ada developers
academy

Ada Developers Academy Course Syllabus

OBJECTIVE

The Ada Developers Academy core program teaches full-stack web development. We believe that full-stack web development is a skillset that allows students to grow quickly, and to choose what specialty of software development they want to pursue in their post-Ada careers. We cover Python, SQL, Flask, HTML/CSS, JavaScript, React and computer science fundamentals. Our complete curriculum is online and open-source.

We focus on teaching skills that are readily transferable from one technology stack to another, making Ada students adaptable and flexible candidates for all types of developer positions. Students have been successfully placed in internships and jobs working in many different programming languages and tech stacks.

STRUCTURE

The education portion of our program is split into three units focusing on programming fundamentals, backend development, and frontend development. Learning is supplemented through daily readings, problem sets, algorithm exercises, projects, and a three week capstone project.

LEARNING PATH

Precourse

Python 101

Review and practice core programming concepts:

- Variables and scope
- Functions
- Iteration
- Control Flow (conditionals)
- Lists and Dictionaries

Unit 1

Programming fundamentals

Further core programming concepts and learn computer science fundamentals. Explore test-driven development, exception handling, data structures, Big O, object-oriented programming and version control with Git/Github:

- Practice logical problem solving using guided techniques to break down problems and utilize pseudocode.
- Create and iterate over nested data structures
- Use version control to record, maintain, and collaborate on projects
- Identify, debug, and handle errors by using the stack trace, VS Code's debugger.
- Handle exceptions using try and except clauses
- Explore the relationship between memory allocations and arrays in Python
- Apply time and space complexity analysis to arrays and operations on arrays
- Use linear and binary search algorithms to search through array
- Write and utilize automated unit tests using Pytest
- Dive into Object-oriented programming by building and testing classes with instance and class methods in Python
- Utilize inheritance, composition, and decorators
- Understand the relationship between memory, references and values
- Analyze algorithms using Big O notation

LEARNING PATH

Unit 2

Intro to Back-end Web Development

Learn the fundamentals of back-end development. Design, build, and maintain a database using PostgreSQL. Design APIs in Flask to interact with databases and extend programs using third party APIs.

- Use CRUD actions and SQL Queries to build, modify and interact with databases
- Create one-to-many and many-to-many relationships between tables
- Design and build an API to explore HTTP, REST, and the client/server model relationship
- Build and design requests and responses using data and JSON
- Build an API in Flask to define endpoints, query params, create models, and perform CRUD actions using HTTP requests.
- Test APIs using Postman and Pytest
- Deploy APIs using Heroku
- Explore use cases for hash tables and recursion in problem-solving
- Collaborate in small groups to practice agile and maintaining projects using git

Unit 3

Intro to Front-end and Full-stack Development

Learn the basics of front-end development. Design and build static websites using HTML, CSS, and JavaScript. Develop web apps in React to present data and handle user interaction.

- Organize web content in HTML, using best semantic HTML practices
- Apply styles to web pages using CSS selectors by element type, class, ID, and relationship in the DOM
- Design the layout and flow of elements using CSS Grid and Flexbox
- Solve problems using variables, loops, and functions in ES6
- Utilize automated tests in JavaScript
- Manipulate the DOM and handle events in JavaScript
- Make and handle asynchronous API calls in JavaScript
- Create and render functional components in a React JS webapp
- Manage props and PropTypes in a React JS webapp
- Manage state using the useState hook in a React JS webapp
- Make API calls using the useEffect hook in a React JS webapp
- Practice full-stack development with distinct front-end and back-end layers