|  |
| --- |
| The World Wide Web (WWW or simply the Web) is certainly what most people think of when they see the word “Internet.” But the WWW is only a subset of the Internet, as illustrated in Figure 1.    **Short History of the Internet**  The Internet is not alone in providing instantaneous digital communication. Earlier technologies like radio, telegraph, and the telephone provided the same speed of communication, albeit in an analog form.  Telephone networks in particular provide a good starting place to learn about modern digital communications. In the telephone networks of old, calls were routed through operators who physically connected caller and receiver by connecting a wire to a switchboard to complete a circuit. These operators were around in some areas for almost a century before being replaced with automatic mechanical switches, which did the same job: physically connect caller and receiver.    One of the weaknesses of having a physical connection is that you must establish a link and maintain a dedicated circuit for the duration of the call. This type of network connection is sometimes referred to as *circuit switching* and is shown in Figure 2.  In the 1960s, as researchers explored digital communications and began to construct the first networks, the research network (Advanced Research Projects Agency Network) ARPANET  was created. ARPANET did not use circuit switching but instead used an alternative communications method called packet switching. A packet-switched network does not require a continuous connection. Instead, it splits the messages into smaller chunks called packets and routes them to the appropriate place based on the destination address. The packets can take different routes to the destination, as shown in Figure 3.  Figure 3 Internet network as an example of packet switching  This early ARPANET network was funded and controlled by the United States government and was used exclusively for academic and scientific purposes. The early network started small with just a handful of connected university campuses and research institutions and companies in 1969 and grew to a few hundred by the early 1980s.  To promote the growth and unification of the disparate networks, a suite of **protocols**was invented to unify the networks. A protocol is a name given to a formal set of publicly available rules that manage data exchange between two points. Communications protocols allow any two computers to talk to one another, so long as they implement the protocol. By 1981 protocols for the Internet were published and ready for use. New networks built in the United States began to adopt the **TCP/IP (Transmission** **Control Protocol/Internet Protocol)**communication model while older networks were transitioned over to it. Any organization, private or public, could potentially connect to this new network so long as they adopted the TCP/IP protocol. On January 1, 1983, TCP/IP was adopted across all of ARPANET, marking the end of the research network that spawned the Internet. Over the next two decades, TCP/IP networking was adopted across the globe.  ª **Transmission Control Protocol (TCP):**  a connection-oriented communications protocol that facilitates the exchange of messages between computing devices in a network.  ª **Internet Protocol (IP):**is the method or protocol by which data is sent from one computer to another on the Internet.    **The Birth of the Web**  The next decade saw an explosion in the numbers of users, but the Internet of the late 1980s and the very early 1990s did not resemble the Internet we know today. During these early years, email and text-based systems were the extents of the Internet experience.  This transition from the old terminal and text-only Internet of the 1980s to the Internet of today is of course due to the invention and massive growth of the World Wide Web. This invention is usually attributed to the British Tim Berners-Lee (now Sir Tim Berners-Lee), who, along with the Belgian Robert  Cailliau, published a proposal in 1990 for a hypertext system while both were working at CERN in Switzerland. Shortly thereafter Berners-Lee developed the main features of the web.  This early web incorporated the following essential elements that are still the core features of the web today:  ª A Uniform Resource Locator (URL) a unique identifier used to locate a resource on the internet.  ª The Hypertext Transfer Protocol (HTTP) a protocol used to transfer data over the web.  ª A software program (later called web server software) that can respond to HTTP requests.  ª Hypertext Markup Language (HTML) a language used to create webpages.  ª A program (later called a browser) that can make HTTP requests from URLs and that can display the HTML it receives.    **How the web works**  Before you can understand web coding and development, you need to take a step back and understand a bit about how the web itself works. In particular, you need to know what happens behind the scenes when you click a link or type a web page address into your browser.    Here’s a high-level blow-by-blow of what happens:  1.      You tell the web browser the web page you want to visit.  You do that either by clicking a link to the page or by typing the location known as the ***uniform resource locator*or *URL*** into the browser’s address bar (see Figure 1-1).).  2.      The browser decodes the URL.  Decoding the URL means two things: First, it checks the prefix of the URL to see what type of resource you’re requesting; this is usually http:// or https://, both of which indicate that the resource is a web page. Second, it gets the URL’s domain name — the something.com or whatever.org part — and asks the *domain name system*(DNS) to translate this into a unique location — called the IP (Internet Protocol) address — for the webserver that hosts the page (see Figure 1-2).  3.      The browser contacts the web server and requests the web page.  With the web server’s unique IP address in hand, the web browser sets up a communications channel with the server and then uses that channel to send along a request for the web page (see Figure 1-3).  4.      The web server decodes the page request.  Decoding the page request involves a number of steps. First, if the web server is shared between multiple user accounts, the server begins by locating the user account that owns the requested page. The server then uses the page address to find the directory that holds the page and the file in which the page code is stored (see Figure 1-4).  5.      The web server sends the web page file to the web browser (see Figure 1-5).  6.      The web browser decodes the web page file.  Decoding the page file means looking for text to display, instructions on how to display that text, and other resources required by the page, such as images and fonts (see Figure 1-6).  7.      If the web page requires more resources, the web browser asks the server to pass along those resources (see Figure 1-7).  8.      For each of the requested resources, the webserver locates the associated file and sends it to the browser (see Figure 1-8).  9.      The web browser gathers up all the text, images, and other resources and displays the page in its entire digital splendor in the browser’s content window (see Figure 1-9).  Another way to look at this process is to think of the web as a giant mall or shopping center, where each website is a storefront in that mall. When you request a web page from a particular site, the browser takes you into that site’s store and asks the clerk for the web page. The clerk goes into the back of the store, locates the page, and hands it to the browser. The browser checks the page and asks for any other needed files, which the clerk retrieves from the back. This process is repeated until the browser has everything it needs, and it then puts all the page pieces together for you, right there in the front of the store. This metaphor might seem a bit silly, but it serves to introduce yet another metaphor, which itself illustrates one of the most important concepts in web development.  In the same way that our website store has a front and a back, so, too, is web development separated into a front end and a back end:  ª **Front end:** That part of the web page that the web browser displays in the browser window. That is, it’s the page stuff you see and interact with.  ª **Back end:** That part of the web page that resides on the webserver. That is, it’s the page stuff that the server gathers based on the requests it receives from the browser.  As a consumer of web pages, you only ever deal with the front end, and even then you only passively engage with the page by reading its content, looking at its images, or clicking its links or buttons.  However, as a maker of web pages — that is, as a web developer — your job entails dealing with both the front end and the back end. Moreover, that job includes coding what others see on the front end, coding how the server gathers its data on the back end and coding the intermediate tasks that tie the two together.    **Static Websites versus Dynamic Websites**  In the earliest days of the web, a ***webmaster*** (the term popular in the 1990s for the person who was responsible for creating and supporting a website) would publish web pages and periodically update them. Users could read the pages but could not provide feedback. The early days of the web included many encyclopedic, collection style sites with lots of content to read (and animated icons to watch).  In those early days, the skills needed to create a website were pretty basic: one needed knowledge of the HTML and perhaps familiarity with editing and creating images. This type of website is commonly referred to as ***a static website***, in that it consists only of HTML pages that look identical for all users at all times.  The image below illustrates a simplified representation of the interaction between a user and a static website.  Within a few years of the invention of the web, sites began to get more complicated as more and more sites began to use programs running on web servers to generate content dynamically. These server-based programs would read content from databases, interface with existing enterprise computer systems, communicate with financial institutions, and then output HTML that would be sent back to the users’ browsers. This type of website is called here in this text a ***dynamic website*** because the page content is being created at run time by a program created by a programmer; this page content can vary from user to user.    The image below illustrates a very simplified representation of the interaction between a user and a dynamic website. |

[◀︎ Preliminary Activity for Week 1](https://bcpeducollege.elearningcommons.com/mod/assign/view.php?id=423357&forceview=1)

Top of Form

Jump to...                     Jump to...                     Announcements                     Additional Learning Materials                     Preliminary Activity for Week 1                     Analysis, Application, and Exploration For Week 1                     Generalization for Week 1                     Evaluation for Week 1                     Assignment for Week 1                     Preliminary Activity for Week 2                     Lesson Proper for Week 2                     Analysis, Application, and Exploration for Week 2                     Generalization for Week 2                     Evaluation for Week 2                     Assignment for Week 2                     Preliminary Activity for Week 3                     Lesson Proper for Week 3                     Analysis, Application, and Exploration for Week 3                     Generalization for Week 3                     Evaluation for Week 3                     Assignment for Week 3

Bottom of Form

**Server-Side and Client-Side**

Often in web design, you’ll hear references to “**client-side**” or “**server-side**” applications. These terms are used to indicate which machine is doing the processing. Client-side applications run on the user’s machine (also referred to as the **frontend**), while server-side applications and functions use the processing power of the server computer (the **backend**).

The ***server*** is a computer agent that is normally active 24 hours a day, 7 days a week, listening for queries from any client who make a request. A ***client*** is a computer agent that makes requests and receives responses from the server, in the form of response codes, images, text files, and other data.

\*       **The Client**

Client machines are the desktops, laptops, smartphones, and tablets you see everywhere in daily life. These machines have a broad range of specifications regarding operating system, processing speed, screen size, available memory, and storage. In the most familiar scenario, client requests for web pages come through a web browser. But a client can be more than just a web browser. When your word processor’s help system accesses online resources, it is a client, as is an iOS game that communicates with a game server using HTTP.

**Frontend Development**

Frontend development includes the following web technologies:

ª  HyperText Markup Language (HTML) is a markup language used to define the structure and contents of web pages.

ª  Cascading Style Sheets (CSS) are used for styling web pages.

ª  JavaScriptis a programming language used for making web pages interactive.

\*       **The Server**

The server in this model is the central repository, the command center, and the central hub of the client-server model. It hosts web applications, stores user and program data, and performs security authorization tasks. The essential characteristic of a server is that it is listening for requests, and upon getting one, responds with a message. The exchange of information between the client and server is summarized by the request-response loop.

**Backend Development**

The following technologies are typically in the domain of the backend developer:

ª  Server software (Apache, Microsoft IIS)

ª  Web application languages (PHP, Ruby, Python, JSP, ASP.NET)

ª  Database software (MySQL, Oracle, SQL Server)

**TERMS**

\*       **Hypertext Preprocessor (PHP)** is a server scripting language, and a powerful tool for making dynamic and interactive Web pages.

\*       **Ruby**is an interpreted, high-level, programming language most commonly used for Web Development.

\*       **Python** is an interpreted, high-level and general-purpose programming language. Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

\*       **MySQL**is an open-source relational database management system.

\*       **Microsoft SQL Server** is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications, which may run either on the same computer or on another computer across a network.

**Full-Stack Developers**

When looking for a job in web development, you will frequently see posts looking for “full-stack” developers. That means a person who is fluent in both frontend (HTML, CSS, JavaScript) and backend (server applications, databases) languages.

**Other Roles**

Not surprisingly, there are a myriad of other roles that contribute to the creation and maintenance of a site. Here are a few common roles that fall just outside the moniker “web design.”

ª  **Product manager**

Ä  The product manager of a website or application guides its design and development in a way that meets business goals. This member of the team must have a thorough understanding of the target market as well as the processes involved in the creation of the site itself. Product managers develop the overall strategy for the site from a marketing perspective including how and when it gets released.

ª  **Project manager**

Ä  The project manager coordinates the designers, developers, and everyone else who is working on the site. They manage things like timelines, development approaches, deliverables, and so on. The project manager works with the product manager and other product owners to make sure that the project gets done on time and on budget.

ª  **SEO specialist**

Ä  A website or application isn’t much good if nobody knows it exists, so it is crucial that a site be easily found by search engines. **Search Engine Optimization** (*SEO*) is a discipline focused on tweaking the site structure and code in a way that increases the chances it will be highly ranked in search results. There may be an SEO specialist on the in-house team, or a company may choose to hire an outside SEO firm. SEO is sometimes perceived as a dark art, but there are many ways to improve findability that are not underhanded. In fact, the number one technique for improving SEO is simply having good content with savvy HTML markup.

ª  **Multimedia producers**

Ä  One of the cool things about the web is that you can add multimedia elements to a site, including sound, video, animation, and even interactive games. Creating multimedia elements is generally best left to artists and technicians in those fields, although they may be part of the web team if video, animation, or interactivity are core to the site’s mission.

