

**Developing Salesforce Lightning Web Components**

**Course Number:** SF-110
**Duration:** 5 days

**Overview**

This Developing Salesforce® Lightning Web Components training course teaches attendees everything they need to know to begin creating Salesforce Lightning Web Components using Visual Studio Code and the Salesforce command line interface. Students also learn how to incorporate styles, navigate the Lightning Architecture, and more.

**Prerequisites**

All students must have a strong technical background are already familiar with Apex and Visualforce or Aura Components such as Salesforce developers, OR have object-oriented development experience and are looking to quickly develop skills on Salesforce Lightning.

**Materials**

All Salesforce training students will receive comprehensive courseware.

**Software Needed on Each Student PC**

* Salesforce CLI installed
* Visual Studio Code, with Salesforce Extension Pack
* Latest version of Google Chrome browser
* Web access to create and use a Salesforce Developer Edition practice site
* Access to email during class to follow Salesforce verification links

**Objectives**

* Review W3C Web Components
* Understand Lightning web component architecture
* Incorporate Salesforce DX, Visual Studio Code and Salesforce CLI
* Use JavaScript, CSS, and HTML/XHTML as they pertain to Lightning Web Components
* Consume the Lightning data service
* Debug and test web components
* Use localization, locker service, and event handling
* Understand Aura comparison and coexistence

**Outline**

* Introduction
	+ Course Overview
	+ Setting up a Practice Site
* VS Code Overview
	+ Command Palette
	+ Creating a Project
	+ Connecting to Practice Site
	+ Deploying to Practice Site
* Lightning Component Framework Overview
	+ Structure of a Lightning Web Component
		- Component Folder
		- Component HTML File
		- Component JavaScript File
		- Component Configuration File
		- Component CSS File
		- Component Tests
		- Component Namespaces
	+ Examining a Lightning Web Component
		- webcomponents.dev
		- Parent and Child Components
* Creating Your First Lightning Web Component
	+ Create the HelloWorld Component
	+ Update Component HTML File
	+ Update Component JavaScript File
	+ Update Component Configuration File
	+ Deploy Component to Practice Site
	+ Add Component to a Lightning Page
* HTML Templates
	+ Data Binding
	+ Conditional Rendering
	+ Rendering Lists
* Reactivity
	+ Fields
	+ Public Properties
	+ Reactivity and Data Types
* XML Configuration File
	+ Targets
	+ Target Configs
	+ Objects
	+ Supported Form Factors
* Component Reference Guide
* Styling Lightning Web Components
	+ Lightning Design System
	+ Using CSS
* Access Salesforce Resources
	+ Static Resources
	+ Labels
	+ Internationalization
* Composition
	+ Compose Components
	+ Set Properties on Children
	+ Call Methods on Children
	+ Access Component Elements
* Hooks and Events
	+ Lifecycle Hooks
		- constructor()
		- connectedCallback()
		- renderedCallback()
	+ Events
		- Create and Dispatch Events
		- Handle Events
		- Event Best Practices
* Lightning Data Service
	+ Using Base Components
	+ Building Custom UI Compnents
	+ Display Data in a Table
* Wire Service
	+ Get Record Data
	+ Handle Errors
* Using Apex
	+ Importing Methods
	+ Exposing Methods to Components
	+ Wiring Apex Methods
	+ Import Objects and Fields
	+ Calling Apex Methods Imperatively
* Messenger Service
	+ Create a Message Channel
	+ Publish on a Message Channel
	+ Subscribe and Unsubscribe on a Message Channel
* Navigation
	+ Basic Navigation
	+ PageReference Types
* Toast Messages
* Streaming API
	+ Push Technology
	+ Bayeux Protocol and CometD
	+ empApi Module
* Testing Lightning Web Components
	+ Install Node.js
	+ Install Jest
	+ Write Jest Tests
	+ Run Jest Tests
* Questions and Answers/Conclusion