Technology Requirements for Web Development

by Sophia

WHAT'S COVERED

In this lesson, you will learn about the minimum tools and resources needed to successfully start and complete a web development project. You will learn about the different types of software developers use to write, test, and run our code. You will be introduced to the differences between client-side and server-side coding languages and learn about their uses.

Specifically, this lesson will cover the following:

BEFORE YOU START

This lesson will provide a general overview of the various resources and tools available to web developers. You will be given guidance on recommended tools to add to your "toolbox" in the next Challenge.

1. Minimum Requirements for Web Development

Today, the minimum requirements to develop software and websites for the web include a computer with access to the Internet, a basic text editor application, and a web browser application. However, to be more effective, developers will want to be familiar with a variety of tools that support your efforts and improve your efficiency and productivity. Being comfortable with a quality tool set will allow a developer to focus more time on producing a quality product and less time on managing files, resources, and the project overall.

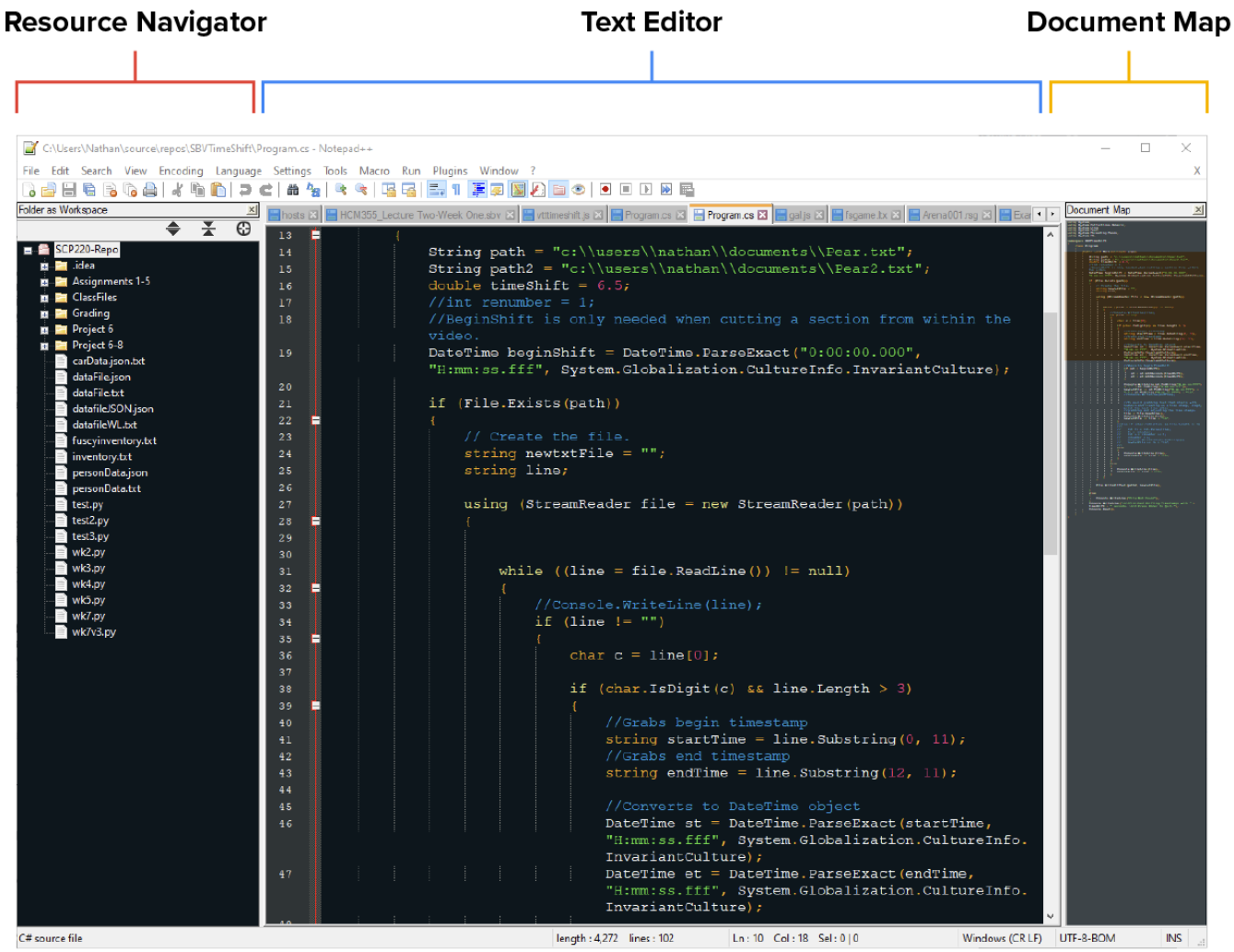
2. Ideal Tools for Web Development

Why use a rock to drive a nail when you can use a hammer? Programmers can use notepad.exe on a Windows device, or TextEdit on a Mac, to write a website. However, there are specialized applications available that make it much easier for developers and can actually help us to complete our work faster and with less error. In fact, there are applications designed to provide aid in specific types of development like database development vs. Windows development vs web development.

2a. Code Editor

Code editors are applications designed for a wide range of coding and programming languages. Code editors often include a basic **text editor**, but will also colorize the **syntax** to help the developer see errors and navigate the code structure. Code editors also come with basic built-in tools like a resource navigator and code map, to help developers keep track of their project files and resources. This is particularly useful as your projects become larger and more complex.

Some code editors, like Notepad++, shown in the screenshot below, also have **plugins** available to add additional features, support for additional languages, as well as tools for debugging and running, and testing the code. These added features help you personalize the editor to your preferences.



Other code editors, such as Brackets, include other features like the live editor which can render the output of certain code files in real time, even as you type it!

