

# Module 3 Glossary: JavaScript Programming for Web Applications

Term	Definition
<b>AJAX</b>	“Asynchronous JavaScript and XML” that encompasses more than asynchronous server calls through JavaScript and XML. It is not programming language or technology but rather a programming concept. Ajax represents a series of techniques that provide richer, interactive web applications through HTML, JavaScript, Cascading style sheets, and modifying the web page through the Document Object Model. The name is misleading though because nowadays, JSON is commonly used instead of XML.
<b>Anonymous Functions</b>	A type of function that has no name or we can say which is without any name. It is declared without any identifier and is often used as a parameter of another function. It is a common way to execute a function immediately after its declaration.
<b>Array</b>	A data structure that aids the programmer in the storage and retrieval of data by indexed keys. Arrays use a zero-based indexing scheme, meaning that the first element of an array has an index of zero. Arrays grow or shrink dynamically by adding or removing elements.
<b>Classes</b>	Classes act as a blueprint or template for building objects with similar characteristics and behaviours. A class encapsulates data (in the form of properties) and functions (in the form of methods) that work on that data.
<b>Client-Side Script</b>	A program that accompanies an HTML doc or embedded in HTML. Scripts run during load of a document or when an action is performed. They can be used to validate forms, process input, or dynamically create document elements. Embed a script in HTML, with the <script> tag in either of the following ways: <ul style="list-style-type: none"> <li>• &lt;script&gt; // JS code &lt;/script&gt;</li> <li>• &lt;script src="path name"&gt;&lt;/script&gt;</li> </ul> Use <noscript> tag for browsers with JavaScript disabled or ones that don't support JavaScript.
<b>Document Objects</b>	Document representing the main web page that gives access to all HTML elements on the page. When page is loaded HTML doc becomes a document. It is referred to with “document”.
<b>DOM</b>	“Document Object Model” is a programming interface (API) between HTML and JavaScript. It allows for dynamically accessing and updating content, structure, and style. JavaScript uses the DOM to access and modify web page elements in the browser.
<b>Element Nodes</b>	All HTML tags.
<b>Element Objects</b>	The most general base class that all element objects in a Document inherit. It only has methods and properties common to all elements. Everything in a HTML page is an element. And one element can have other elements nested within itself.
<b>Event</b>	An event is something either a browser or a user does that the JavaScript can react to such as a button click or when a user submits input on a form.
<b>Event Binding</b>	Refers to telling the browser that a function should be called whenever some ‘event’ occurs.
<b>Event Handlers</b>	A function that declares what to do when an action is performed such as the click of a button. Example: <pre>&lt;button type="button" onclick="showAnswers()"&gt; Show Solution &lt;script&gt;   function showAnswers() {     //code     alert("A")   } &lt;/script&gt; &lt;/button&gt;</pre> Note that showAnswers() is an event handler.
<b>Extend</b>	This keyword is used in class declarations or class expressions to create a class that <b>is</b> a child of another class.
<b>Functions</b>	Functions are modules of code that execute a particular task. They may take-in data, called arguments or parameters, and sometimes return data as well, called the return value. Functions are commonly defined with this syntax: <pre>function functionName() {   // function code;   // optional return statement; }</pre>
<b>IIFE</b>	“Immediately Invoked Function Expression” runs immediately after it is defined. After the function executes it cannot be called again elsewhere in the program. It is a type of self-executing function.
<b>Nodes</b>	The basis of all elements in the Document Object Model (DOM) structure.
<b>Objects</b>	Objects are instances created from a class. They are real-world entities that represent the characteristics defined by the class. Objects have a special set of properties that store data and methods that specify behaviours. These methods and properties can be accessed and changed to carry out specific tasks and communicate with other programs.
<b>Prototypes</b>	A function prototype lets you easily define and add properties or methods to an object. Prototypes exist for all objects that can be created with the keyword “new”. All object constructors create objects that inherit properties and methods

Term	Definition
	<p>that are defined by the prototype. At instantiation objects inherit the current state of the prototype. Note however, that scripts can override prototype properties and functions. Following is an example of using a prototype to add a method to the Car class:</p> <pre>function Car(make, model, year) {   this.make = make;   this.model = model;   this.year = year; } Car.prototype.getName = function() {   return this.make + ' ' + this.model + ' ' + this.year; }</pre>
Script	Offers developers means to modify and extend HTML documents in highly interactive ways. Scripts can be used to validate forms or to process input as it is typed. Scripts can be triggered by events that occur on a web page, such as the clicking of a button. Scripts can be used to dynamically create document elements on an HTML page.
Self-Executing Functions	Often used to initialize data or declare DOM elements. These functions can be anonymous.
Text Nodes	The nodes that contain actual text that go between an element start tag and end tag.
this	Keyword “this” refers to current instance of the object. The value of “this” can vary depending on how the object is called.



# Skills Network