Course Glossary

Welcome! This alphabetized glossary contains many of the terms you'll find within this course. This comprehensive glossary also includes additional industry-recognized terms not used in course videos. These terms are important for you to recognize when working in the industry, participating in user groups, and participating in other certificate programs.

Term	Definition
Ajax (Asynchronous JavaScript and XML)	A set of web development techniques using various technologies, including JavaScript and XML/JSON, to create asynchronous web applications. It allows for updating parts of a web page without reloading the entire page.
Array	A data structure used to store a collection of elements, each identified by an index or a key.
Arrow Function	A concise way to write functions in JavaScript, introduced in ES6, with a simplified syntax compared to traditional function expressions.
Asynchronous	Code execution that doesn't happen sequentially, allowing other code to run while waiting for an operation to complete. Commonly used with callbacks, promises, and async/await.
Axios	A popular Promise-based HTTP client for making HTTP requests in the browser and Node.js. It provides an easy-to-use API and supports features like interceptors, cancellation, and more.
BOM (Browser Object Model)	A set of objects provided by the browser that represent the browser's window. It includes objects like window, navigator, history, screen, and location, allowing interaction with the browser itself.
Callback Function	A function passed as an argument to another function, to be executed later or upon the occurrence of a certain event.
Callback Hell	A situation where multiple nested callbacks make code difficult to read and maintain, often encountered in asynchronous JavaScript.
Closure	A function that retains access to variables from its parent scope even after the parent function has finished executing.
Conditional Statements	Constructs like if, else if, and else used to execute different code based on specified conditions.
DOM (Document Object Model)	A programming interface for web documents that represents the structure of HTML and XML documents, allowing scripts to manipulate the content and structure of web pages.
Error Handling	The process of managing errors that occur during the execution of a program. In JavaScript, errors can be caught and handled using constructs like try...catch, allowing developers to gracefully manage unexpected issues.
ES6/ES2015	Refers to the sixth edition of the ECMAScript standard, introducing new syntax and features like arrow functions, classes, let/const, and more.
Event	An action or occurrence recognized by a program that can be handled by event listeners to trigger specific functionality.
Fetch	A modern API for making network requests in browsers, used to fetch resources asynchronously from the server. It uses promises to handle responses.
Function Closure	A function that retains access to variables from its parent scope, even after the parent function has finished executing. This allows the inner function to access and manipulate the variables of the outer function.
Function Hoisting	In JavaScript, function declarations are hoisted to the top of their scope during the compilation phase. This means that you can call a function before it's declared in the code.
Function	A block of code that can be called and executed to perform a specific task or calculate a value.
JavaScript (JS)	A high-level, interpreted programming language used to create dynamic and interactive web content.
JSON (JavaScript Object Notation)	A lightweight data interchange format that is easy for humans to read and write and easy for machines to parse and generate.
Loop	A control flow statement that allows code to be executed repeatedly based on a condition. Examples include for, while, and do...while loops.
Module	A reusable piece of code that encapsulates related functionality, allowing for better code organization and reuse.
Node.js	An open-source, server-side JavaScript runtime environment that allows execution of JavaScript code outside of a web browser. It enables building scalable network applications.
NPM (Node Package Manager)	A package manager for JavaScript that allows developers to discover, install, and manage packages and dependencies for Node.js applications.
Object	A complex data type that stores key-value pairs, often used to represent real-world entities.
Promise Chaining	The practice of linking promises together using their then() method to execute asynchronous operations sequentially or handle multiple asynchronous operations.
Promise	An object representing the eventual completion or failure of an asynchronous operation, often used for handling asynchronous operations in a more readable and manageable way.
String	A primitive data type used to represent textual data. Strings are sequences of characters enclosed in single or double quotes.
Template Literals	Introduced in ES6, these allow embedding expressions into strings using backticks () and ${} syntax, providing a more readable way to create strings with variables.
TypeScript	A superset of JavaScript that adds static typing and other features to help catch errors and make JavaScript development more scalable and maintainable.
Variable	A named storage location used to hold data values that can change during the execution of a script.
Window Object	The global object in the browser environment that represents the browser window. It contains properties and methods that can be accessed globally within the browser.