice:** Manipulating lists, dictionaries, and sets.

Afternoon (2-3 hours):

- **Introduction to Flask Templates (Jinja2):** Understanding template files, passing variables, basic syntax (‘ ‘).

Sunday: Small Python Project and Continued Flask/HTML Review

Morning (3-4 hours):

- **Small Python Project:** Integrate core Python concepts.

Afternoon (2-3 hours):

- **Review:** Basic HTML structure and fundamental JavaScript concepts.

Advice for Staying Consistent and Managing Distraction

- **Understand Your Inconsistency and Distractions:** Identify triggers, track your time, and reconnect with your "why."
- **Implement Strategic Systems, Not Just Willpower:**
  - Set clear and small goals.
  - Time blocking.
  - Minimize distractions proactively.
  - Create a consistent routine.
  - Prepare your environment.
- **Build Momentum and Reward Progress:** Start small, celebrate wins, and visualize success.
- **Manage Your Mindset and Energy:** Practice self-compassion, prioritize rest, use the "Two-Minute Rule," and identify peak performance times.
- **Develop Strategies for When Distractions Arise:** Use the "Parking Lot" technique and the Pomodoro Technique, and practice mindfulness.

My "Advice" to Myself (If I Were You): *Okay, [Your Name], you know what you need to do. It won't always be easy, and you'll slip up sometimes. But the key is to build systems that make consistency easier than inconsistency. Start small, celebrate progress, and be kind to yourself when you falter. You've got this. Focus on showing up, even if it's just for a little while, and the momentum will follow.*

The ”See-Do-Reflect-Connect” Learning Cycle

1. See (Initial Exposure):

- Use resources like YouTube tutorials for an overview and basic understanding.

2. Do (Active Practice & Experimentation):

- **Replicate:** Recreate without constant reference.
- **Modify:** Experiment with parameters and approaches.
- **Solve Small, Targeted Problems:** Work on exercises related to the concept.

3. Reflect (Critical Thinking & Analysis):

- Identify challenges and struggles.
- Analyze errors to understand the ”why.”
- Connect to underlying principles.
- Formulate questions.

4. Connect (Seek Deeper Understanding & Broader Context):

- Consult official documentation.
- Explore multiple perspectives.
- Engage with online communities.
- Build small personal projects.

Why this is more effective: Emphasizes active learning, deeper understanding, personalized learning, and long-term retention.