[Getting Started with Visual Studio Code and Building HTML Websites](https://www.codecademy.com/learn/paths/web-development)

[**Upgrade to Pro**](https://www.codecademy.com/subscriptions/proAnnualV4a/checkout?redirect_url=%2Fpaths%2Fweb-development%2Ftracks%2Fstyling-a-website%2Fmodules%2Flocal-website-development%2Farticles%2Fvisual-studio-code)



**Getting Started with Visual Studio Code and Building HTML Websites**

Visual Studio Code is one of the most popular and powerful text editors used by software engineers today.

In this article, we will go over the steps necessary to download a popular text editor called Visual Studio Code, also referred to as “VS Code.” By the end of the article you will be able to create a folder in Visual Studio Code that contains an HTML document that you can open in your web browser.

**I. Introduction**

**What are ‘text editors’?**

Text editors, also called code editors, are applications used by developers to write code. They can highlight and format your code so that it’s easier to read and understand. If you’ve used Codecademy, you’re already familiar with a text editor. It’s the area you write your code in.

Using a text editor is part of creating your “development environment,” the set of tools that you use for working on coding projects. It will allow you to take what you’ve learned on Codecademy and put it into practice as you work on projects on your computer. Not only will this introduce you to tools that are commonly used by professional developers but it also means that you’ve grown as a developer and are ready to start working on your own—great work!

Specific to writing code, text editors provide a number of advantages:

* Language-specific syntax highlighting
* Automatic code indentation
* Color schemes to suit your preferences and make code easier to read
* Plug-ins, or add-on programs, to catch errors in code
* A tree view, or visual representation, of a project’s folders and files, so you can conveniently navigate your project
* Key shortcuts, or combinations, for faster development

You may also have read or heard about IDEs, or “integrated development editors.” An IDE allows you to not only edit, but also compile, and debug your code through one application or interface. While the text editor we recommend isn’t considered an IDE, it has many IDE-like features that make life as a developer easier without needing a lot of resources that an IDE usually requires. The best of both worlds!

**Choosing a Text Editor**

There are a number of text editors to choose from. For example, Visual Studio Code is one of the most popular text editors used by developers. (That’s Visual Studio Code and not Visual Studio, which is slightly different. We want the former, the one with ‘Code’ in the name.) Other popular text editors you may have heard of are Atom and Sublime Text.

Any of these text editors mentioned are great for development but to make things easier, we suggest you start off with Visual Studio Code. Some of the benefits of this editor are:

* Free to use
* Open-source, (meaning a program’s code can be viewed, modified, and shared)
* IDE-like features
* Supported by a large community of users and Microsoft

When you are further along in your coding career, you can try other code editors to see what features work best with your personal development workflow.

**II. Installing Visual Studio Code**

So, we’ve chosen our text editor, now we just need to install it on our computer!

