Introduction to Web Development with HTML, CSS, JavaScript

Table of Contents

[MODULE 1 Introduction to Application Development 2](#_Toc180777849)

[Module 1 Summary 2](#_Toc180777850)

[Module 1 Glossary: Overview of Cloud Computing 2](#_Toc180777851)

[MODULE 2 HTML Overview 3](#_Toc180777852)

[Module 2 Summary 3](#_Toc180777853)

[Common HTML elements 3](#_Toc180777854)

[Module 2 Cheatsheet: HTML5 Overview 4](#_Toc180777855)

[Module 2 Glossary: HTML5 Overview 4](#_Toc180777856)

[MODULE 3 CSS Overview & HTML5 elements 4](#_Toc180777857)

[Special HTML Elements 4](#_Toc180777858)

[Additional HTML Elements: HTML <fieldset> and <legend> 4](#_Toc180777859)

[HTML and HTML5 Elements Cheat Sheet 5](#_Toc180777860)

[Module Summary 5](#_Toc180777861)

[Module 3 Glossary: CSS Overview & HTML5 Elements 5](#_Toc180777862)

[MODULE 4 JavaScript Programming for Web Applications 5](#_Toc180777863)

[JavaScript APIs 5](#_Toc180777864)

[Hands-on Lab: JavaScript - Browser Console 6](#_Toc180777865)

[Module Summary 6](#_Toc180777866)

[Module 4 Cheatsheet: JavaScript Programming for Web Applications 6](#_Toc180777867)

[Module 4 Glossary: JavaScript Programming for Web Applications 7](#_Toc180777868)

[MODULE 5 Career Opportunities and Final Project 8](#_Toc180777869)

## **MODULE 1 Introduction to Application Development**

## Module 1 Summary

* Front-end developers work on the parts of the website or app that the user sees and interact with.
* Back-end developers work on the logic and functionality that keeps the website or app running and responding to users’ inputs.
* Full-stack developers have both sets of skills.
* Front-end developers and back-end developers work closely together.
* Frameworks and libraries extend the functionality of coding languages such as JavaScript and Python.
* Common languages for front-end development include: HTML, CSS, and JavaScript.
* Common languages and frameworks for back-end development include: Python, Django, and Flask.
* Version control systems keep track of changes and resolve conflicts between them.
* CI/CD (Continuous Integration with Continuous Delivery/Deployment) is a best practice developers use to deliver frequent changes reliably.

## Module 1 Glossary: Overview of Cloud Computing

|  |  |
| --- | --- |
| Term | Definition |
| AngularJS | An open-source JavaScript framework for dynamic web applications. |
| Application Programming Interface (API) | Code that allows two software programs to communicate with each other. |
| Build Automation | Allows you to download dependencies, compile code, package binary code, run tests, deploy to production. |
| Build Automation Servers | Execute build-automation utilities on a scheduled or triggered basis. |
| Build Automation Utilities | Generate executables by compiling and linking code. |
| Continuous Integration/Continuous Deployment (CI/CD) | A method for releasing code and integrating it into code that has already been developed in order to prevent the application from breaking throughout the app's lifecycle. |
| Django | A framework for Python web development. |
| Dynamic Content | Data that is created each time a request is sent to a server. |
| Endpoint | The point at which an API connects with the software program. |
| Frameworks | Provide a standard way to build an application. Frameworks dictate architecture and program flow. |
| IDE | "Integrated Development Environment" helps create and manage code. |
| Inversion of Control | A predefined workflow where the developer is not in full control of how the application operates. |
| JavaScript Framework | An application framework written in JavaScript to create responsive sites. |
| LESS | "Learner Style Sheets" add more style and functions to CSS. |
| less.js | A JavaScript tool that converts LESS styles to CSS. |
| Libraries | Reusable collections of code. |
| Opinionated | Frameworks that have a lot of control are sometimes considered "opinionated". |
| Package Managers | Coordinate with file archivers to extract packages, verify check sums and digital certificates, locate, download, and install updates of existing software from a repository as well as manage dependencies. Common package managers include Debian Package Management System (DPMS), Red Hat Package Manager for Linux, Chocolatery for Windows, Homebrew and MacPorts for MacOS. |
| Packages | Archive files that include app files, instructions for installation, and metadata. |
| React.js | A JavaScript framework developed by Facebook that helps build and drop elements onto a page. |
| Responsive Design | A design technique that automatically resizes a display to adapt to a specific screen size. |
| Route | Allows front-end client to plug into the correct socket on the backend. They are the paths that network traffic takes from a virtual machine (VM) instance to other destinations. |
| SASS | "Syntactically Awesome Stylesheets" are an extension of CSS. |
| Static Content | A display of data that has been previously stored on a server. |
| Version Control | Allows you to revert to earlier versions of code, resolves conflicts between the same files, and split and merge different code branches. |
| Vue.js | A community-based JavaScript framework focused on UI. Includes UI components such as buttons and other visual elements, and is both a library and a framework. |

## **MODULE 2 HTML Overview**

## Module 2 Summary

* HTML provides the basic structure and content for a website using tags.
* Tags represent the elements of an HTML page.
* The HTML DOM Tree describes how a website is structured.
* HTML uses APIs to enhance the user experience, providing features for advanced animation, audio, and video.
* Scripting provides a more interactive user experience when browsing websites.
* It is recommended to not rely on scripting as it can be disabled.
* HTML5 sandboxes help manage iframe mashups.
* HTML5 Browser Support Tables describe which browsers support which HTML5 features.
* JavaScript is used to check if an element is supported by a browser.
* CSS provides consistent style and design throughout the website.
* There are two types of CSS layouts to design websites: fluid and fixed.