HINT

While a code editor is a great tool for beginning web developers, you are encouraged to use an Integrated Development Environment as you practice the activities in this course. This will help ensure you are prepared for the final project. You will learn more about Integrated Development Environments in the next section.

TERMS TO KNOW

Code Editor

Computer application designed specifically for editing programming, scripting, and markup text-based code files.

Text Editor

Computer software designed for editing nonspecific text-based files.

Syntax

The rules of arranging keywords, identifiers, and characters into meaningful statements within a language.

Plugin
A software component that adds a specific function or feature to an existing program.

2b. Integrated Development Environment (IDE)

IDEs, or Integrated Development Environments, are the next step up from code editors and provide developers with more sophisticated tools to aid in the development process. IDEs include the colorized text editor, resource navigator, and document map available in most code editors, but they also include tools such as more advanced debugging tools, management tools for data sources and assets, as well as tools to facilitate remote collaboration. Web development IDEs also sometimes include a **WYSIWYG (What You See Is What You Get) editor** which allows a developer to interact with a webpage's content as if they were viewing the page from a browser. WYSIWYG editors are a great low-code or no-code choice for anyone wanting to manage a website's content without having to know much or any HTML or CSS.

IDEs can be intimidating for novice developers as there are a lot of features and tools that may not be as intuitive to use. Developers should take the time to explore different IDEs related to their field of development and stick to one that they are either comfortable with or one that offers the most flexibility and convenience.

Some good, flexible, IDE options for web development include:

- Visual Studio Code
- Adobe Dreamweaver
- Eclipse
- NetBeans

Keep in mind that the examples above are applications that get installed locally on your development system. This works fine if you only work from a single system. If you work remotely, travel often, or don't have administrative access to the system you are working on, there are other options.

TERMS TO KNOW

Integrated Development Environment (IDE)
Software application that contains a code editor, utilities, and tools designed to aid in the creation of software applications, services, and assets.

WYSIWYG (What You See Is What You Get) Editor
A type of software that allows users to see, create, and edit website content as it would appear in the final product.