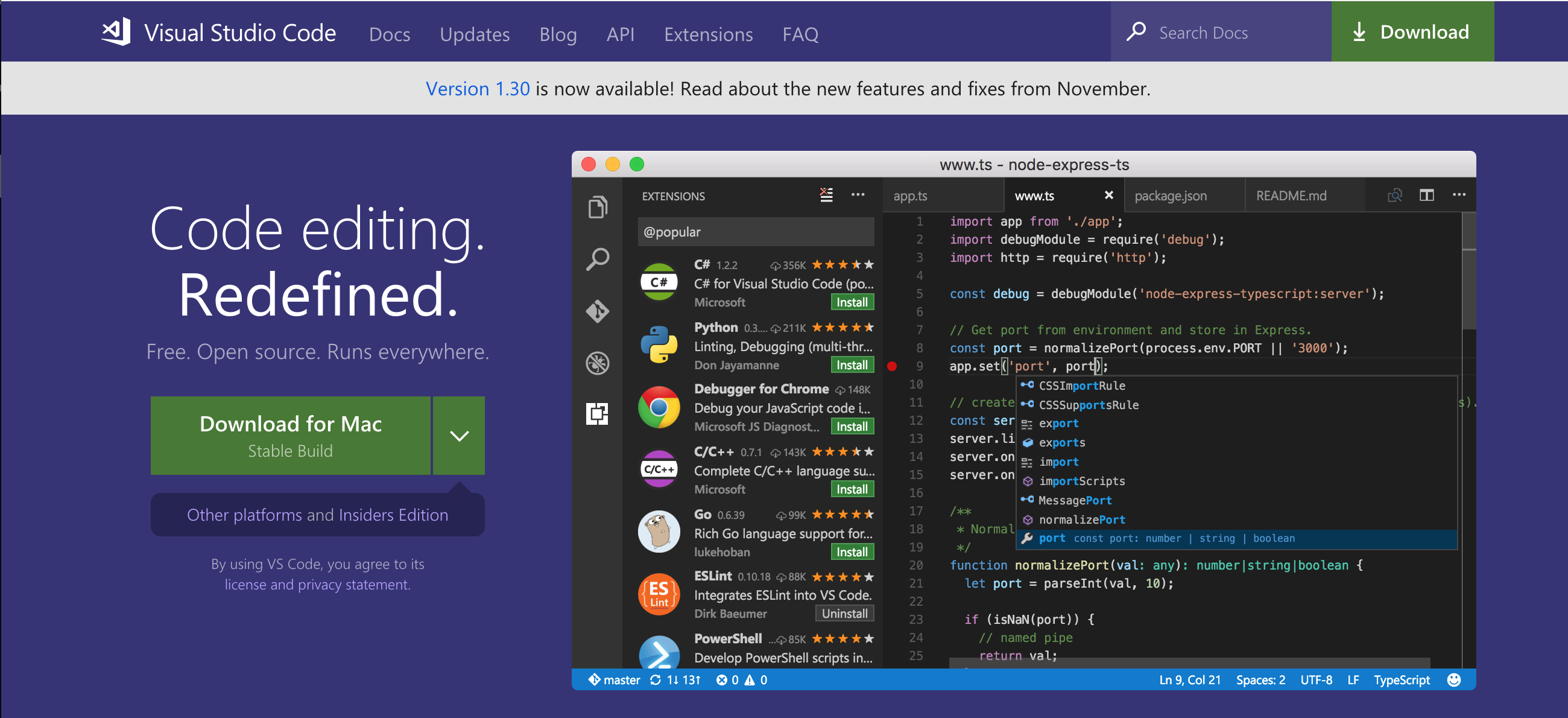
**Video Instructions**

For the visual learners, this video details how to download and install Visual Studio Code. Written instructions are below.

**Installation Steps**

The installation process for computers running macOS, Windows, and Linux, (specifically Ubuntu and Debian), will be very similar and using Visual Studio Code across all of them will be the same.

1. Visit the [Visual Studio Code website](https://code.visualstudio.com/) to download the latest version of Visual Studio Code.



1. You should see your computer’s operating system displayed, but if it’s not correct, click on the down arrow and find the option that matches your operating system from the drop down menu and click on the down arrow icon under “Stable.”

**Windows users:** This will download the latest version of Visual Studio Code as an .exe file.

**Mac users:** This will download the latest version of Visual Studio Code for Mac as a .zip file.

**Linux users:** .deb and .rpm are different file types for storing data. We suggest you download the .deb file so auto-updates work as the Visual Studio Code documentation suggests.

1. Once the Visual Studio Code file is finished downloading, we need to install it. Find the Visual Studio Code file in your file manager, the program that lets you see the files and folders on your computer.

**Windows users:** Open the .exe file by clicking on it and on run the installer. Keep clicking ‘Next’ and then finally ‘Finish.’

**Mac users:** The downloaded .zip file should be in your ‘Downloads’ folder. Open the file. If you see this message choose “Open.”