## Common HTML elements

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9SZWFkaW5ncy9Nb2R1bGUlMjAyL0NvbW1vbi1IVE1MLUVsZW1lbnRzLm1kIiwidG9vbF90eXBlIjoiaW5zdHJ1Y3Rpb25hbC1sYWIiLCJhZG1pbiI6ZmFsc2UsImlhdCI6MTcwMDY3MDY1NH0.X8VaEC6W70GIGKq7E8SNm0WB7bzNPtK-vLhfJgsl1OI>

## Module 2 Cheatsheet: HTML5 Overview

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9DaGVhdHNoZWV0cy9DMk0yJTIwY2hlYXRzaGVldCUyMHYxLjIubWQiLCJ0b29sX3R5cGUiOiJpbnN0cnVjdGlvbmFsLWxhYiIsImFkbWluIjpmYWxzZSwiaWF0IjoxNzAwNjcwNTE1fQ.1-QZuK-fIC9mU2Xi3YnG1CVzBh3kVxd0oQcBY2jhCtw>

## Module 2 Glossary: HTML5 Overview

|  |  |
| --- | --- |
| Term | Definition |
| DOM Tree | “Document Object Model” is the data representation of the objects that comprise the structure and content of a document on the web. |
| XML | An “eXtensible Markup Language” Designed to store and transport data allowing users to define their own markup languages, primarily to display documents on the web. |
| XHTML | An “eXtensible Hypertext Markup Language” similar to HTML but with stricter formatting rules. |
| Web Storage APIs | APIs that allow data storage in a browser. |

## **MODULE 3 CSS Overview & HTML5 elements**

## Special HTML Elements

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9SZWFkaW5ncy9Nb2R1bGUlMjAzL2NvbW1vbl9odG1sX3RhZ3NfYW5kX3N0cnVjdHVyYWxfZWxlbWVudHMubWQiLCJ0b29sX3R5cGUiOiJpbnN0cnVjdGlvbmFsLWxhYiIsImFkbWluIjpmYWxzZSwiaWF0IjoxNzAwNjcwNjU5fQ.uoh4yJ1sWDagsYFKjx7ar_CjU-pvR7JmNOH9oO2Cz7Y>

## Additional HTML Elements: HTML <fieldset> and <legend>

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9SZWFkaW5ncy9Nb2R1bGUlMjAzL0hUTUxfRm9ybV9MYWJfXzFfLm1kIiwidG9vbF90eXBlIjoiaW5zdHJ1Y3Rpb25hbC1sYWIiLCJhZG1pbiI6ZmFsc2UsImlhdCI6MTcwMDY3MDY1Nn0.cHB2WUkRSm7BU6c_k-VvoqKvevecN7lPRqrtazU6w40>

## HTML and HTML5 Elements Cheat Sheet

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9DaGVhdHNoZWV0cy9DMk0zX2NoZWF0c2hlZXRfdjEuMi5tZCIsInRvb2xfdHlwZSI6Imluc3RydWN0aW9uYWwtbGFiIiwiYWRtaW4iOmZhbHNlLCJpYXQiOjE3MDA2NzAyMjV9._SdR1RLoZa8vcC1bMxXuGSvZ8zGRE3n94NIVSv6gKE8>

## Module Summary

* CSS creates a uniform look throughout each element of each page of the website.
* CSS is usually coded in external style sheets and creates base styles for a website.
* CSS frameworks assists in implementing UI elements and creating dynamic web pages.
* CSS has two types of frameworks:
* Utility-first frameworks, which provide utility classes to help in building one's own styles and layouts.
* Component frameworks, which provide a wide selection of pre-styled components and templates that can be implemented onto a website.
* Plain (Vanilla) CSS lets developers write the styles and layouts of a website.
* HTML5 elements provide structure and function to websites.
* HTML5 uses the <input> tag to allow users to input information.

