Modern Front-end JavaScript Literacy

Course Summary

Description

Modern Front-end JavaScript Literacy teaches all of the concepts and language students need in order to communicate and collaborate effectively, and gives students hands-on experience with JavaScript and client-side development tools. The course goes beyond most introductory web development courses and teaches the best practices, tools, and principles that are used by professional developers.

The three-day version is available for people who have prior experience with web development or JavaScript.

The five-day version is designed for people who have no prior web development or programming experience. It teaches everything in the 3-day course, plus HTML, CSS, and fundamental best practices of web development.

Topics

- Web Development Fundamentals
- HTML5
- CSS3
- Responsive Design Techniques
- JavaScript History
- Writing JavaScript
- Test-Driven Development
- Building with Webpack
- Web APIs

Audience

This course is for junior web developers, developers who are new to front-end web development, and professionals who work with JavaScript programmers and need to be able to communicate and collaborate with them

Prerequisites

For the three-day course, students should have at least an intermediate knowledge of web development and some experience with creating web sites using HTML, CSS, and JavaScript.

For the five-day course, students should be comfortable working with a Windows or Mac computer and should have some exposure to web development.

Duration

Three to five days

Modern Front-end JavaScript Literacy

Course Outline

I. Introduction to Web Development

- A. The Web Platform
- B. Command Prompt
  - 1. Basic Commands
  - 2. Know Your Shell
- **Lab: Working with the Unix/Linux Command Shell**
- C. How the Web Works
- D. Understanding Protocols
  - 1. TCP/IP
  - 2. DNS
  - 3. HTTP
  - 4. FTP
  - 5. SMTP
- **Lab: Configuring an HTTP (Web) Server**
- E. How Web Browsers Work
- **Lab: Working with Chrome Developer Tools–Element Tab**
- F. How Web Servers Work

II. HTML

- A. Early History of HTML
- B. The HTML Dark Ages
- C. XHTML
- D. HTML5
- E. What's New in HTML5?
- F. Purpose of HTML
- G. Hypertext
- H. Markup Language
- I. Meta-characters
- J. Entities
- K. Elements
- L. Element Example
- M. Tags
  - 1. Most Common Tags
- N. Common Empty Elements
- O. Attributes
- P. Comments
- Q. White Space
- R. Accessibility
- S. WCAG 2.0 Conformance Levels
- T. HTML Elements
- **Lab: Exploring vim**
- **Lab: Writing HTML in vim**
- **Lab: Working with Code Editors-ATOM**
- U. All You Need to Know: HTML
- **Lab: Creating an HTML Form**

III. Cascading Style Sheets

- A. How CSS Works
- B. Default Styles
- C. CSS Rules
- D. The Box Model
- E. The Box Model
- F. Intro to Selectors
  - 1. Basic Selectors
  - 2. Attribute Selectors
  - 3. Pseudo-Classes and Pseudo-Elements
  - 4. Combining Selectors
  - 5. Using Multiple Selectors
  - **Lab: Using CSS Selectors**
- G. Cascading
- H. Values and Units
- I. CSS Layout
- J. Element Types
  - 1. block Elements
  - 2. inline Elements
- K. Positioning Methods
  - 1. static
  - 2. relative
  - 3. absolute
  - 4. fixed
- L. Flexbox Layout
- **Lab: Positioning with CSS**
- M. Responsive Design (RWD)
  - 1. Why RWD?
- N. CSS Libraries
  - 1. Bootstrap
  - 2. Reset Style Sheets
  - 3. Font Awesome
  - 4. Material Design
  - 5. Materialize
  - 6. Battle of the Frameworks
  - 7. Others
- O. What's Wrong with CSS?
- P. What Is a CSS Preprocessor?
  - 1. CSS Preprocessors