**Skills Every Web Designer Needs**

We have focused on quite a few technical skills that will be helpful in building websites. Skills that are just as critical to your success.

ª  **Excellent communication skills**

In your work, you will need to communicate in person, on the phone, in email, and in text messaging tools with clients, team members, and superiors. Be clear, proactive, and straightforward with what you have to say. Good communication requires not only that you express yourself clearly, but also that you be a good listener.

Make sure that you understand issues being discussed, and don’t be afraid to ask for clarification if you don’t.

ª  ***Flexibility***

*Be able to change direction quickly because not only does web technology change quickly, but you will no doubt be thrown curveballs in your day-to-day work as well. For example, you may arrive at work one day to find that the client has changed your priorities completely. You might find that they’ve cancelled your project entirely. You might be asked to learn new skills and shift positions in the team. Staying adaptable is the key to survival.*

ª  ***Critical thinking and good judgment***

*Problem-solving is central to all of the disciplines related to web design, so you need to be able to use critical thinking skills to come up with solutions and always employ basic common sense.*

ª  ***A good attitude***

Creating sites mean being part of a team, even if you work at home as a freelancer. Be mindful that the attitude with which you approach your work is contagious, so strive to be a positive and friendly team member.

**Choosing Your Text Editor**

The bread-and-water version of a text editor is the barebones program that came with your computer: Notepad if you run Windows, or TextEdit if you have a Mac. You can survive as a web developer using these programs. Although you can get by with the simple text editors that come with your computer, a dedicated code editor makes the task of writing HTML, CSS, and JavaScript much easier. Code editors understand the syntax of the code you write, so they can do things for you like color coding, error detection, and automatically finishing simple tasks like closing HTML tags. Some provide page previews so you can view the results of your code as you work.

These nutrients are the features and tools that are crucial to being an efficient and organized developer:

ª  **Syntax highlighting**

Ä  Syntax refers to the arrangement of characters and symbols that create correct programming code, and syntax highlighting is an editing feature that color-codes certain syntax elements for easier reading. For example, while regular text might appear black, all the HTML tags might be shown in blue and the CSS properties might appear red. The best text editors let you choose the syntax colors, either by offering prefab themes, or by letting you apply custom colors.

ª  **Line numbers**

Ä  It might seem like a small thing, but having a text editor that numbers each line, can be a major timesaver. When the web browser alerts you to an error in your code, it gives you an error message and, crucially, the line number of the error. This enables you to quickly locate the culprit and (fingers crossed) fix the problem pronto.

ª  **Code previews**

Ä  A good text editor will let you see a preview of how your code will look in a web browser. The preview might appear in the same window as your code, or in a separate window, and it should update automatically as you modify and save your code.

ª  **Code completion**

Ä  This is a handy the feature that, when you start typing something, displays a list of possible code items that complete your typing. You can then select the one you want and press Tab or Enter to add it to your code without having to type the whole thing.

ª  **Text processing**

Ä  The best text editors offer a selection of text processing features, such as automatic indentation of code blocks, converting tabs to spaces and vice versa, shifting chunks of code right or left, removing unneeded spaces at the end of lines, hiding blocks of code, and more.

Here are just a few of the better-known code editors for web products that are worth exploring:

ª  **Sublime Text:** Available for both Windows and Mac. $80, but a free trial is available. [www.sublimetext.com](https://bcpeducollege.elearningcommons.com/mod/page/www.sublimetext.com%20)

The Sublime Text editor is definitely one of our favorites! It offers a free version for testing, but all continual users are required to pay $80 to keep it active. While $80 might sound steep for a text editor, it’s important to note that the licenses are per-user, rather than per-machine, so you can enjoy Sublime Text on as many computers and operating systems as you wish with your license.

ª  **Atom:** Available for Windows and Mac. Free! [http://atom.io](http://atom.io/)

With Atom, you gain access to an open-source text editor with developers in mind. In fact, the creators of Atom state that they made it just for developers. Also, there’s a community of developers who contribute themes and plugins, much like WordPress or some other open-source tools.

ª  **Brackets:** Available for Windows and Mac. Also, free! <http://brackets.io/>

The Brackets text editor comes from the folks at Adobe, in an attempt to offer a more modern, open source solution for developers creating websites. This is a free text editor, with some appealing visual tools for previewing your work and allowing for frontend developers to examine the changes. Writing code is the main focus in Brackets, and it’s done with the help of inline editors, live previews, and nicely organized files.

ª  **CoffeeCup –**Available for Windows.<https://www.coffeecup.com/html-editor/>

CoffeeCup’s HTML Editor provides extremely advanced text editing for coding and overall web design management. The editor has a free trial, but in order to continue, you must pay the $49 one-time fee. There’s also a fully free version, but the features are pretty watered down. We like CoffeeCup for creating HTML documents, but you might consider avoiding it for many other code languages. However, it makes sense to take advantage of CoffeeCup if you’re simply learning a language like HTML or PHP.

ª  **TextMate:**Available for Windows and Mac.<https://macromates.com/>

TextMate comes offered as a free download, but you can also decide to upgrade to the premium version for $59. Keep in mind that this payment only gets you one license, so you would have to pay for multiple seats if you have a whole team of people in need of the text editor.

ª  **Komodo Edit (Or IDE):**Available for Windows and Mac.<https://www.activestate.com/products/komodo-edit/>

The idea behind Komodo Edit is to offer something powerful, yet with a certain level of simplicity, so that even beginners should be able to grasp it. You can download Komodo Edit for Mac or Windows operating systems. It’s free and open source, allowing those who don’t need all the advanced features a text editor that gets smaller projects done.

ª  **Notepad++:** Available for Windows only. Another freebie. <https://notepadplus-plus.org/>

Debatably the most popular advanced text editor on the market, Notepad++ comes in a compact package with no fees and powerful editing components. It is given away for free on a General Public License, meaning that all developers and content creators are able to take advantage of the text editor right after a quick download. Notepad++ runs on Microsoft Windows, and it strives to use less computing power than the average text editor.

ª  **UltraEdit:**Available for Windows, Mac and Linux.

UltraEdit also serves as a viable solution for your text editing needs. It’s not free. In fact, you have to start by paying at least $99.95. That gets you the standard UltraEdit text editor, along with free upgrades for any future releases. You can also install the software on three machines, whether it’s Windows, Mac, or Linux computers.

ª  **CodeShare:**<https://codeshare.io/>

The CodeShare text editor takes a completely different approach to online code editing. It’s built for developers, and it has a focus on giving these developers the opportunity to share code in real-time and speak to each other through a video chat. So, essentially it’s a real-time code editor combined with a Skype-like communication service.

ª  **Visual Studio Code:** Available for Windows, Linux, and Mac. <https://visualstudio.microsoft.com/>

As one of the younger players in the game (launched in 2015) Visual Studio Code puts forth quite an effort for building a stable community and ensuring that users are getting the features they need. The hard work definitely shows since the plugin library has been growing quite a bit. It’s also an open-source project that you can download directly to macOS, Windows, or Linux for free.

***NOTE: To do the exercises in this book, all you’ll need is the text editor that came with your operating system and free image creation software. There is no need to purchase anything to follow along.***

**A variety of browsers**

One of the biggest challenges for web designers is that our sites may look and behave differently from browser to browser. For this reason, it is critical that we test our designs early and often on the widest range of browsers possible.

These are the browsers designers and developers keep around for testing:

ª  Chrome (google.com/chrome)

ª  Firefox (www.mozilla.org)

ª  MS Edge (www.microsoft.com/en-us/windows/microsoft-edge; Windows only)

ª  Internet Explorer 9–11 (www.microsoft.com; search “Internet Explorer”;

ª  Windows only)

ª  Safari (support.apple.com/downloads/#safari; Mac only)

ª  Opera (opera.com)

You will also need to test on a variety of smartphone browsers including iOS Safari, Android browsers, and third-party mobile browsers.

**1. Sublime Text**

The [Sublime Text](https://www.sublimetext.com/) editor is definitely one of our favorites! It offers a free version for testing, but all continual users are required to pay $99 to keep it active. While $99 might sound steep for a text editor, it’s important to note that the licenses are per-user, rather than per-machine, so you can enjoy Sublime Text on as many computers and operating systems as you wish with your license.

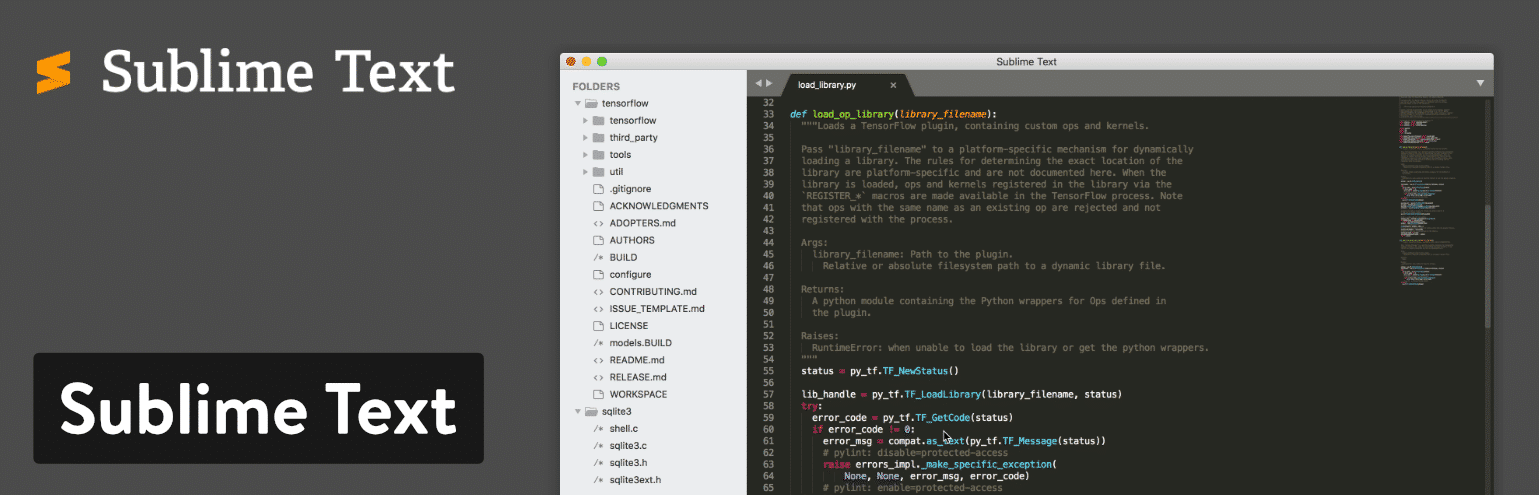
As for the features, Sublime Text has an advantage in that it’s **extremely lightweight (low resource usage)**, but still keeps around some of the more advanced features you would expect out of a top text editor. The primary benefit is that Sublime Text offers shortcuts and search tools for immediately finding certain functions and making changes to multiple lines at once. Jumping to specific symbols or words takes only a few seconds.

Top of Form

See how Kinsta stacks up against the competition.                      Select your provider                                              WP Engine                                                  SiteGround                                                  GoDaddy                                                  Bluehost                                                  Flywheel                                                  HostGator                                                  Cloudways                                                  AWS                                                  Digital Ocean                                                  DreamHost                                                  Other                         Compare

Bottom of Form

In addition, Sublime Text automatically creates an index of all functions and methods so that you can work with the shortcuts and customize it for locating bits of code while working.

[](https://www.sublimetext.com/)Sublime Text editor

Going along with the trend of shortcuts in Sublime Text, the text editor lets you type in a few keystrokes to move directly to menu items. So, if you wanted to sort something in your document, you wouldn’t be required to sift through the entire menu to find that functionality.

Sublime Text has a Python API, meaning that a wide variety of plugins can be integrated with the text editing solution. This includes the thousands of plugins that are consistently created by the Sublime Text community.