## Module 3 Glossary: CSS Overview & HTML5 Elements

|  |  |
| --- | --- |
| Term | Definition |
| Component Framework | Component frameworks provide pre-styled components and templates which are easy to add to any website. |
| CSS | "Cascading Style Sheets" is a style sheet language that describes how HTML elements are displayed. It is the design that is layered over the top of an HTML web page. |
| Fixed Layout | A fixed layout is a layout where you specify the height and width of elements, and those values remain the same regardless of which operating system or browser you use to access the website. |
| Fluid Layout | A fluid layout is a layout in which the height and width of elements is flexible and can expand or contract based on the browser window, the operating system, and other user preferences. |
| Utility Framework | The utility framework provides utility classes that are scoped to individual CSS properties, which helps in building custom designs in HTML files. |

## **MODULE 4 JavaScript Programming for Web Applications**

## JavaScript APIs

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9SZWFkaW5ncy9Nb2R1bGUlMjA0L0phdmFTY3JpcHQtQVBJcy5tZCIsInRvb2xfdHlwZSI6Imluc3RydWN0aW9uYWwtbGFiIiwiYWRtaW4iOmZhbHNlLCJpYXQiOjE3MDA2NzA2NjB9.xwDU8POn_u7RunO_3n1vntUPawOXxTGj7rEHWR9Q30E>

## Hands-on Lab: JavaScript - Browser Console

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9sYWJzL1RoZWlhJTIwTGFicy8wMyUyMC0lMjBKYXZhc2NyaXB0L0pTQmFzaWNzUHJhY3RpY2UubWQiLCJ0b29sX3R5cGUiOiJpbnN0cnVjdGlvbmFsLWxhYiIsImFkbWluIjpmYWxzZSwiaWF0IjoxNzAwNjcwMTE0fQ.oP_uAdRMWTpJtG0sGcT20V60ybz3z4qaPwR_FrE8TpU>

## Module Summary

* JavaScript is a scripting language that enables developers to add dynamic content to webpages.
* JavaScript variables are declared using the var keyword and take their type from the value assigned.
* Program execution is controlled by statements like If…Then…Else, Switch, For loops, and While loops.
* JavaScript uses blocks of code, called functions, that can be called from anywhere in the script.
* New methods and properties can be added to an object by modifying the prototype for that object.
* Prototypes allow you to define properties and methods for all instances of a specific object.
* Client-side scripts are programs that accompany HTML documents and are used by developers to incorporate more interactive elements.
* The script tag can incorporate a script within an HTML document or call a script from an external file.
* The Document Object Model (DOM) is the programming interface between HTML or XHTML and JavaScript.
* Developers can access HTML DOM elements from JavaScript scripts using the correct DOM notation.
* APIs are often used to access HTML DOM elements in web pages.

## Module 4 Cheatsheet: JavaScript Programming for Web Applications

<https://author-ide.skills.network/render?token=eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJtZF9pbnN0cnVjdGlvbnNfdXJsIjoiaHR0cHM6Ly9jZi1jb3Vyc2VzLWRhdGEuczMudXMuY2xvdWQtb2JqZWN0LXN0b3JhZ2UuYXBwZG9tYWluLmNsb3VkL0lCTURldmVsb3BlclNraWxsc05ldHdvcmstQ0QwMTAxRU4tU2tpbGxzTmV0d29yay9DaGVhdHNoZWV0cy9DMk00JTIwQ2hlYXRzaGVldCUyMHYxLjIubWQiLCJ0b29sX3R5cGUiOiJpbnN0cnVjdGlvbmFsLWxhYiIsImFkbWluIjpmYWxzZSwiaWF0IjoxNzAwNjcwNjIzfQ.Bp2I7ev9YGCf_SqDK9sYX6kPlx_ZV4Oy5oT31Z-oB0E>

## Module 4 Glossary: JavaScript Programming for Web Applications

|  |  |
| --- | --- |
| Term | Definition |
| AJAX | “Asynchronous JavaScript and XML” that encompasses more than asynchronous server calls through JavaScript and XML. It is not a 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. |
| Anonymous Functions | A type of function that has no name or 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. |
| Array | 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. |
| Classes | 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. |
| Client-Side Script | A program that accompanies an HTML document or is embedded in HTML. Scripts run during the 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: **<script> // JS code </script>** or **<script src="path name"></script>**. Use **<noscript>** tag for browsers with JavaScript disabled or ones that don’t support JavaScript. |
| Document Objects | Document representing the main web page that gives access to all HTML elements on the page. When the page is loaded, the HTML document becomes a document. It is referred to with “document”. |
| DOM | “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. |
| Element Nodes | All HTML tags. |
| Element Objects | 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 an HTML page is an element. And one element can have other elements nested within itself. |
| Event | 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. |
| Event Binding | Refers to telling the browser that a function should be called whenever some ‘event’ occurs. |
| Event Handlers | A function that declares what to do when an action is performed, such as the click of a button. Example: **<button type="button" onclick="showAnswers()"> Show Solution <script> function showAnswers() { //code alert("A") } </script> </button>**. Note that **showAnswers()** is an event handler. |
| Extend | This keyword is used in class declarations or class expressions to create a class that is a child of another class. |
| Functions | 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: **function functionName() { // function code; // optional return statement; }** |
| IIFE | “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. |
| Nodes | The basis of all elements in the Document Object Model (DOM) structure. |
| Objects | 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. |
| Prototypes | 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 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. Example: **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; }** |
| Script | Offers developers a 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 |

Top of Form

Bottom of Form

## **MODULE 5 Career Opportunities and Final Project**