2c. Web-Based IDEs

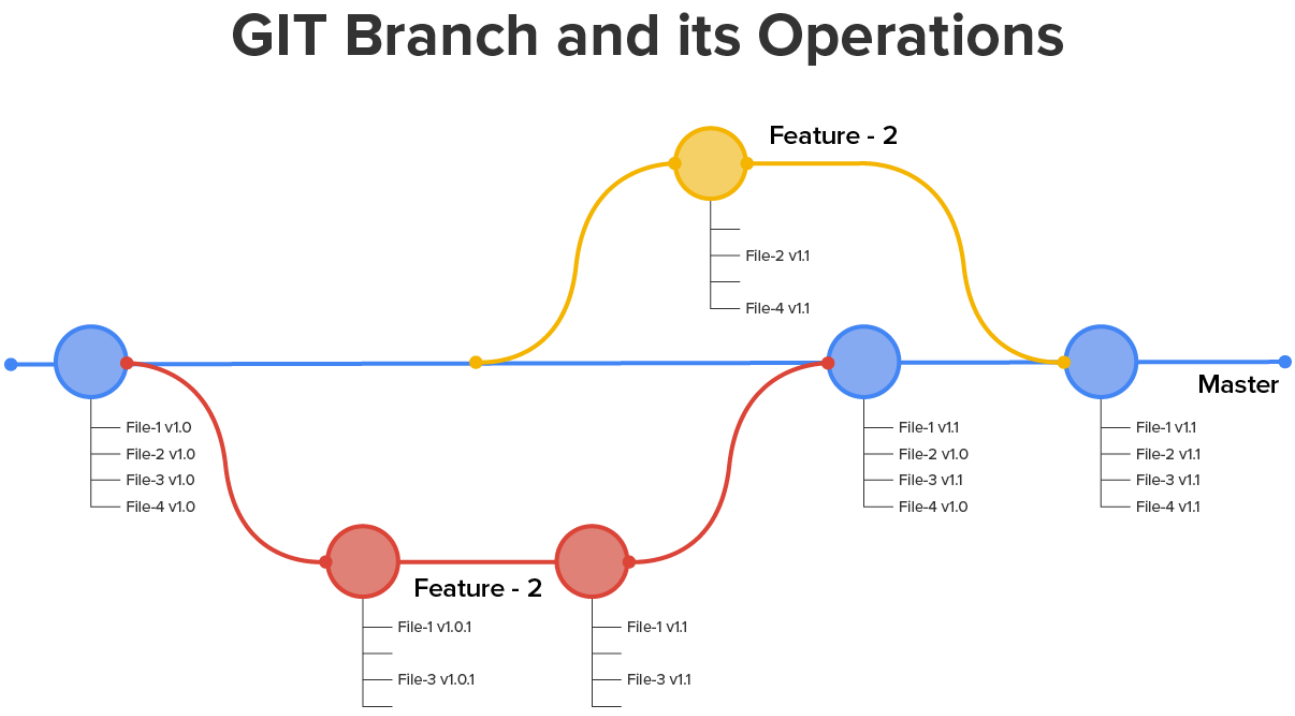
Web-based Integrated Development Environments are full IDEs that include online storage and are accessible through a basic web browser. Once registered, these sites allow us to create a project. The project is hosted on the site and allows for multiple files and resources to be created or uploaded. The site includes all of the features of a modern IDE and can test and run your code within the site itself.

You can build and run your code files in the browser, just as if you were running it on your local system. The benefits of using web-based IDEs are numerous and include:

- Your development process is completely uncoupled from the physical computer you are using.
- Collaboration and code sharing is inherent with online web-based platforms.
- Code can be compiled and code files and resources can be packaged for deployment.
- Most have integrated support for **Git version control** system and **forking** of projects.
- Some provide the ability to use **package managers** to install additional software and code packages.