IV. Introduction to JavaScript

- A. History of JavaScript
- B. JavaScript is NOT Java
- C. Is JavaScript "Real" Programming?
- D. Where Can You Use JavaScript?
- E. How JavaScript Works
- F. Is JavaScript Slow?
- G. Client-Side vs. Server-Side Code
- H. Browser Engine
- I. JavaScript Engine
- J. JavaScript Basics
  - 1. JavaScript Syntax
  - 2. JavaScript Data Types
  - 3. JavaScript Primitives
  - **Lab: Using JSBin**

Due to the nature of this material, this document refers to numerous hardware and software products by their trade names. References to other companies and their products are for informational purposes only, and all trademarks are the properties of their respective companies. It is not the intent of ProTech Professional Technical Services, Inc. to use any of these names generically

Modern Front-end JavaScript Literacy

Course Outline (cont'd)

- 4. Variables and Arrays
- 5. Creating and Using Variables
- 6. Creating and Using Arrays
- 7. JavaScript Operators
- **Lab: Using Chrome Developer Tools – JavaScript Console**
- **Lab: Using Array Methods**
- 8. Functions
- 9. Global Functions
- 10. Custom Functions

- K. JavaScript Objects
  - 1. Built-in Objects
  - 2. Creating Objects
  - 3. Using Objects
  - **Lab: Using JavaScript Objects**
  - 4. Prototypal Inheritance
- L. The Document Object Model
  - 1. What Is the DOM?
  - 2. Understanding Nodes
  - 3. EventTarget
  - 4. DOM Events
  - 5. DOM Manipulation with JavaScript
  - 6. DOM Manipulation with JQuery
  - 7. DOM Manipulation with React
  - **Lab: Performing DOM Manipulation**
- M. Survey of Popular Libraries and Frameworks
  - 1. JQuery
  - 2. Angular
  - 3. Backbone
  - 4. React
- **Lab: Using jQuery to Build a Chat Interface**
- **Lab: Using AJAX with jQuery**
- N. REST
- **Lab: Using JSON and REST to Work with Spark Rooms**
- **Lab: Using JavaScript Functions to Format Room Data**
- O. Modularity
  - 1. Why Is Modularity Important?
  - 2. CommonJS
  - 3. RequireJS
  - 4. ES6 Modules
  - 5. Package Management with npm

V. Front-end Tools and Techniques

- A. Git
  - 1. What Is Version Control?
  - 2. History of Git
  - 3. What Is Git?
  - 4. Three States of Git
  - 5. Git Workflow
  - **Lab: Controlling Your Versions with Git**
- B. Reproducible Builds
  - 1. Why Automate Your Build?
  - 2. Build Requirements
  - 3. npm
- **Lab: Initializing npm**
  - 4. node_modules
  - 5. package.json
  - 6. Npm Install
  - **Lab: Using npm**
- C. Task Runners
  - **Lab: Setting Up a Task Runner**
  - 1. Gulp
  - 2. gulpfile.js
  - 3. Gulp 4 API
  - 4. gulp.task
  - 5. gulp.src
  - 6. gulp.dest
  - 7. gulp.watch
  - 8. Run Gulp
- D. Static Code Analysis
  - 1. Lint Tools
  - 2. Two Ways to Configure ESLint
  - 3. ESLint: What Can Be Configured?
- **Lab: Automating Linting**
  - 4. ESLint Rules
- E. Browser Development Tools
- **Lab: Using Chrome Developer Tools - Sources Tab**
- F. Test-Driven Development
  - 1. Goal of TDD
  - 2. TDD Steps
  - 3. The TDD Cycle
    - a) Red
    - b) Green
    - c) Refactor
  - 4. Assertions
  - 5. JavaScript Testing Frameworks
  - 6. JS Exception Handling
  - 7. Jasmine Overview
  - 8. How Jasmine Works
  - 9. Test Suites
  - 10. Specs
  - 11. Expectations
  - 12. Matchers
  - 13. TDD vs. BDD
- **Lab: Getting Started with Jasmine**
- G. Webpack
  - 1. How webpack Works