**Which Features Make This One of the Best Text Editor Tools?**

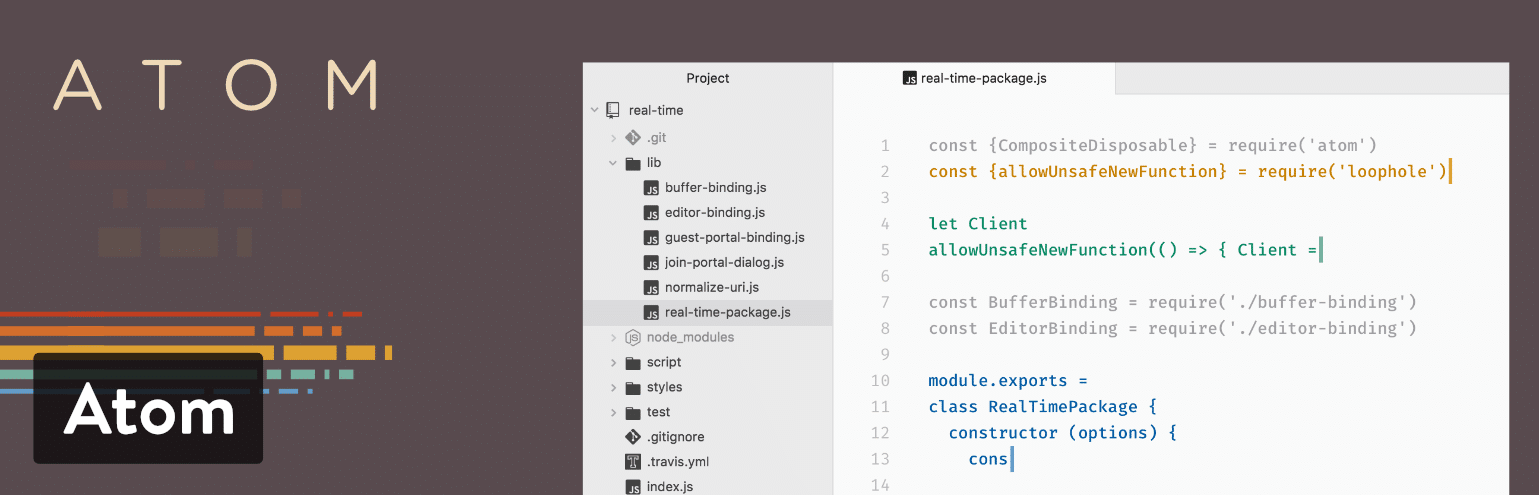
* Sublime Text lets you test out the text editor before committing to the investment. Although it’s a bummer you have to pay for this text editor, you at least get to try it out–and the pricing isn’t that expensive.
* The text editor runs on multiple platforms such as Mac, Windows, and Linux. It’s also cross-platform, so one license will work on all of your devices–no need to purchase more.
* Sublime Text provides split editing for managing and editing files next to each other. You can also open multiple windows and place them on different monitors.
* The Python API opens up opportunities to upgrade Sublime Text with plugins you, or other people, create.
* Sublime Text has extremely user-friendly and powerful shortcuts. From finding and modifying multiple lines to locating certain functions in the menu, Sublime Text should make shortcut lovers happy.
* You can also customize just about anything in Sublime Text. This is especially true when talking about shortcuts and menus. We recommend tweaking the settings to [open files in the same window (new tab)](https://woorkup.com/sublime-text-open-files-same-window/).
* Has some great community themes available. Check out the [Dracula Sublime theme](https://draculatheme.com/sublime/).

Sublime Text makes the most sense for heavy coders. These developers will enjoy the shortcut functionality and the high level of customization. The cross-platform abilities are also nice for launching the same text editor on all machines.

**2. Atom**

With [Atom](https://atom.io/), you gain access to an **open source text editor** with developers in mind. In fact, the creators of Atom state that they made it just for developers. Also, there’s a community of developers who contribute themes and plugins, much like WordPress or some other opensource tools. An experienced developer should have no problem working with Atom, since it offers clean collaboration tools, a sleek editor, and some great organization tools for keeping your projects in check.

All of your projects can be shared and edited in real-time, helping out teams that are far away from each other or simply those teams that want a more dedicated workspace. Furthermore, Atom has a GitHub package already included with the text editor. This way, your team can create everything from branches to stages in one interface.

[](https://atom.io/)Atom text editor

Another great thing about Atom is that it’s a cross-platform system, working on operating systems like Linux, OS X, and Windows. The smart autocompletion is one of my favorite features, and the multiple panes should make you feel right at home with several panes open to edit code between files. Much like an iPhone (or WordPress,) Atom has extras that you can install called packages. These are offered to expand the functionality of the basic text editor. You can also install themes to make your editor look prettier and easier on the eyes.

**Which Features Make This One of the Best Text Editor Tools?**

* Atom is an opensource text editor with a large community of developers. This means you get consistent updates and new themes and packages. Check out the [Dracula theme for Atom](https://draculatheme.com/atom/).
* It’s a cross-platform solution that works on the major operating systems.
* The editor is useful for coding alone, but its true strengths come into play when you need to collaborate with other people. All of the editing and creation can be done in real-time.
* Atom provides a GitHub package for integrating and doing things like pull requests and resolving merge conflicts.
* You can search for new packages and themes right from the text editor.
* It’s fairly easy to edit your code with smart autocompletion, a file system browser, and a [find and replace](https://kinsta.com/knowledgebase/wordpress-search-and-replace/) feature.
* Atom offers multiple panes for comparing and editing code side by side.

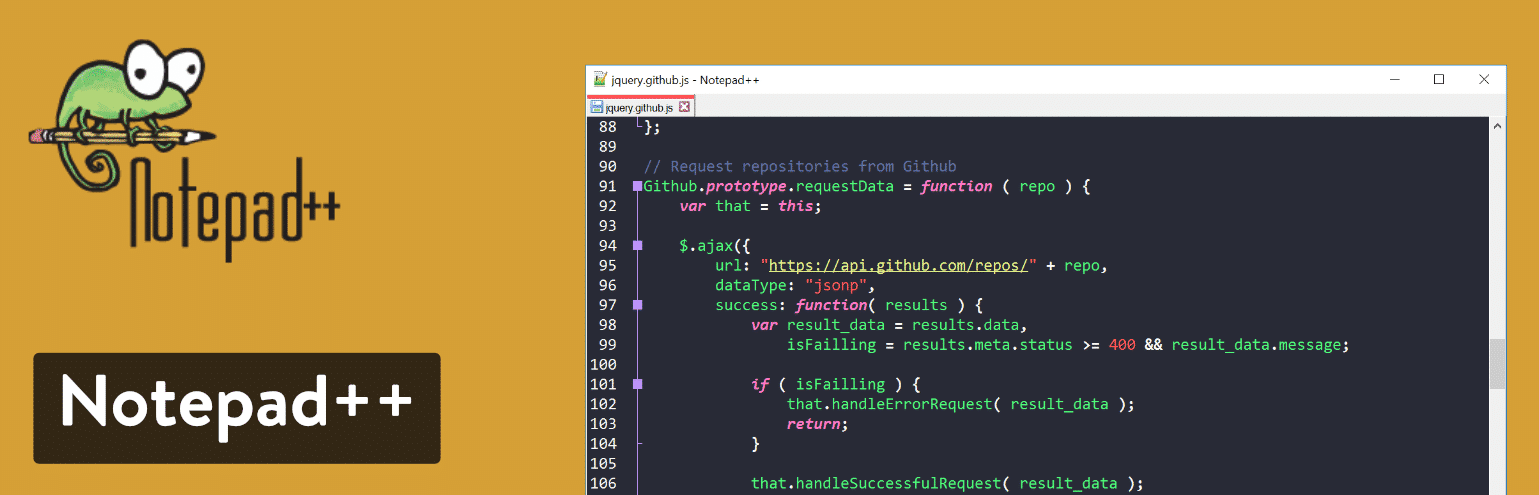
I would recommend Atom for those developers who want a collaboration tool in addition to the text editor. You can manage projects with other developers and see changes right in front of your eyes. Atom is also nice since it’s opensource–meaning you get it for free, the community is solid, and you can choose from packages and themes. In short, if you like working with [WordPress](https://kinsta.com/knowledgebase/what-is-wordpress/), Atom seems to present itself in a similar fashion.

**Your perfect website text editor needs the perfect host.**[**Try Kinsta for Free**](https://hubs.ly/H0pklC_0)**.**

**3. Notepad++**

Debatably the most popular advanced text editor on the market, [Notepad++](https://notepad-plus-plus.org/) comes in a compact package with **no fees** and powerful editing components. It is given away for free on a General Public License, meaning that all developers and content creators are able to take advantage of the text editor right after a quick download. Notepad++ **runs on Microsoft Windows**, and it strives to use less computing power than the average text editor.

One of the things that makes Notepad++ stand out is the fact that it’s already been translated into over 80 languages, allowing access to people all over the world. Furthermore, you’re able to translate Notepad++ into your native language if you don’t find your language on the list of translations.

[](https://notepad-plus-plus.org/)Notepad++ text editor

Writing code and manipulating text in Notepad++ comes easy, as it utilizes syntax highlighting and folding. There’s also a wonderful search and replace tool, along with an entirely customizable user interface. For instance, you might want a vertical tab or a document list–all are possible with the Notepad++ text editor.

**Which Features Make This One of the Best Text Editor Tools?**

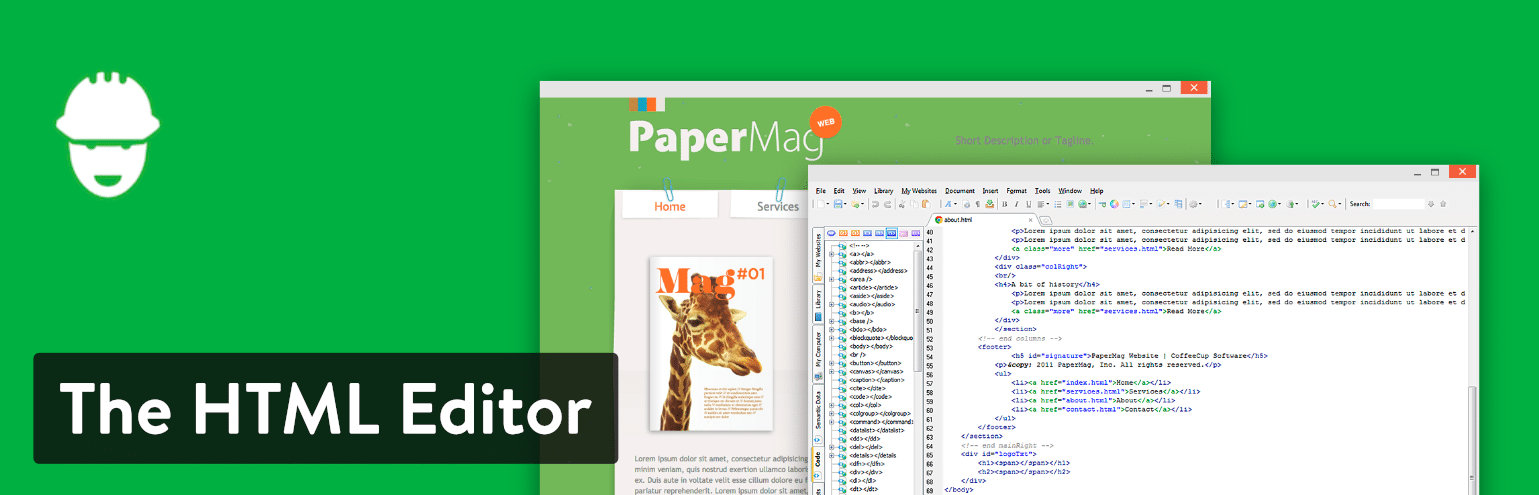
* Notepad++ is completely free for everyone to use.
* The text editor is already translated into dozens of languages, and it provides the documentation needed to translate into more languages.
* You receive a multi-view editor with syntax highlighting and folding.
* The customization tools are easy to understand and powerful enough for the most advanced developer.
* The auto-completion settings ensure that you can finish functions, parameters, and words without having to type them in over and over.
* It offers a multi-document interface for switching between tabs and managing multiple projects at once.
* Notepad++ provides a list of plugins to improve the functionality of the text editor or integrate with other programs.
* You can open a function list, which reveals an outline of all functions found in the current file. This also includes a search engine for quickly locating functions in large documents.

If you’re a beginner who needs a text editor, Notepad++ is far easier to understand than some of the other options on this list. Not only that, but you’ll enjoy the ability to locate words and replace them all, while also revealing certain bits of code with highlighting. As for advanced users, Notepad++ also does the trick, with its syntax highlighting for HTML, PHP, and JavaScript. The plugins also help out when trying to connect via FTP or integrate with other text editors.