KEY CONCEPT

Git is a version control system that keeps a detailed log of any and all changes made to code files. Not only does this provide the ability to revert back to previous versions of the file, but it also allows multiple developers to work on a single code file. When multiple developers work on the same project, each one will use Git to create a "branch" in the project for them to work on their assigned programming task. **Branching** the project means that the change logs from each branch will be tracked separately, but still as part of the project. When each developer is finished, they use Git to merge their branch back into the central project. To "fork" a project is to make a complete copy of a project folder. The new copy is not associated with the project's Git system. Forking, for example, is used to make a copy of an open-source project in order to make your own version.



Source: Digital Varys

TERMS TO KNOW

Git
A free and open-source version control system designed to provide version control over software to enable distributed collaboration, forking, branching, and change tracking and merging.

Version Control
The practice of tracking and managing changes to software code.

Forking
A Git operation that creates a complete copy of a project, one that is not associated with the original project.

Package Manager
Command-line interface (CLI) software used to locate, download, and install pre-build software packages that add additional functionality and capabilities to a coding language, a framework, or directly into a project.

Branching
A Git operation that causes changelogs to be kept separate to allow a developer to work separately and then merge their changes into the central project.

2d. Content Management System (CMS)

Another approach to site creation and content management is to utilize a **GUI-based application** called a **Content Management System (CMS)**. **Graphical User Interfaces (GUI)** provide a method of interacting with technology that uses visual elements such as images, icons, drop-down menus to control the system. CMSs are software tools designed to simplify the content management and design of an entire website. For example, creating a new page, applying the site's design theme, and integrating the new page into the site's navigation menus are now as easy as clicking a few buttons.

There are many popular CMS options for web development, each with its own strengths and weaknesses. Here are some of the most common examples, Wordpress, Joomla, Drupal, Squarespace, Wix, Shopify, etc. We will look into Wordpress in more details in the next tutorial.

BIG IDEA

With the use of a CMS, developers can focus more of their time on content creation. Another benefit of a CMS is that it enables non-developers to get involved in the process of content creation and site management, thus freeing non-developers from their dependency on a third-party developer.


TERMS TO KNOW

GUI-Based Application
A computer software application that provides graphic elements on the computer screen that allow users to interact with the computer.

Content Management System (CMS)
A GUI-based system of software designed to aid in the creation and management of website pages and content.

Graphical User Interface (GUI)
A method of interacting with technology that uses visual elements such as images, icons, drop-down menus that users directly interact with to control the system.

3. Coding Languages

Web developers will need to know several markup and scripting languages in order to complete most web development projects. These coding languages are generally grouped by where they operate, client side or server side.

IN CONTEXT

The term “client-side language” refers to code that executes on the client’s system. The term “server-side language” refers to code that executes on the server. Two similar terms that are sometimes used interchangeably are **front-end** languages and **back-end** languages. However, “front end” refers to whatever the user sees and interacts with as the client while “back end” refers to how client requests are processed and handled by the server.

Whenever you hear the term **full-stack developer**, this refers to an individual that can handle both front-end and back-end development.

TERMS TO KNOW

Front End
Refers to anything that the user sees and interacts with when they access a website.

Back End
Refers to the technology that resides and runs on the web server and directly or in-directly supports the operations of the website.

Full-Stack Developer
A developer with the skills and knowledge to develop both the front end of a website that the user interacts with as well as the back-end technology that supports the website.

3a. Client-Side Languages

The three client-side coding languages used are:

- HTML (HyperText Markup Language)
- CSS (Cascading Style Sheet)

- JS (JavaScript)

WATCH

View this video for a brief introduction to HTML and CSS.

Client-side scripting languages are executed on the client's systems. When a visitor requests a webpage from a server, the client-side code (HTML, CSS, and JS) is what gets transmitted back to the visitor's browser. This code and content is then rendered to the screen by the browser and any JS is executed within the browser.

HyperText Markup Language (HTML) is a markup language and is used to define the webpage, its content, and its overall structure using **tags**. Tags that surround a section of text instruct the browser how to render the text to the screen. HTML is used to provide structure to data such as creating a table for information or a numbered list of instructions as well as linking to and bringing in resources such as images and videos.

