

**Introduction to Secure Coding for Web Developers**

**Course Number:** SEC-100
**Duration:** 1 day

**Overview**

In this intensive, live, online Introduction to Web Application Security training, developers directly explore common code vulnerabilities and how to mitigate them. The class is full of “uh, oh” moments when developers dramatically see the consequences of failing to write secure code, followed by the “ah, ha” satisfaction of correcting their code to make the vulnerabilities disappear.

**Note:** To ensure ample one-on-one engagement with the instructor, this class is capped at 12 people, overriding Accelebrate’s default cap of 15.

**Prerequisites**

Prior experience with HTML, CSS, JavaScript, and SQL is presumed.

**Materials**

This coding class includes electronic courseware and code examples.

**Software Needed on Each Student PC**

* Google Chrome and any other modern web browsers with which you would like to test
* JavaScript IDE of your choice
* Node.js installed
* Related lab files

**Objectives**

* Validate data to defend against injection and other attacks
* Prevent Cross-Site Scripting (XSS) vulnerabilities
* Gain a deep understanding of SQL injection attacks and how to defend against them

**Outline**

* Introduction
* Data Validation
	+ Understand the Business Logic for the Data You Accept
	+ Limit the Amount of Data You Are Willing to Accept
	+ Use a Whitelist to Sanitize Incoming Data
	+ Properly Handle Exceptions
	+ Reminder: The User Interface Can Always be Bypassed
	+ Working With Encoded Data
	+ Data Validation Checklist
* Cross-Site Scripting (XSS)
	+ Reflected Cross-Site Scripting
	+ What an Attacker Can Accomplish
	+ Persistent Cross-Site Scripting
	+ Preventing Cross-Site Scripting Vulnerabilities
	+ Testing for Cross-Site Scripting Vulnerabilities
	+ Cross-Site Scripting Prevention Checklist
* SQL Injection (SQLI)
	+ Exploiting a SQL Injection Vulnerability
	+ Blind SQL Injection Attacks
	+ Union-Based SQL Injection Attacks
	+ Second Order SQL Injection Vulnerabilities
	+ Direct Database Attacks
	+ Preventing SQL Injection Vulnerabilities
	+ Making Attacks More Difficult
	+ Identifying Existing Vulnerabilities in Your Code
* Conclusion