**Linux users:** The downloaded file should be in your ‘Downloads’ folder.

Find it in your file manager, double click and choose ‘Install’ in the GUI software center, or run the following commands, one at a time, in the terminal:

sudo dpkg -i downloaded\_filename.deb

sudo apt-get install -f

1. Make sure you have your Visual Studio Code application saved in a place you know you will easily be able to find it.

**Windows users:** It will automatically be placed in your Start menu.

**Mac users:** Click and drag the Visual Studio Code icon from the Downloads folder to the Applications folder.

**Linux users:** It should appear in your task bar of programs.

That’s it, you’ve successfully installed your text editor and are ready to start coding!

**III. Practice: Use Visual Studio Code to start an off platform project**

As you move through various lessons and paths here on Codecademy, you may find yourself needing to create a project on your own computer and not on the Codecademy learning environment. This can be tricky, but it’s an exciting step that signals that you are ready to work independently.

To do this, we’ll need to use the text editor we installed above. Let’s take a moment to try out Visual Studio Code.

**What are ‘development folders’?**

Before using your text editor, it’s important to establish an organized file system. As the number and size of your projects grow, it becomes increasingly important to know where to save new projects and find old projects.

Most developers store their projects in an easy-to-find directory, (what you might be used to calling a ‘folder’). Here at Codecademy, we recommend naming this directory **projects**. It will store all of your coding projects. Whenever you create a new project, no matter how small, you should always make a new folder inside your projects directory. You will find that single-file projects can quickly turn into large, multi-folder projects.

**Practice: Let’s make a project**

Below are the steps you need to follow to create a new folder for all of your programming projects. You will also learn how to load a new project folder into Visual Studio Code and make your very first “hello world” HTML project.

We’d recommend that you watch the above video and then follow the written steps below.

**1. Make a development folder.**

Navigate to a folder using your file manager or the terminal. Make sure it is a folder you visit regularly and will remember. Create a new folder called **projects**.

**Mac users:** This may be your User account or “Home” folder.

**Windows users:** You may want to save this on your C drive.

**Linux users:** You may want to save this in your User folder inside of the “Home” folder.

Inside the **projects** folder, create a new folder called **HelloWorld**. Everything you add to this folder will be part of your **HelloWorld** project.

**2. Open Visual Studio Code**

**3. Open your development folder**

Click on the ‘Explorer’ icon on the left hand menu and click on the button ‘Open Folder’ and choose your development folder. This will launch your file manager.

Navigate to the **HelloWorld** folder and select Open. The folder will open in Visual Studio Code’s side pane. At this point, there should not be any contents in the folder. We’ll add a file in the next step.

**4. Add a file.**

Before you learn how to add files to a project folder, it is important to understand the purpose of file extensions. A file extension is the suffix of a filename (the last 3 or 4 characters in a filename, preceded by a period) and describes the type of content the file contains. For example, the HTML file extension is .html, and it tells the browser (and other applications) to interpret the contents of the file as an HTML document. Once Visual Studio Code loads a project folder, you can add files. The steps below describe how to add files. Don’t worry about doing this on your own computer. We’ll get to that next.

In Visual Studio Code’s Explorer pane, click on your development folder’s name. You’ll see four icons appear to the right of the folder name. Click the ’New File’ icon. Type the new file’s name with its appropriate file extension ( for example, .html, .css, .csv). It is critical that you include the correct file extension, so programs like linters know how to interpret its contents. Press Enter when done.

**5. Begin coding!**

Copy and paste the following boilerplate HTML code:

<html>

<head>

<title>Hello World</title>

</head>

<body>

<h1>Hello World</h1>

</body>

</html>

Save your file often with the Auto Save feature and track changes with a version control system if you know how to use one. (To turn Auto Save on, click on ‘File’ then ‘Auto Save’. When it’s on, you’ll see a check mark next to ‘Auto Save’.) This will decrease the chances of losing unsaved work.