Cascading Style Sheets (CSS) is a **selector**-based language that integrates with HTML to provide a high degree of control over how all webpage content is rendered in a browser. Selectors are defined to specify what HTML tags or structures are to be stylized and a list of property-value pairs is provided to control specific aspects of the content. While much of CSS is dedicated to controlling how content looks, the latest version of CSS provides several additional features that control the functionality and behavior of webpage content.

JavaScript (JS) is a powerful scripting language that provides developers the ability to fully customize how a website behaves and operates. JS is primarily used to edit and update the webpage's content in response to user events, timed events, and even asynchronous server responses from the server. However, JS can even be used to create entire **Single Page Applications (SPA)**, web-based applications that operate completely within the web browser.

BIG IDEA

HTML is used to create the page and its content, CSS is used to stylize the content and control the layout of the content, and JS is used to provide interactivity and functionality to the site's elements and features.

TERMS TO KNOW

Client-Side Scripting
Any computer language that is executed on a client system.

HyperText Markup Language (HTML)
The standard language for creating structured documents designed to be displayed within a web browser.

Tag
The building blocks of HTML. Tags are used to surround text and other tags to control layout, organization, and to apply styles.

Cascading Style Sheets (CSS)
The standard, rule-based style language for describing the presentation of an HTML document.

Selector
A component of a CSS that determines which element in the HTML should receive a set of style properties.

JavaScript (JS)
The computer language that is used to manipulate HTML and CSS elements of a webpage that executes on the client's system.

Single Page Application (SPA)
An architectural style of building a web application that rewrites the content of a single webpage to present the user with an interface.

3b. Server-Side Languages

Server-side languages are scripting languages that are stored and executed on the server. These scripts are used to perform additional operations such as accessing a database, generating and sending an email, or even generating an entire webpage. Other uses for server-side scripting are to offload processing burdens and provide a more light-weight interface for the user's system.

While there is a wide range of choices available, some of the more common server-side scripting languages are:

- PHP: A popular server-side scripting language framework
- Python: A general purpose language and framework utilized most often for server-side scripting
- ASP.NET/C#: A server-side scripting language and framework designed for Windows systems
- GoLang: A modern server-side scripting language focused on creating distributed web services for mobile site and application development
- Ruby: A server-side scripting language focused on providing simplicity and productivity in mind
- JavaScript with NodeJS: NodeJs is itself a powerful framework that unleashes the power of JavaScript from the browser; also provides a framework to allow JavaScript to be used as a server-side scripting language
- Java: A platform-independent, general-purpose language that can be easily implemented as a server-side scripting language

Unlike client-side coding languages wherein the developer does not have much of a choice as to what language they want to learn and use, server-side scripting is quite the opposite. While certain scenarios might dictate which server-side language would be better, developers have more choice in which server-side scripting languages they want to work with.

TERM TO KNOW

Server-Side Languages
Any computer language that executes on a server.

4. Developer Information Resources

Software developers, regardless of what area of development we work in, need to have some level of self-sufficiency and need to be self-resourceful. It is important for developers to be aware of the types of resources available to them. Let's talk about the different types of resource sites and how they can be useful.

KEY CONCEPT

Some sites include all three of the following resource types.

1. Reference Sites: Each coding language has a site dedicated to that language and often includes two key sections, the language reference and the library reference. The language reference focuses on how to use the coding language. This includes proper syntax, keywords, operators, shorthands, and procedures. The language reference is focused on detailing the use and operations of the pre-built repository of code and objects that come with each language. For example, each language has some sort of 'math' library module that contains all of the common mathematical operations.

EXAMPLE

- php.net
- python.org
- go.dev

2. Tutorial Sites: Tutorial sites are websites dedicated to helping individuals learn new skills. They are often organized into sequenced chapters that explain each concept, show demonstrations, and even provide interactive examples. These types of sites are excellent for those who are learning new languages or concepts, particularly those who are new to development and struggle with the terminology used in typical reference sites.