**4. CoffeeCup – The HTML Editor**

CoffeeCup’s [HTML Editor](https://www.coffeecup.com/html-editor/) provides extremely advanced text editing for coding and overall web design management. The editor has a free trial, but in order to continue, you must pay the $49 one-time fee. There’s also a fully free version, but the features are pretty watered down.

We like CoffeeCup for creating HTML documents, but you might consider avoiding it for many other code languages. However, it makes sense to take advantage of CoffeeCup if you’re simply [learning a programming language](https://kinsta.com/blog/best-programming-language-to-learn/) like HTML or PHP.

[](https://www.coffeecup.com/html-editor/)CofeeCup The HTML Editor

As mentioned, creating and editing HTML is fairly simple with CoffeeCup. This takes your web design to the next level, especially with the tools for tag references and code completion. There are plenty of components to take advantage of that instantly update various elements across your entire website.

What’s more is that the CoffeeCup text editor comes with several responsive website themes. You may want to start from scratch if you’re trying to learn more about coding, but the themes expedite the process for when you need to generate client websites by a deadline.

**Which Features Make This One of the Best Text Editor Tools?**

* CoffeeCup has a free version and free trial. Also, the premium version is affordable.
* This is one of the best text editor options with themes. These aren’t completed websites, but it gets you close to some designs that might look appealing for some of your clients. Why reinvent the wheel when you can start further into the process?
* The FTP connection allows you to go live with your website with the click of a button. Publish your website anywhere you want by selecting a server or web host.
* CoffeeCup is one of the more visual text editors, with a standard split-screen view, but also a live preview to see what your code creates on the frontend.

CoffeeCup is a great choice for those interested in learning HTML, CSS, and PHP. The components library, combined with the theme selection, delivers a good starting point for those who are on a time crunch or don’t feel like beginning from scratch.

Top of Form

Bottom of Form

**Want to know how we increased our traffic over 1000%?**

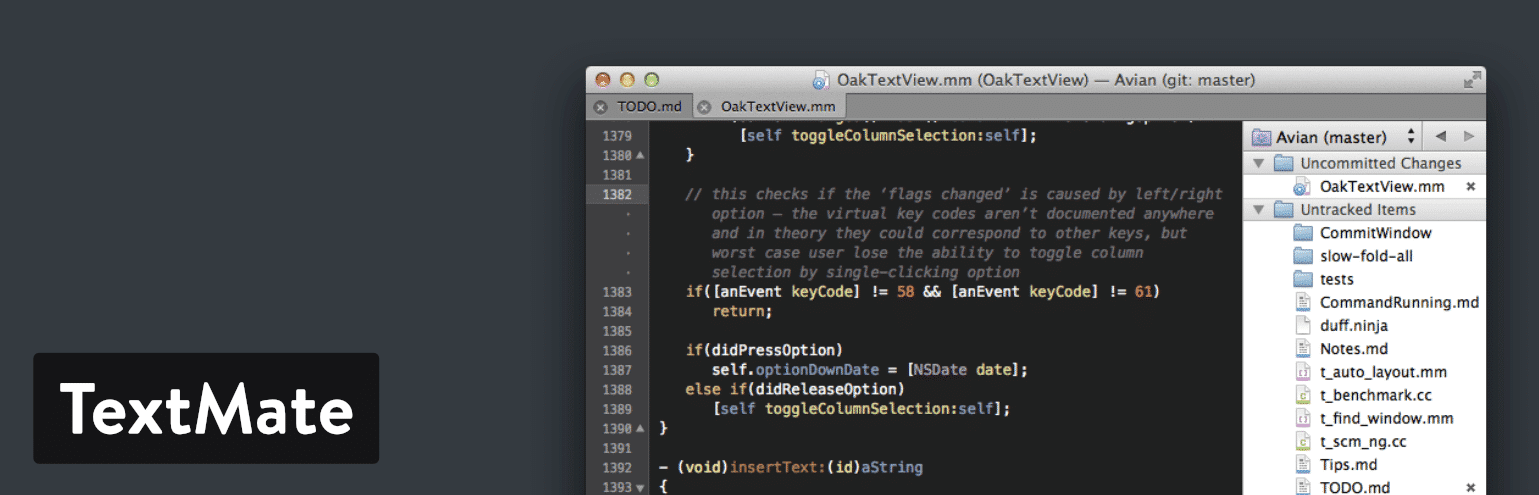
Join 20,000+ others who get our weekly newsletter with insider WordPress tips!

[**Subscribe Now**](https://kinsta.com/blog/best-text-editors/#newsletter)

**5. TextMate**

[TextMate](https://macromates.com/) comes offered as a free download, but you can also decide to upgrade to the premium version for $59. Keep in mind that this payment only gets you one license, so you would have to pay for multiple seats if you have a whole team of people in need of the text editor.

To get started, TextMate strictly works on the macOS. It seems like a simple editor at first, but it actually has quite a bit of functionality crammed into a small package. Some of the standards you would expect from a text editor include find search and replace tools, autocompletion, and board management. All [programming languages](https://kinsta.com/blog/scripting-languages/) are supported by TextMate, and it does have a tool for Xcode projects.

[](https://macromates.com/)TextMate text editor

TextMate lets you create multiple insertion points for editing and swapping out pieces of code in bulk. You also receive a list of all version modifications. So, the list includes file changes and allows you to jump back in time if needed.

**Which Features Make This One of the Best Text Editor Tools?**

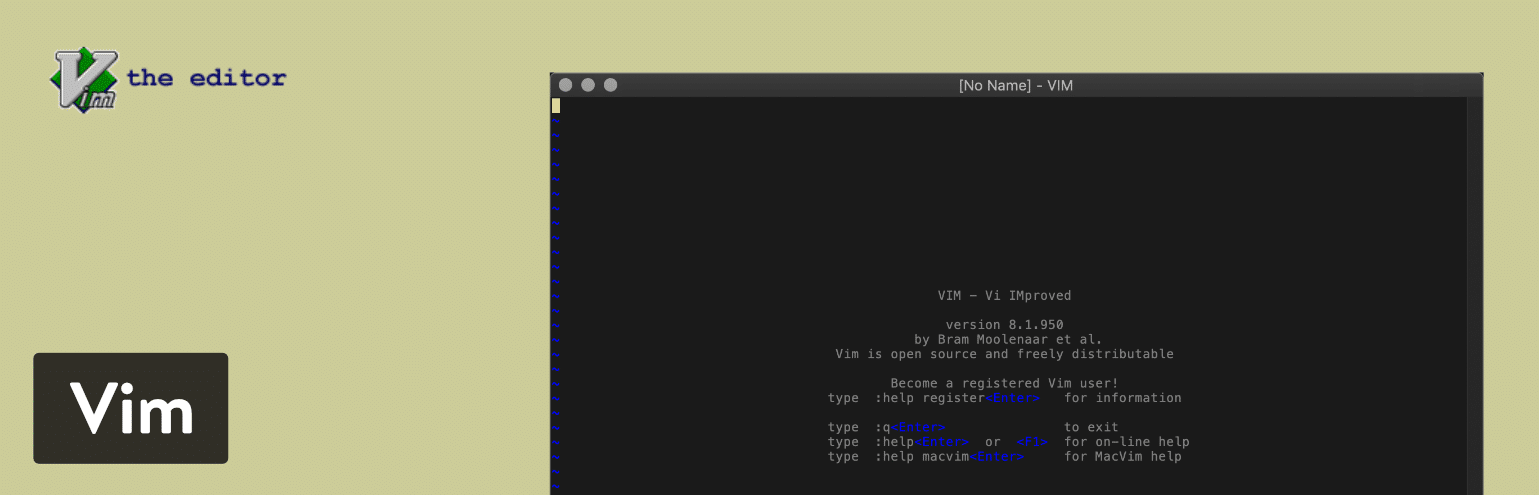
* TextMate comes with free and paid versions, both of which provide excellent features.
* It’s a lightweight text editor with a clean interface.
* TextMate offers bundles so that you can customize almost every element in the text editor. Want to change languages? That’s possible. Want to adjust your workflows? Go for it.
* You have the option to generate macros to speed up your production processes. All of your repetitive work is eliminated with these macros.
* An advanced file searching tool is included, along with multiple carets for swapping and adjusting multiple lines of code all at once.

If you’re using a Mac, and would like a text editor that supports all coding languages, TextMate is a solid choice for you. It also appears to be a nice solution for those who like bundles and macros.

**6. Vim**

With support for Windows, Linux, and Mac, the [Vim](https://www.vim.org/)text editor is a stable, reliable editor that integrates with many popular tools. It’s designed for use both from a command-line interface (CLI) and as a standalone application in a graphical user interface (GUI).

Vim was created in 1991. Back then, it was considered one of the most prominent text editors, which allowed developers to generate updates and scripts with a series of commands. This makes it one of the oldest text editors, and it’s quite impressive that Vim is still being used by developers around the world.

[](https://www.vim.org/)Vim text editor

It’s worth noting that Vim is one of the most advanced text editors on this list. However, that doesn’t mean it’s the most user-friendly. Vim definitely requires a learning curve and a willingness to immerse yourself in a whole new feature-set that might not be the same as some other text editors you are used to.

Having said that, you can expect a strong search and replace functionality, along with a large collection of plugins to extend the feature-set in this text editor. We also like the fact that Vim has a large online community to trade tips and learn about new tools to expand upon the base text editor.

You’ll probably notice just how archaic the Vim website looks. Well, the interface for the text editor isn’t much different. That being said, it’s still a high-powered solution for your text editing needs.

**Which Features Make This One of the Best Text Editor Tools?**

* Vim provides a system that’s friendly for people who enjoy using plugins for expanding upon the text editor.
* The Vim online community is a place for learning more about extensions as well as new scripts, tips, and tools.
* It is completely free and is an open source solution that receives updates on a regular basis.
* It’s also one of the oldest text editors on the market, and it continues to have some of the most powerful features and a strong following.
* You can integrate Vim with several tools that you are already using for your business or design needs.
* It doesn’t matter what programming languages you’re using, because Vim supports hundreds of languages, along with several different types of file formats.
* With the open source nature of Vim, and the vibrant community, you can vote for new features and really make an impact on the future development of the text editor.
* You can utilize the text editor on all of your machines, since it works on Linux, Windows, and Mac. There are popular projects like [MacVim](https://macvim-dev.github.io/macvim/)which provide a better GUI.

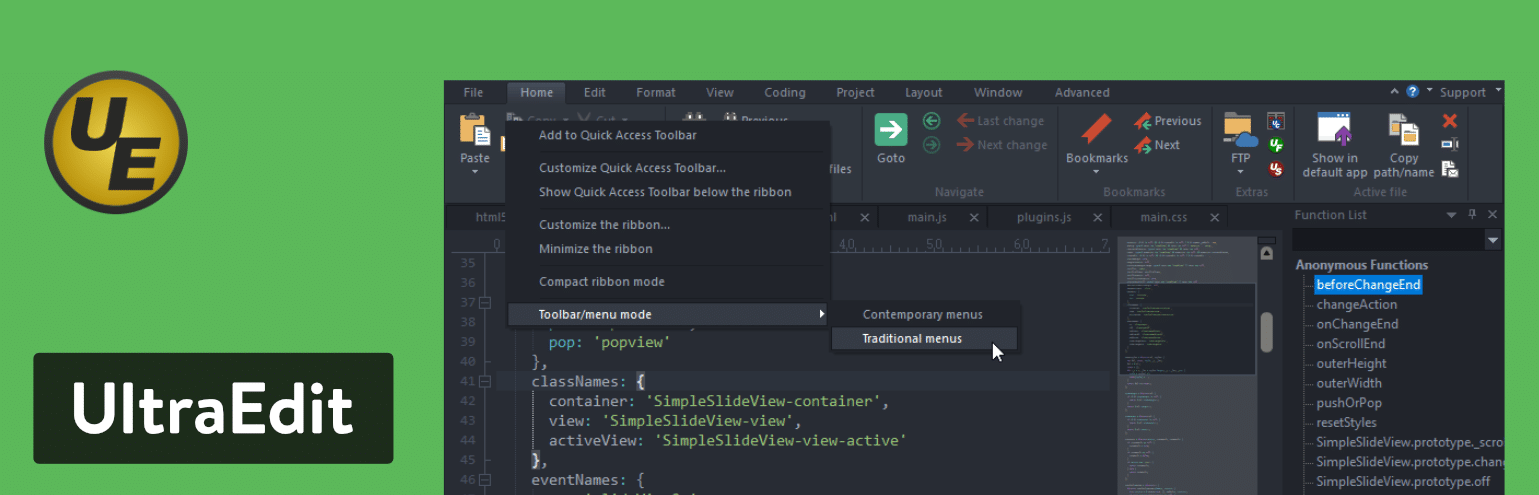
It’s tough to say which developers would enjoy Vim as a text editor. It’s an old system with an outdated interface. Yet, it still has the charm and powerful feature-set that the average developer needs. I would recommend it to more advanced developers who enjoy using open source software and being a part of a community–like the one you can find behind Vim.