*File Extensions and Syntax Highlighting*

Syntax is the set of rules that tell us how to create correctly written code. Visual Studio Code and other text editors are able to interpret file extensions and provide language-specific syntax highlighting. Syntax highlighting is a tool for making code easier to read. Take a look at your index.html file. The text and tags are different colors. This is how Visual Studio Code highlights .html syntax. With each new language you learn, Visual Studio Code will highlight text in a way that makes your code easy to read. This may be different than other text editors and also different than the way your code is highlighted on Codecademy.

*Optional: Change the color scheme*

Although Visual Studio Code comes with default syntax highlighting, you may want to change the colors used. Good color themes will make reading all those lines of code easy on your eyes. (Try out low contrast, dark themes like “Solarized Dark” or “Dracula Dark.”)

To do this, select Color Theme from the Welcome page when you first open Visual Studio Code, or click on Code in the menu bar at the top of your desktop window, then click on Preferences, followed by Color Theme. You can also search for color themes to install using the Extensions menu .

**6. View your HTML file in the browser**

At this point, your file is ready to be viewed in a web browser. The following steps should be taken outside of Visual Studio Code:

Navigate to the **index.html** file in your Hello World folder through your file manager or terminal.

Double click or open **index.html**. The page should open in your default web browser. Take second to marvel at your handiwork—you made your first project with Visual Studio Code.

**Go further with Visual Studio Code’s features**

If you already feel comfortable with the previous steps, explore the following features to further customize your development environment. You don’t need to use these suggestions to complete the projects on Codecademy but they can help make you more efficient when writing code and are what make Visual Studio Code such a useful editor!

* [**Debugging code in the editor:**](https://code.visualstudio.com/docs/editor/debugging) That’s right, you can run and test code from the editor!
* [**Version control:**](https://code.visualstudio.com/docs/editor/versioncontrol) You don’t need to switch to the terminal on your computer to track changes with Git.
* [**Integrated terminal:**](https://code.visualstudio.com/docs/editor/integrated-terminal) You can run command line commands from your editor with Visual Studio Code.

**IV. Wrapping up**

Congratulations! You’ve successfully set up your text editor and are ready to create websites on your own computer.

Happy coding!

[Getting Started with Atom](https://www.codecademy.com/learn/paths/web-development)

[**Upgrade to Pro**](https://www.codecademy.com/subscriptions/proAnnualV4a/checkout?redirect_url=%2Fpaths%2Fweb-development%2Ftracks%2Fstyling-a-website%2Fmodules%2Flocal-website-development%2Farticles%2Ff1-text-editors)



# Getting Started with Atom

In this article, we will go over the steps necessary to download a popular text editor called Atom. By the end of the article you will be able to create a folder in Atom that contains an html document that you can open in your web browser.

## Introduction

Text editors, also called code editors, are applications used by developers to write code. They highlight and format your code so that it’s easier to read and understand. If you’ve used Codecademy, you’re already familiar with a text editor! It’s the area you write your code in.

Text editors provide a number of advantages to web developers:

* Language-specific syntax highlighting
* Automatic code indentation
* Color schemes to suit your preferences and optimize code readability
* Plug-ins to catch errors in the code
* A tree view of your project’s folders and files, so you can conveniently navigate your project
* Key shortcuts for faster development

#### 1. CHOOSING A TEXT EDITOR

There are a number of text editors to choose from. Atom and Sublime Text are two of the most popular text editors used by developers.

Sublime Text has been the text editor of choice for many years. It is stable and reliable.

Atom was released by GitHub after Sublime Text. It’s a fully customizable text editor. Since Atom is written in HTML, CSS, and JavaScript, you can customize it yourself once you’ve learned those languages.

Either text editor is great for development, so you can’t make a bad decision here. When you are further along in your coding career, try another code editor to see what features work well with your workflow.

**Exercise I: Download Atom**

This video details how to download and install Atom.