EXAMPLE

- w3schools.com
- tutorialspoint.com

3. Developer Forums: Developer forums are sites that include discussion forums wherein users can post a question and others in the developer community will respond. These sites are not for questions like, "How do I print text to the console" but more for scenario and situational questions like, "Here is my code... I am trying to x, but am getting errors and I cannot, for the life of me, see why. Any advice?" The other benefit is that you often don't have to ask the question yourself since the chances are someone else has already asked a similar question and received multiple responses, one of which. that contains the best response, is typically marked with a big green checkmark or the word "Answer".

EXAMPLE

- Stackoverflow.com

WATCH

View this video for more information on creating a toolbox for the course.

SUMMARY

In this lesson, you were introduced to the **Minimum Requirements for Web Development** as well as some of the **Ideal Tools for Web Development** to include **Code Editors**, **IDE (Integrated Development Environment)**, **Web-based IDEs**, and **CMS (Content Management Systems)**. You also learned about the web **coding languages** and the difference between **client-side** and **server-side** programming. Lastly, we finished with discussing the **developer information resources** that all good developers should be aware of and comfortable with using.

Source: THIS TUTORIAL WAS AUTHORED BY SOPHIA LEARNING. PLEASE SEE OUR TERMS OF USE.

TERMS TO KNOW

Back End
Refers to the technology that resides and runs on the web server and directly or in-directly supports the operations of the website.

Branching
A Git operation that causes changelogs to be kept separate to allow a developer to work separately and then merge their changes into the central project.

Cascading Style Sheets (CSS)
The standard, rule-based style language for describing the presentation of an HTML document.

Client-Side Scripting
Any computer language that is executed on a client system.

Code Editor
Computer application designed specifically for editing programming, scripting, and markup text-based code files.

Content Management System (CMS)
A GUI-based system of software designed to aid in the creation and management of website pages and content.

Forking
A Git operation that creates a complete copy of a project, one that is not associated with the original project.

Front End
Refers to anything that the user sees and interacts with when they access a website.

Full-Stack Developer

A developer with the skills and knowledge to develop both the front end of a website that the user interacts with as well as the back-end technology that supports the website.

GUI-Based Application

A computer software application that provides graphic elements on the computer screen that allow users to interact with the computer.

Git

A free and open-source version control system designed to provide version control over software to enable distributed collaboration, forking, branching, and change tracking and merging.

Graphical User Interface (GUI)

A method of interacting with technology that uses visual elements such as images, icons, drop-down menus that users directly interact with to control the system.

HyperText Markup Language (HTML)

The standard language for creating structured documents designed to be displayed within a web browser.

Integrated Development Environment (IDE)

Software application that contains a code editor, utilities, and tools designed to aid in the creation of software applications, services, and assets.

JavaScript (JS)

The computer language that is used to manipulate HTML and CSS elements of a webpage that executes on the client's system.

Package Manager

Command-line interface (CLI) software used to locate, download and install pre-build software packages that add additional functionality and capabilities to a coding language, a framework, or directly into a project.

Plugin

A software component that adds a specific function or feature to an existing program.

Selector

A component of a CSS that determines which element in the HTML that should receive a set of style properties.

Server-Side Languages

Any computer language that executes on a server.

Single Page Application (SPA)

An architectural style of building a web application that rewrites the content of a single webpage to present the user with an interface.

Syntax

The rules of arranging keywords, identifiers, and characters into meaningful statements within a language.

Tag

The building blocks of HTML. Tags are used to surround text and other tags to control layout, organization, and to apply styles.

Text Editor

Computer software designed for editing non-specific text-based files.

Version Control

The practice of tracking and managing changes to software code.

WYSIWYG (What You See Is What You Get) Editor

A type of software that allows users to see, create, and edit website content as it would appear in the final product.