[Nano](https://www.nano-editor.org/) is another popular alternative to Vim used by developers.

**7. UltraEdit**

[UltraEdit](https://www.ultraedit.com/) also serves as a viable solution for your text editing needs. It’s not free. In fact, you have to start by paying at least $99.95. That gets you the standard UltraEdit text editor, along with free upgrades for any future releases. You can also install the software on three machines, whether it’s Windows, Mac, or Linux computers.

After that, there are plenty of upgrades and additions to UltraEdit, such as an FTP program, compare tools, and finder options. You can decide whether or not any of these will help with your work process, but each one of them requires an additional payment. As for the core text editor, UltraEdit is known for its performance and customization abilities, along with some nice-looking themes so that you don’t always have to start from scratch. You can replace and find files, and most of the tasks like these are done in a rapid fashion.

[](https://www.ultraedit.com/)UltraEdit text editor

The multi-caret selection tool is sure to speed up development by allowing you to delete, paste, and cut anything you would like with the selection from your cursor. Live previews are also available. They’re displayed side-by-side as you make markdown changes right next to the preview. The feature set from UltraEdit is one of the most comprehensive on this list, so we can’t cover them all. However, we do particularly enjoy the customizable user interface, which allows you to lay out your workspace however you’d like.

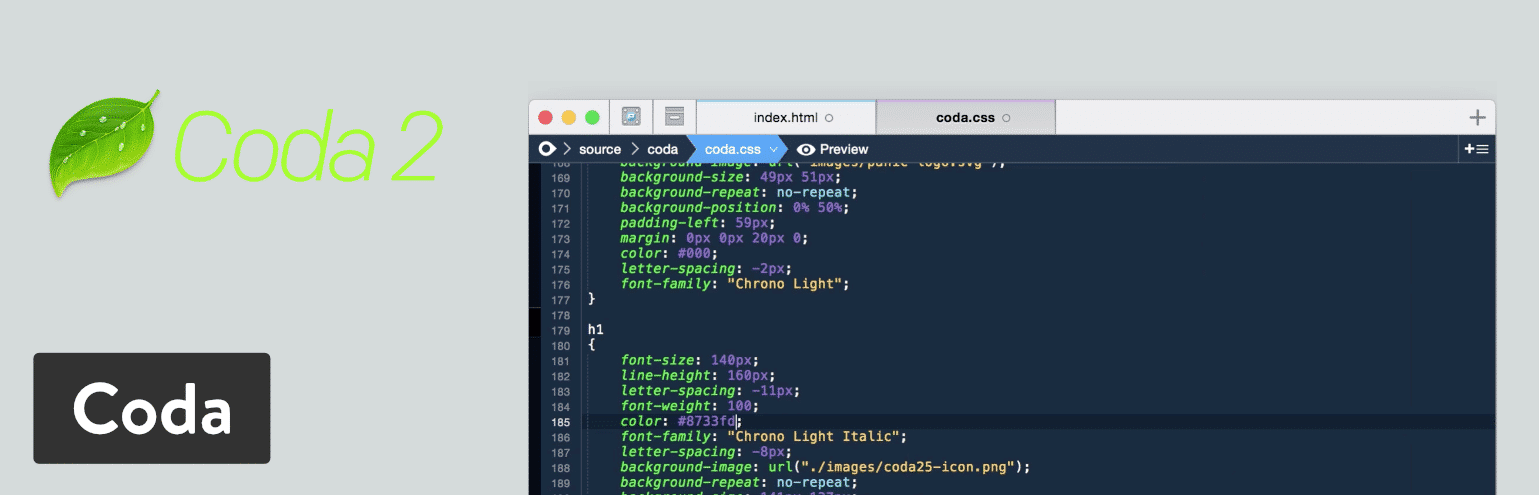
**Which Features Make This One of the Best Text Editor Tools?**

* UltraEdit is a premium solution that provides customer support, a decent community, and a wide variety of features that you wouldn’t always get with a free or open source text editor.
* It’s one of the fastest text editors you’ll find, especially in terms of replacing and finding files and selecting and editing multiple bits of code.
* The HTML live preview brings a visual aspect to your programming process. It sits right next to the text editor and renders an HTML preview for you to see right in the program.
* You’re able to edit extremely large files with UltraEdit. Many times, you’ll find that a less powerful or free text editor will crash when you try to customize these large files. That’s not the case with UltraEdit.
* This is a multiplatform text editor, and when you pay for one license you can use it on three different machines.
* All the tools are configurable in the text editor. You can even choose from some editor themes that serve as skins for the program.
* There are several smart templates that automatically complete some of the code as you type.

UltraEdit serves a few purposes. First of all, it allows developers and programmers an option for uploading and editing large files. You’ll also notice that it has incredible power and performance for locating files and searching for areas in those files. We also like it for corporate use, seeing as how you can install it on three different platforms and it’s one of the more secure and trusted solutions on the market. From regular text editing to web development and file comparing, UltraEdit provides an impressive set of key features that come at a reasonable price.

**8. Coda**

[Coda](https://panic.com/coda/) is for all the Mac lovers out there. It costs $99 for one copy, and the pricing gets progressively cheaper per copy if you purchase multiple copies at the same time. In addition, Coda is very reminiscent of many other Mac products, in that it has one of the most beautiful interfaces of every text editor on the market today. Mac users rave about it, and it’s not hard to see why. First of all, you receive a built-in terminal and an inspector to go along with an interface that lets you edit both remote and local files.

[](https://panic.com/coda/)Coda text editor

Coda is also an elder in this space, seeing as how it’s been around for over a decade. Coda2 is actually the text editor that you would be installing on your computer. This version offers some unique features such as local indexing and a CSS override for editing CSS on a live website. Another notable new feature is the publishing tool that tracks all of your files outside of the Coda app. You can then save and publish any of these remote files directly into the text editor.

On the surface, Coda is a basic text editor with features for syntax highlighting, code folding, and autocompletes. However, you will find some rather unique editing options, like something called a wildcard token that lets the user rapidly generate items like gradients and colors as you type. Although the text editor goes for $99, you receive a great value, along with excellent customer support and plenty of free e-books and resources dedicated just to the Coda editor.

**Your perfect website text editor needs the perfect host.**[**Try Kinsta for Free**](https://hubs.ly/H0pklC_0)**.**

**Which Features Make This One of the Best Text Editor Tools?**

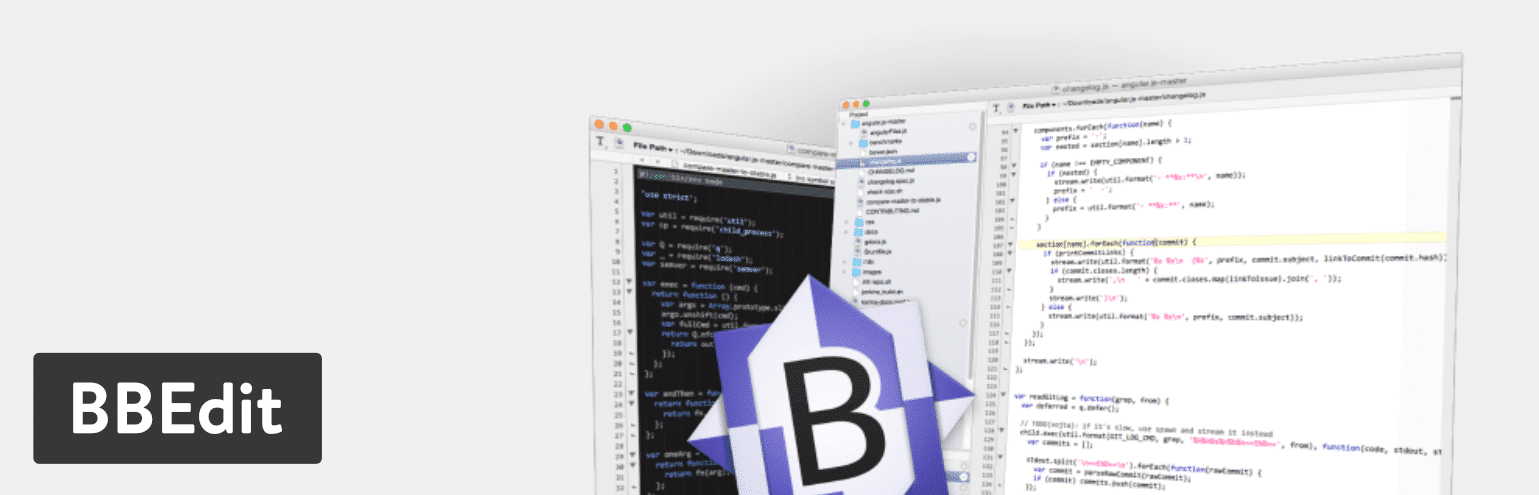
* Coda provides a one-click wildcard token for adding little changes throughout your document when working through the find and replace tool.
* It offers one of the cleanest, most modern interfaces out of all text editors.
* You receive excellent basic tools for working with a wide variety of languages and highlighting syntax.
* It is designed specifically for Mac users, so it’s great if you love Macs.
* File management is a breeze on Coda, seeing as how you can edit remotely through FTP or Amazon S3 servers, or even access your local files and publish them remotely.
* Much of your coding is completed through the text editor on your desktop. However, you can remotely edit your code, or even test out previews on your iPhone or iPad while editing in Coda.
* The new version includes some revolutionary tools such as CSS overriding, Panic syncing for passwords and private keys, and local indexing to autocomplete anything from classes to functions.

If you want the most beautiful interface on this list, this is the text editor for you. It’s also important to remember that this is one of the few text editors made just for Mac computers. So, if you’re a Mac user, it makes sense to go with Coda. You also might consider it if you would like to improve your coding previews or need to figure out a way to edit your files both remotely and locally.

**9. BBEdit**

[BBEdit](https://www.barebones.com/products/bbedit/) is an editor that spawned from the same technology used in TextWrangler. Now that TextWrangler has been decommissioned, all of its users are being pushed to use BBEdit instead.

Similar to Coda, BBEdit is also exclusively used on the Mac operating system. It offers highly advanced features, but also strives to have a bare-bones like appearance to go along with the acronym BB. BBEdit includes excellent features like Git integration and auto-completion. Syntax highlighting and quick lookups are also provided, along with editing windows that you can split up and place next to each other for easier editing. The current version of BBEdit costs $49.99 for an individual license. You can also upgrade to new versions for cheaper prices.

[](https://www.barebones.com/products/bbedit/)BBEdit text editor

With BBEdit, you have complete control over the text in your editor, it’s extremely easy to search and locate items due to the clean interface and smooth workflows. Text handling is one way to use BBEdit, seeing as how it offers excellent features such as canonization, hard wraps, and case changes. You’ll also find this text editor useful for web development, with its powerful previews in any browser and the ability to add special characters to any HTML.

The BBEdit text editor has impressive options for programming as well. For instance, many programmers enjoy the ability to auto-indent and check their syntax for certain scripts like Python and Ruby. We also like the fact that this text editor has code folding so that you can read your files easier by hiding larger sections of code. Overall, BBEdit is one of the best text editors, especially for those who are already using the Mac operating system.