In this exercise, we recommend you follow these steps to download Atom.

**OS X**

Atom works on Macs running OS X 10.8 or later. Visit the [Atom homepage](https://atom.io/)and click Download For Mac. In a few moments, Atom will appear in your Downloads folder as a .zip file:

Click on atom-mac.zip to extract the application, then drag the new icon into your Applications folder. Double-click the application icon to load Atom and get started.

**Windows**

Atom supports Windows 7 or later. Visit [this](https://github.com/atom/atom/releases/tag/v1.12.9)webpage and download **atom-windows.zip**. In a few moments, Atom will appear in your Downloads folder as a .zip file.

Follow the instructions in the Windows Installer to get started. You can visit Atom’s Windows install page for more detailed instructions.

#### 2. Development Folders

Before using your text editor, it’s important to establish an organized file system. As the number and size of your projects grow, it becomes increasingly important to know where to save new projects and find old projects.

Most developers store their projects in an easy-to-find directory (what you might be used to calling a folder). Here at Codecademy, we recommend naming this directory projects. It will store all of your coding projects. Whenever you create a new project, no matter how small, you should always make a new folder inside your projectsdirectory. You will find that single-file projects can quickly turn into large, multi-folder projects.

**Exercise II: Create a dev folder**

Below are the steps you need to follow to create a new folder for all of your programming projects. You will also learn how to load a new project folder into Atom. For steps 1 and 2, navigate to a folder using Finder (Mac users) or My Computer (PC users).

1. Navigate to a folder you visit regularly and create a new folder called projects. On Mac, this may be your User account. On PC, you may want to save this on your C drive.
2. Inside the projects directory, create a new folder called HelloWorld. Everything you add to this folder will be part of your HelloWorld project.
3. Open Atom on your computer.
4. Atom provides a tree view of your project, so you can conveniently navigate to different folders and files. In the Atom menu bar, choose File > Add Project Folder. This will launch your file manager. Navigate to the HelloWorld folder and select Open. The folder will open in Atom’s side pane. At this point, there should not be any contents in the folder. We’ll add a file in the next exercise.

#### 3. Adding a File

When you open Atom, the Welcome Guide will appear. For now, we’ll skip getting to know Atom and start writing some code.

Before you learn how to add files to a project folder, it is important to understand the purpose of file extensions.

A file extension is the suffix of a filename and describes the type of content the file contains. The file extension is always the last 3 or 4 characters in a filename, preceded by a period. For example, the HTML file extension is .html, and it tells the browser (and other applications) to interpret the contents of the file as a web page.

Once Atom loads a project folder, you can add files. The steps below describe how to add files. Don’t worry about doing this on your own computer. We’ll get to that in Exercise III.

1. In Atom’s top bar, select File > New File. An untitled, blank file will appear.
2. In Atom’s top bar, choose File > Save or Save As. Name the file with its appropriate file extension (.html, .css, .csv). It is critical that you include the correct file extension, so programs know how to interpret its contents.
3. Begin coding! Save your file often. This will decrease the chances of losing unsaved work.

**Exercise III: Add a file**

In this exercise, you will create an **index.html** file in your Hello World project.

1. In Atom’s top bar, choose File > New File. An untitled, blank file will appear.
2. Before you save the file, copy and paste the following boilerplate HTML code:

<!DOCTYPE html>

<html>

<head>

<title>Hello World</title>

</head>

<body>

<h1>Hello World</h1>

</body>

</html>

**Notice:** All of the text in your file is the same color. This will change after you save the file as .html.

3. In Atom’s top bar, choose File > Save or Save As.

4. Name the file **index.html**. It’s crucial that you use the file extension .html so the text editor and web browser know how to interpret your code.

#### 4. File Extensions and Syntax Highlighting

Atom and other text editors are able to interpret file extensions and provide language-specific syntax highlighting. Syntax highlighting is a tool for making code easier to read. Take a look at your **index.html** file. The text and tags are different colors. This is how Atom highlights **.html** syntax. With each new language you learn, Atom will highlight text in a way that makes your code easy to read. This may be different than other text editors and also different than the way your code is highlighted on Codecademy.