**Which Features Make This One of the Best Text Editor Tools?**

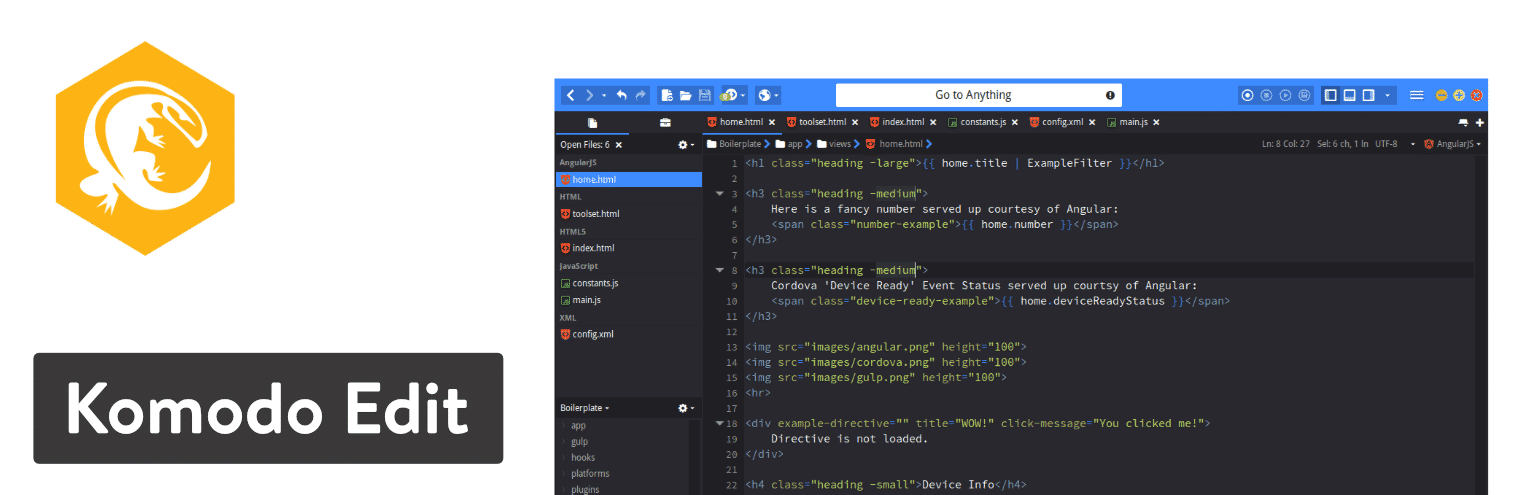
* BBEdit features code folding and text completion, both of which make for a cleaner interface and speedier coding.
* It works with a wide variety of languages and has syntax checking for multiple scripts.
* You receive a bare-bones interface that is meant to clear out all distractions while still keeping around the features that make for a strong text editor.
* This text editor is created just for Mac users, so you won’t have to worry about getting used to a new interface.
* Search and replace tools are available for adjusting individual items throughout multiple files. There are also several navigation functions as well as syntax coloring.
* You can add special characters to your code and insert attributes and tags that might assist you while writing your HTML.
* Under construction HTML and markdown can be previewed right next to the code being edited. Not only that, but all updates you make to the code are automatically changed in the preview before your eyes.
* It offers automated completion tools for suggesting things like clippings and symbols.
* You can take advantage of basic and more advanced text handling features such as the ability to exchange characters and words, the straightening of quotes, and full canonization.
* All of your projects have folder listings so that you can organize your work and then browse and edit your directory listings with speed and precision.
* All files can be accessed and manipulated through [SFTP and FTP](https://kinsta.com/knowledgebase/ftp-vs-sftp/).

BBEdit works wonders for those on the Mac operating system. It has the tools you need for web development, along with text handling. So, you might use it if you’re a writer or a programmer. It’s also worth noting that with the fairly low price and incredibly clean interface, it might make more sense to go with BBEdit over Coda.

**10. Komodo Edit (Or IDE)**

The idea behind [Komodo Edit](https://www.activestate.com/products/komodo-edit/) is to offer something powerful, yet with a certain level of simplicity, so that even beginners should be able to grasp it. You can download Komodo Edit for Mac or Windows operating systems. It’s free and open source, allowing those who don’t need all the advanced features a text editor that gets smaller projects done.

Also, if you require more advanced tools like code profiling or unit testing, the [Komodo IDE](https://www.activestate.com/products/komodo-ide/download-ide/) upgrade does the trick. Support for all languages and frameworks is provided through Komodo IDE, making it ideal for web development. Also, this upgrade doesn’t cost you anything since it’s also an open source project.

[](https://www.activestate.com/products/komodo-edit/)Komodo Editor text editor

However, I would recommend trying out Komodo Edit first to see if it has all of the tools you need for projects. After all, it’s a cleaner interface and lighter weight solution for keeping projects better organized. Komodo IDE has all of the features from Komodo Edit but then it adds on dozens more. For instance, both offer things like the multi-language editor, skin sets, and the ability to make multiple selections.

However, you would have to switch to the Komodo IDE text editor if you need print debugging, or if you’d like tutorials for learning about the system.

**Which Features Make This One of the Best Text Editor Tools?**

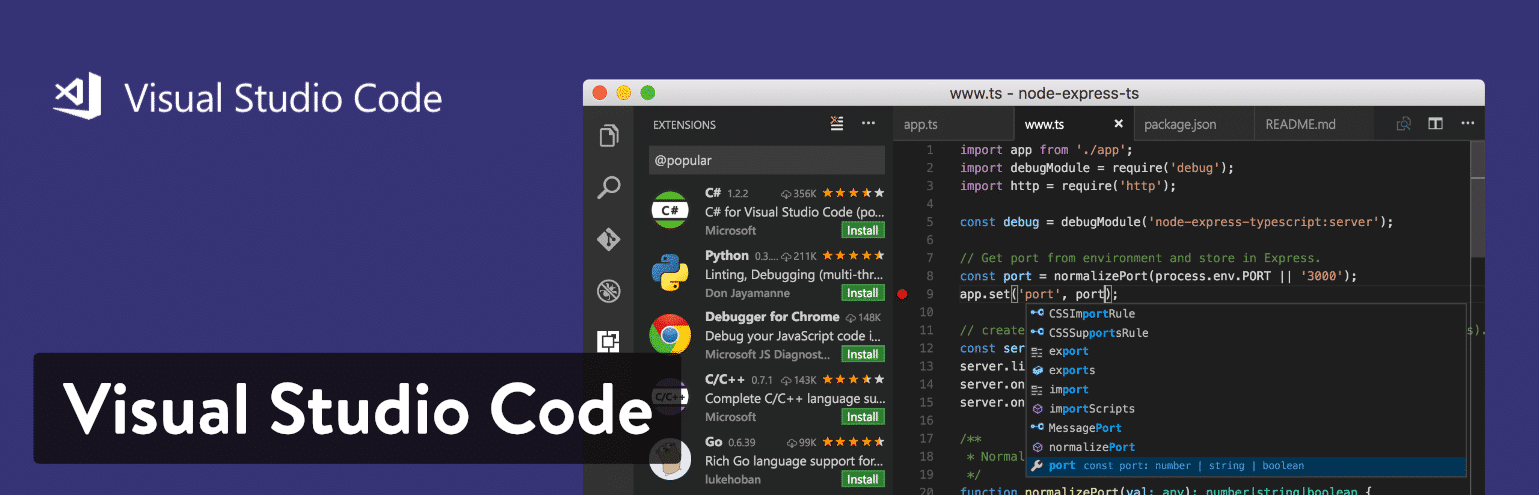
* Komodo Edit is one of the best text editor options since it offers two separate versions, one that’s great for light users and another that’s for the advanced power user.
* Both of the versions, including Komodo Edit and Komodo IDE, can be downloaded for free.
* The Komodo IDE version has real-time code collaboration tools for connecting with other developers on your team and sharing and editing files together.
* The interactive shells in Komodo IDE include Python, Ruby, and Perl.
* The Komodo IDE option also has plenty of integrations. Some of these are Gulp, Grunt, Yarn, and Vagrant.
* The live previewing in Komodo IDE ensures that you can render HTML visuals while adjusting your code.
* The Komodo Edit version has the ability to track changes and make multiple selections.
* Both versions have multi-language editors that include skin and icon sets.

When you think about it, Komodo makes solutions for everyone. Since Komodo Edit and Komodo IDE are both offered for free, less experienced and advanced users can take advantage of the text editor. So, if you’d like a lightweight version, or just something simpler, go with Komodo Edit. If you’re looking for more ambitious tools, the Komodo IDE text editor provides everything from print debugging to custom workspaces.

**11. Visual Studio Code**

As one of the younger players in the game (launched in 2015) [Visual Studio Code](https://code.visualstudio.com/) puts forth quite an effort for building a stable community and ensuring that users are getting the features they need. The hard work definitely shows, since the plugin library has been growing quite a bit. It’s also an open source project that you can download directly to macOS, Windows, or Linux for free.

A few distinct areas make Visual Studio Code an appealing text editor for all developers. First of all, Visual Studio Code is often considered to be faster than older editors on the market. You can also clear out all of the clutter by opting for the Zen Mode, which removes all menus and items that don’t involve your editor.

[](https://code.visualstudio.com/)Visual Studio Code text editor

We also enjoy the IntelliSense feature, which takes syntax highlighting and auto-completion to the next level–with the help of smart completions based on function definitions and other items like variable types. The Visual Studio Code editor has Git commands programmed into the system. This way, you’re able to pull or push from all hosted SCM services.

Finally, the Visual Studio Code website has numerous tabs for you to learn about the software. The documentation page walks you through steps like the setup and working with different languages. You’re also able to check out some tips and tricks and learn all of the Visual Studio Code keyboard shortcuts. Along with a blog, updates page, extensions library and API information, Visual Studio Code looks like a great choice.

**Which Features Make This One of the Best Text Editor Tools?**

* Visual Studio Code is a completely free text editor with open source access and a large collection of extensions to make some upgrades.
* The community is strong and the website includes plenty of documentation, a full blog, and lots of information in the form of API and [FAQs page](https://kinsta.com/blog/wordpress-faq-plugins/).
* It provides built-in Git commands.
* The IntelliSense feature does a wonderful job of upgrading the standard autocomplete and syntax highlighting features you would expect from most text editors. Essentially, it creates smart completions based on items like function definitions and variable types.
* You have the option to debug your code directly in the editor. Therefore, there is no reason for print statement debugging.
* This is a multiplatform text editor, with support for Linux, Mac, and Windows.

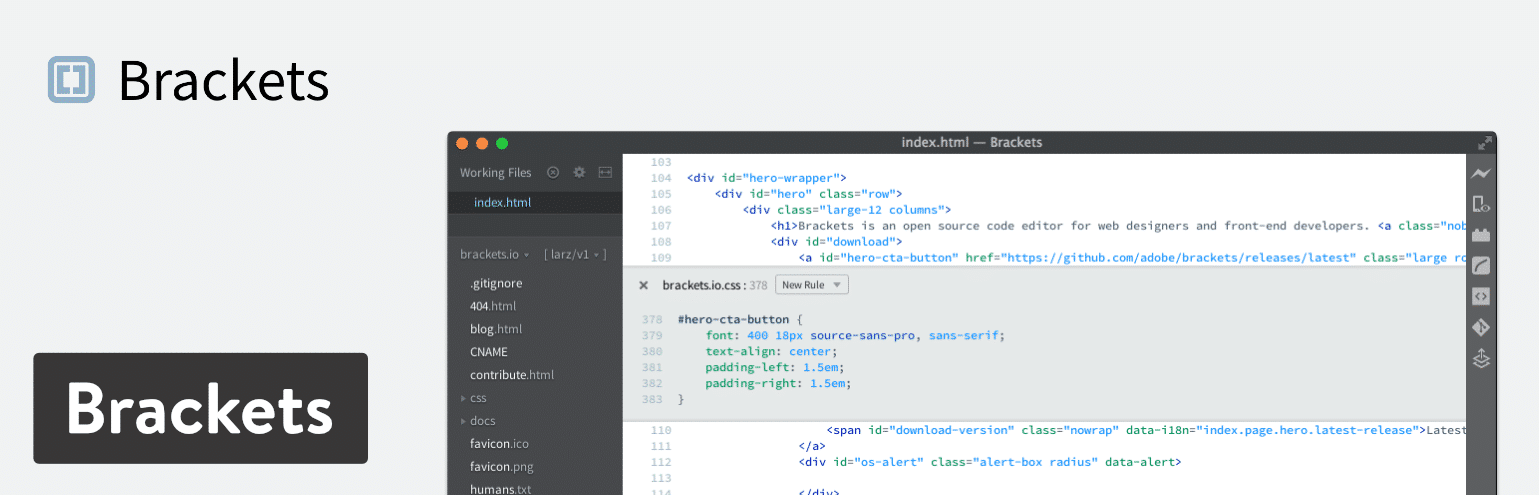
There are a few reasons you might consider Visual Studio Code for your text editing needs. The first is if you would like a free text editor that actually has a thriving community behind it. The second reason would be if you like the idea of extending and customizing your text editor in terms of language, theme, and debugger support. We also really like the IntelliSense feature, so it makes sense for people who are also intrigued by the smart completions.

Looking for a cool Visual Studio Code theme? Check out [Shades of Purple](https://marketplace.visualstudio.com/items?itemName=ahmadawais.shades-of-purple), created and maintained by [Ahmad Awais](https://twitter.com/MrAhmadAwais).

**12. Brackets**

The [Brackets](http://brackets.io/) text editor comes from the folks at Adobe, in an attempt to offer a more modern, open source solution for developers creating websites. This is a free text editor, with some appealing visual tools for previewing your work and allowing for frontend developers to examine the changes. Writing code is the main focus in Brackets, and it’s done with the help of inline editors, live previews, and nicely organized files.

Because of its affiliation with Adobe, Brackets has somewhat of an integration with Photoshop, in that it can take pieces of your PSD file and generate code for your project.

[](http://brackets.io/)Brackets text editor

Brackets is open source and offers a decent community. It’s also available on Mac, Linux, and Windows machines. The interface is fairly simplistic, but the text editor has a wide range of features for you to play around with. In addition, developers are able to upload their own extensions via GitHub. If you would like to use one of these extensions, all you have to do is go to the website.

**Which Features Make This One of the Best Text Editor Tools?**

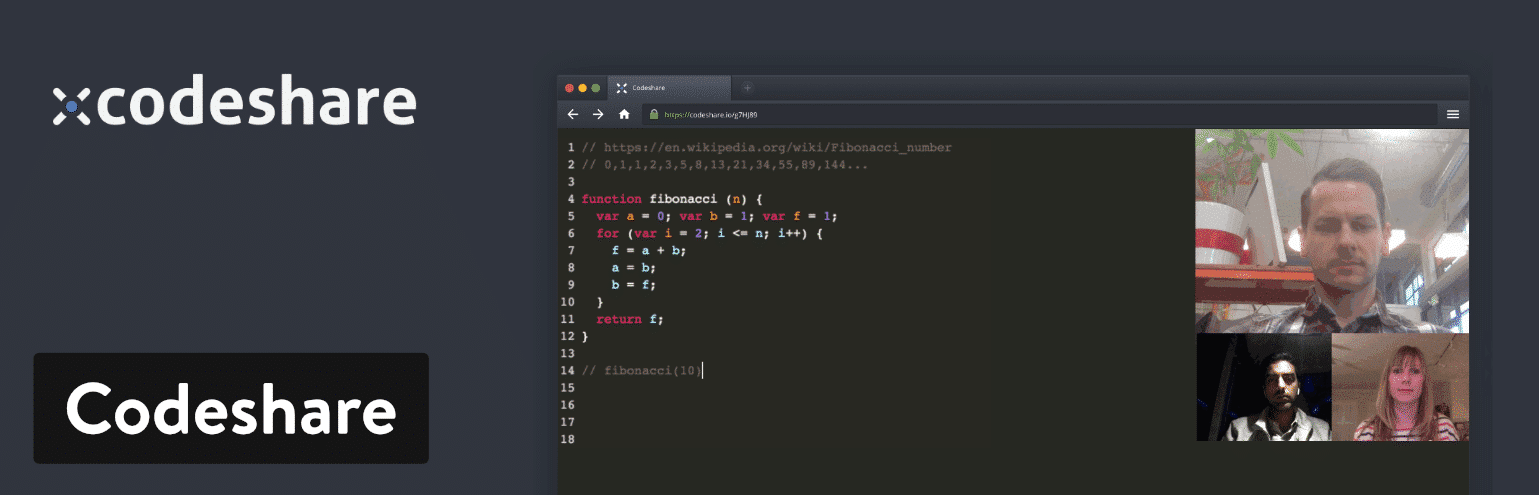
* Brackets delivers on value and its feature-set, especially considering you don’t have to pay a dime and it’s specifically designed for web developers.
* It has somewhat of an integration with PSD files.
* The text editor connects with GitHub.
* Several extensions are provided for you to expand upon the current text editor and potentially customize what your interface looks like.
* Brackets is a cross-platform solution for use on all of your operating systems.
* The live preview feature ensures that all developers can see their work in a more visual manner.
* The inline editors let you select an area inside of your code and open a window right in the editor. This way, you don’t have to have several tabs open while completing your work.

Brackets has plenty of features to consider, but it seems like developers will most enjoy the visual live previews–considering you can’t get this type of presentation in most text editors. It’s also nice for those who want a good value for a free text editor, considering the extensions and overall community vibe is strong.

**13. CodeShare**

The [CodeShare](https://codeshare.io/) text editor takes a completely different approach to online code editing. It’s built for developers, and it has a focus on giving these developers the opportunity to share code in real time and speak to each other through a video chat. So, essentially it’s a real-time code editor combined with a Skype-like communication service.

The reason we like this configuration so much is because the development teams no longer have to be in the same room to see and edit changes right before their eyes. This is highly beneficial for interviews, considering you could remotely hire someone and see their coding abilities along with their face. You also might use it for a debugging session or to review any type of code for your organization.

[](https://codeshare.io/)Codeshare

Keep in mind that any code written on CodeShare is only stored in the text editor for 24 hours. It’s then deleted. So, you either have to save it to your own machine or you have to sign up for an account at CodeShare. A sign-up is not technically required but it does give you that huge feature of saving your code.

Other than that, there is no fee to get started with CodeShare. All you have to do is create an account and then you receive access to the video chat functionality as well as the real-time coding.

**Which Features Make This One of the Best Text Editor Tools?**

* CodeShare stands out as the very first text editor to offer real-time sharing.
* This is also one of the best text editors because it has an integrated video chat solution for logging in multiple members of your team. This might come in handy for interviews or team sessions where seeing a person’s face and hearing their voice would be beneficial.
* It’s a completely free text editor.
* CodeShare is a fairly bare-bones code editor, making it perfect for those who like fewer distractions.
* Your code can be saved if you sign up for a free account.

First of all, Codeshare is made primarily for [developers](https://kinsta.com/blog/hire-wordpress-developer/). So, it really doesn’t make sense to use it if you are a content creator or publisher. That said, Codeshare should be considered if you like the idea of having a video chat embedded into your online code editor. You don’t necessarily have to always use the [video editor](https://kinsta.com/blog/free-video-editing-software/), but it is there as a feature. It’s also worth looking into if you want one of the best real-time code sharing solutions on the market. Overall, we would recommend it for those who would like to code with their teams, interview developers, or teach other people how to program through video.

**Summary**

If you’re still a little confused about which text editor to choose, here’s a list of our final recommendations (remember, we recommend trying each of them out to find the one that works best for your workflow):

* **Sublime Text** – This one makes the most sense for heavy coders. It is lightweight with low resource usage.
* **Atom** – We like Atom for people who also want a collaboration tool with the text editor.
* **Notepad++** – Consider this text editor if you’re a beginner or advanced user. It has the features developers need without being too confusing for those just starting out.
* **CoffeeCup HTML Editor** – If you’re learning coding languages, think about trying this editor.
* **TextMate** – Consider TextMate if you have a Mac and need support for all languages.
* **Vim** – Use Vim if you’re an experienced developer who might enjoy an older interface or prefer something via the command line.
* **UltraEdit** – If you need to upload and edit large files, UltraEdit does the trick.
* **Coda** – Go with Coda if you’re using a Mac and would like an incredibly pretty interface.
* **BBEdit** – This one is also good for Mac users.
* **Komodo Edit** – You can use Komodo as a beginner or experienced pro. Just make sure to download the right version depending on which one you are.
* **Visual Studio Code** – Here’s a text editor with a unique auto-completion feature. Try it out if that sounds interesting to you.
* **Brackets** – Try out Brackets if you like live previews and extensions.
* **CodeShare** – Consider CodeShare if you’re a developer or teacher who could use real-time code sharing and a video chat component.
* **Understanding the Basic of HTML**
* Webpages are created using **Hypertext Markup Language (HTML)**, which is an authoring language used to create documents for the web. HTML consists of a set of special instructions called **tags**to define the structure and layout of content in a webpage. A browser reads the HTML tags to determine how to display the webpage content on a screen. Because the HTML tags define or “mark up” the content on the webpage, HTML is considered a **markup language**rather than a traditional programming language. HTML documents are made up of text content and special codes known as *tags*that tell Web browsers how to display the content. HTML documents are identifiedby their .html or .htm file extensions.
* **Tags and Elements**
* An HTML element consists of everything from the start tag to the end tag, including content, and represents a distinct part of a webpage such as a paragraph or heading. Before we start adding tags to our document, let’s look at the anatomy of an HTML element (its syntax) and firm up some important terminology.

|  |
| --- |
| **The Anatomy of an HTML Element** |

* Elements are identified by *tags* in the text source . A *tag* consists of the element within angle brackets  (**< >**). The browser knows that any text within brackets is hidden and not displayed in the browser window. The element name appears in the *opening tag* (also called a *start tag*) and again in the *closing* (or *end*) *tag*preceded by a slash (/). The closing tag works something like an “off” switch for the element. Be careful not to use the similar backslash character in end tags. The tags added around content are referred to as the *markup*. It is important to note that an *element* consists of both the content *and*its markup. Capitalization in HTML, the capitalization of element names is not important (it is not case-sensitive). So <html>, <Html>, and <HTML> are all the same as far as the browser is concerned. However, most developers prefer the consistency of writing element names in all lowercase (see Note).

|  |
| --- |
| **Note** |
| *There is a stricter version of HTML called XHTML that requires all element and attribute names to appear in lowercase. HTML5 has made XHTML all but obsolete except for certain use cases when it is combined with other XML languages, but the preference for all lowercase element names has persisted.* |

* **Basic Document Structure**
* Œ It is a document type declaration (also called DOCTYPE declaration) that lets modern browsers know which HTML specification to use to interpret the document. This DOCTYPE identifies the document as written in HTML5.
*  The html element is called the root element because it contains all the elements in the document, and it may not be contained within any other element.
* Ž Within the html element, the document is divided into a head and a body. The head element contains elements that pertain to the document that are not rendered as part of the content, such as its title, style sheets, scripts, and metadata.
*  meta elements provide document metadata, information about the document. In this case, it specifies the character encoding (a standardized collection of letters, numbers, and symbols) used in the document as
* Unicode version UTF-8.  Other types of metadata provided by the metaelement are the author, keywords, publishing status, and a description that can be used by search engines.
*  Also in the head is the mandatory title element. According to the HTML specification, every document must contain a descriptive title.
* ‘ Finally, the body element contains everything that we want to show up in the browser window.
* **Attributes and Values**
* You can enhance HTML elements by using attributes, which de­ne additional characteristics, or properties, of the element such as the width and height of an image. An attribute includes a name, such as width, and can also include a value, such as 300px, which sets the width of an element in pixels. Attributes are included within the element’s, start tag.
* You can assign specific attributes to each HTML tag to customize its behavior. Most attributes work by setting a numeric or descriptive value. For example, you can set a paragraph’s alignment on the page using the align attribute along with a type of alignment: left, right, or center.
* Attributes always go inside the opening HTML tag, and it is good form to enclose attribute values in quotation marks.
* You can enhance HTML elements by using attributes, which define additional characteristics, or properties, of the element such as the width and height of an image. An HTML attribute is a name=value pair that provides more information about the HTML element. Attributes are included within the elements, start tag.
* This are the most common HTML Attributes:
* ª id - The id attribute of an HTML tag can be used to uniquely identify any element within an HTML page.
* ª class - The class attribute is used to associate an element with a style sheet, and specifies the class of element.
* ª style - The style attribute allows you to specify Cascading Style Sheet (CSS) rules within the element.
* ª Title - The title attribute gives a suggested title for the element.
* **Elements for Marking Up Text**
* You ’ ve already seen that every HTML document starts off using the <html>, <head>, and <body> elements. Now Let’sadd different types of text elements to a document.
* ³ The six levels of headings: A heading element implies all the font changes, paragraph breaks before and after, and any white space necessary to render the heading. The heading elements are <h1>,<h2>,<h3>, <h4>,<h5>,<h6> with <h1>being the highest (or most important) level and <h6> the least.
* ³ Paragraphs: The <p> element represents a paragraph. A paragraph always starts on a new line, and is usually a block of text.
* ³ The <a> element defines a hyperlink, which is used to link from one page to another. The most important attribute of the <a> element is the href attribute, which indicates the link’s destination.
* ³ The <link> element defines the relationship between the current document and an external resource. The <link> tag is most often used to link to external style sheets.
* ³ The <li> element is used to represent an item in a list. It must be contained in a parent element:
* s  In unordered lists (<ul>), list items are usually displayed using bullet points.
* s  In ordered lists (<ol>), list items are displayed using numbers or alphabets.
* ³ The <div> element is used as a container for html elements . Any sort of content can be put inside the <div> tag!
* **Empty Element**
* An empty element does not contain any text content; instead, it is an instruction to the browser to do something. Perhaps the most common empty element is <img>, the image element.
* s  <hr> - Represents a thematic break between paragraphs of a section or article or any longer content.
* s  <br> - Produces a single line break.
* s  <link> - Defines the relationship between the current document and an external resource.
* s  <meta> - Provides structured metadata about the document content.
* s  <input> - Represents a typed data field allowing the user to edit the data.
* s  <embed> - Represents a integration point for an external, often non-HTML, application or interactive content.

* **Blank Space**
* You can insert blank spaces within a line of text to indent or add emphasis to your text. You can also use blank spaces to help position an element on a Web page, such as a graphic or photo. Type &nbsp; in the line where you want to add a blank space. To add multiple spaces, type the code multiple times. The code stands for nonbreaking space. Web browsers will not create a line break where you insert these characters.


* **Preformatted Text**
* You can use the preformatted tags, <pre> and </pre>, to keep the line breaks and spaces you enter for a paragraph or block of text. Web browsers ignore hard returns, line breaks, and extra spaces between words unless you insert the preformatted tags. If you type a paragraph with spacing just the way you want it, you can assign the preformatted tags to keep the spacing in place.


* **Comments**
* You can put comments into HTML pages to describe the page itself or to provide some kind of indication of the status of the page. Some source code control programs store the page status in comments, for example. Text in comments is ignored when the HTML file is parsed; comments never show up onscreen, that is why they are comments. Comments look like the following:
* <!-- This is a comment -->
* **Special Characters**
* You can use HTML code to insert special characters into your Web page text. Special characters are characters that do not usually appear on your keyboard. The codes used to insert special characters are called ***entities***. Entities consist of
* number or name codes preceded by an ampersand and ending with a semicolon, such as &frac12; for the fraction 1⁄2 or &para; for a paragraph symbol. Type the number or name code for the character, with an ampersand (&) before the code and a semicolon (;). The Web browser displays the designated character in the text.
* The following table lists the common special characters you can insert.
* **Formatting Text**
* If you use a word processor, you will be familiar with the ability to make text bold, italic, or underlined; these are just three of the ten options available to indicate how text can appear in HTML. The full list is bold, italic, monospaced, underlined, strikethrough, teletype, larger, smaller, superscripted, and subscripted text. Technically speaking, these elements affect only the presentation of a document and the markup is of no other use. You can also achieve a similar appearance using CSS. However, they are still commonly used.
* ª The < b > Element
* The <b> tag is used to display text in bold. The <b> tag only makes the content within it display as bold, it does not give any extra importance to the text.
* ª The <i> Element
* The <i> tag is used to display its content in italic. The <i> tag similar to <b> only makes the including text display in italic font and does not give any extra emphasis to the text.
* ª The < u > Element (Deprecated)
* ª The < s > and < strike > Elements (Deprecated)
* ª The < tt > Element
* ª The < sup > Element
* The <sup> tag is used for display text in superscript. The <sup> tag generally displays the text half a character above the baseline.
* ª The < sub > Element
* The <sub> tag is used for display text in subscript. The <sub> tag generally displays the text half a character below the baseline.
* ª The <del> Element
* The <del> tag is similar to the <s> tag, it is used to display a strike line through the content. The <del> tag implies a removal from the document.
* ª The < big > Element
* ª The < small > Element
* **List Elements**
* Humans are natural list makers, and HTML provides elements for marking up three types of lists:
* \*       Unordered lists
* Collections of items that appear in no particular order
* \*       Ordered lists
* Lists in which the sequence of the items is important
* \*       Description lists
* Lists that consist of name and value pairs, including but not limited to terms and definitions
* HTML lists used to present list of information in well-formed and semantic way. The most common forms of HTML lists are ordered and unordered lists.
* ª **Ordered List**
* You can use numbered lists on your Web page to display all kinds of ordered lists. For example, you can use numbered lists to show steps or prioritize items.
* Ä Also called a numbered list, displays information in a series using numbers or letters. An ordered list works well to organize items where sequence matters, such as in a series of steps. To mark an ordered list, insert the **<ol>** and **</ol>** tags at the start and end of the list. Mark each item in an unordered list with a set of list item tags **<li>** and **</li>.**
* Ä  An ordered list can use numbers, letters, or Roman numerals. The default option is to use Arabic numbers, such as 1, 2, and 3.


* ª **Unordered List**
* You can add a bulleted list to your document to set a list of items apart from the rest of the page of text. You can use a bulleted list, also called an unordered list, when you do not need to show the items in a particular order.
* Ä Also called a bulleted list, displays a small graphic called a bullet before each item of information. In an unordered list, the bulleted items can appear in any sequence. To mark an unordered list, insert the **<ul>** tag at the start of the list and the **</ul>** tag at the end of the list. Mark each item in an unordered list with a set of list item tags **<li>** and **</li>.**
* Ä Unordered and ordered lists have optional bullet and number types.
* Ä An unordered list can use one of three bullet options: disc, square, or circle. If no type is identified, the browser displays the default type, a disc. You can also specify an image to use as a bullet.
* ª **Description list**
* You can use a definition list in your document to set text apart in the format of a glossary or dictionary.
* Ä It contains terms and descriptions.
* Ä Definition List makes use of following three tags.
* §  <dl> − Defines the start of the list
* §  <dt> − A term
* §  <dd> − Term definition
* §  </dl> − Defines the end of the list
* **The type Attribute**
* You can use type attribute for <ul>tag to specify the type of bullet you like. By default, a Web browser reads your  bulleted list as disc type. To change different bullet you must add **type** attribute to the <ul> tag.
* The different value of **type** attribute for unordered list are:
* Circle
* Square
* Disc (default)
* For example, if you want a square type, the code would read <ul type=”square”>.
* The different value of **type** attribute for ordered list are:
* upper-alpha      → A  →    A. B. C. D. . . . . .
* lower-alpha             → a     →    a. b. c. d. . . . . .
* upper-roman           → I     →    I. II. III. IV. . . .
* lower-roman            → i     →    i. ii. iii. iv. . . .
* decimal                   → 1     →    1. 2. 3. 4. . . . . .
* For example, if you want a I type, the code would read <ol type=”I”>.
* **The start Attribute**
* You can use start attribute for <ol> tag to specify the starting point of numbering you need. By default, a Web browser reads your numbered list coding and starts with the number 1. To start with a different number, you must add a **start**attribute to the <ol> tag. The different value of **start** attribute are:
* For example, if the numbering is to start at 5, the coding would read <ol start=”5” type=”1”>.
* **HTML Link**
* Links are an essential feature of all web pages. A link is the “*address*” to a document or a resource located on the World Wide Web or elsewhere within your own Web server. These links are known as *hyperlinks.*Hyperlinks allow visitors to navigate between Web sites by clicking on words, phrases, and images. Thus you can create hyperlinks using text or images available on a webpage.
* **Types of Links**
* ¨     **Link to External Web Pages**
* You can use links on your Web page to direct users to other pages on the Internet. For example, you might include a link on your company Web page to a local city directory detailing activities and hotels in the area. Or you might add a link on a product page to the manufacturer’s Web site.
* ¨     **Link to Other Pages on Your Site**
* If your Web site consists of more than one page, you can include links to other pages on the site. For example, your main page may provide links to pages about your business, products, and ordering information as well as to a map of your location. If you maintain a blog, the home page will usually link to postings that you have created in the past.
* ¨     **Link to Other Areas on the Same Page**
* If your Web page is particularly long, you can provide links to different areas on the page. For example, at the top of a page, you might include links to each section heading or photo that appears below. This allows users to jump right to the information they want to view without having to scroll.
* ¨     **Absolute and Relative Links**
* You can use two types of links in your HTML documents: absolute and relative. Absolute links use a complete URL to point to a specific page on a specific Web server. Relative links use shorthand to reference a page and don’t specify the server. You generally use relative links to reference documents on the same Web site.
* **Anchor Tags**
* The HTML code you use to create a link is called an anchor tag, consisting of the beginning <a> and the ending </a>. The href attribute works within the opening anchor tag to define the URL, or Web address, to which you want to link.
* **Syntax for Anchor**
* *As you can see, the label of a link can be text or another HTML element such as an image. You can use the anchor element to create a wide range of links.*
* **The href Attribute**
* The *href* attribute is used to define the address of the file to be linked. In other words, it points out the destination page. You can create a link in your HTML document that, when clicked, takes the visitor to another page on the Web. You can link to a page on your own Web site or to a page elsewhere on the Web.


* **Example of link:**
* This will produce the following result, where you can click on the link generated to reach to the home page of Tutorials Point (in this example).
* **Output:**
* Click the following link
* [Tutorials Point](https://www.tutorialspoint.com/index.htm)
* **The target Attribute**
* You can add instructions to an HTML link that tell the browser to open the link page in a new browser window. You may add this instruction if you want to keep a window to your own site open so the user can easily return to your page. You use a target attribute within the link anchor element (<a>) to open links in new windows. To make all the links on your page open in new windows, you can use the *BASE* element.
* The following are the possible options for target value:
* **HTML Image**
* Images include everything from graphics and clip art to photographs and other visual objects. An important part of using images effectively on your own site is to understand how browsers display the images for others to view. HTML**img** tag is used to display image on the web page. HTML **img** tag is an empty tag that contains attributes only, closing tags are not used in HTML image element.
* Syntax for Image:
* HTML coding lets you display images as inline elements, which means they appear within the body of the page along with text.
* **Basic attributes of image**
* \*       src Attribute
* →   It specifies the image’s filename and the location of the image’s file.
* \*       alt Attribute
* →   It specify alternate text in case the image cannot be displayed in a browser. The alternate text briefly describes the image.
* \*       height and width Attribute
* →   If you want to give some height and width to display image according to your requirement, then you can set it with height and width attributes of image.
* **HTML Table**
* HTML tables were created for instances when you need to add tabular material. Tables may be
* used to organize schedules, product comparisons, statistics, or other types of information. All of the table’s content goes into cells that are arranged into rows. Cells contain either header information or data, which may be any sort of content.
* **The Structure of Table**
* Picture shows the elements that identify the table (**table**), rows (**tr**, for “table row”), and cells (**th**, for “table headers,” and **td**, for “table data / cell”). Cells are the heart of the table, because that’s where the actual content goes.
* **Spanning Cells**
* One fundamental feature of table structure is cell spanning, which is the stretching of a cell to cover several rows or columns. Spanning cells allows you to create complex table structures, but it has the side effect of making the markup a little more difficult to keep track of.
* ª Column Spans
* →   Column spans, created with the **colspan**attribute in the td or th element, stretch a cell to the right to span over the subsequent columns. Here’s an example of colspan:
* ª Row Spans
* →   Row spans, created with the **rowspan**attribute, work just like column spans, but they cause the cell to span downward over several rows. Here’s an example of rowspan:
* **Cell Padding and Spacing**
* By default, cells are sized just large enough to fit their contents, but often, you’ll want to add a little breathing room around tabular content. There are two kinds of space that can be added in and around table cells:
* ª **cell padding**
* →   Cell padding is the amount of space held between the contents of the cell and the cell border. If you don’t specify any cell padding, the cells will have the default value of one pixel of padding.
* →   Because the **cellpadding** attribute may be used with the table element only, the **cellpadding** value applies to all the cells in the table. In other words, you can’t specify different amounts of padding for individual cells with this attribute.
* ª **cell spacing**
* →   Cell spacing is the amount of space held between cells, specified in number of pixels. This example shows the result of the following markup compared to a sample in which no padding or spacing is specified.