**Exercise IV: Open your HTML File in a web browser**

At this point, your file is ready to be viewed in a web browser. The following steps should be taken **outside of Atom**:

1. Back in your file system, navigate to the **index.html** file in your Hello World folder.
2. Double click **index.html**. the page should open in your default web browser.

Congratulations! You can create web pages on your own computer!

[CSS Visual Rules in Chrome Inspector](https://www.codecademy.com/learn/paths/web-development)

[**Upgrade to Pro**](https://www.codecademy.com/subscriptions/proAnnualV4a/checkout?redirect_url=%2Fpaths%2Fweb-development%2Ftracks%2Fstyling-a-website%2Fmodules%2Flocal-website-development%2Farticles%2Ff1-u2-chrome-inspector)



# CSS Visual Rules in Chrome Inspector

In this article, you'll learn how to use browser developer tools to analyze a web page's HTML and CSS, as well as learn techniques to help speed up your development workflow.

## Using Chrome DevTools for CSS Visual Rules

Requirements:

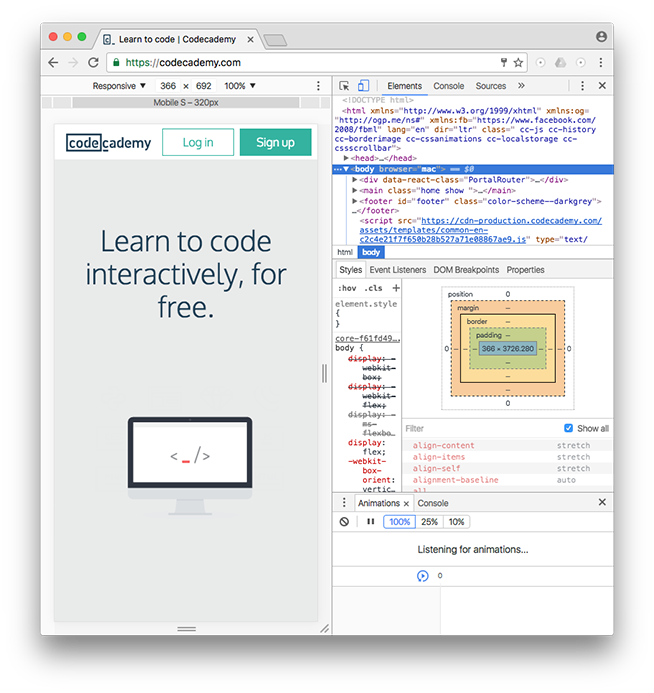
* An Internet browser

#### Introduction

Browser developer tools allow web developers to quickly collect important information on most websites. These tools are available within most major web browsers, like Chrome, Safari, and Firefox, to name a few. Because Google Chrome is the preferred browser for many professional developers, we’ll learn how to use the browser developer tools within Google Chrome, known as Chrome DevTools.

#### Step 1: Accessing DevTools

The quickest way of accessing DevTools in Chrome is to navigate to any website (like this one) and right click (press Ctrl and click for a single button mouse) anywhere on the page. Upon doing so, a menu will appear directly beside the area you clicked on. In the menu, select “Inspect.” This will automatically launch DevTools within your browser. DevTools will appear as a window on either the bottom or right hand side of your screen. It should look something like this:



You should see the Elements, Console, Sources, and Network tabs, among many others.

This rest of this article will focus exclusively on the Elements tab.

#### Step 2: Using DevTools to View CSS Styles

DevTools can provide you with a lot of information about a website, but it’s particularly exceptional at examining a page’s HTML elements, along with the CSS styles for those respective elements. Let’s try it out!

1. Open an incognito Chrome browser (in the browser’s menu, click on “File” then “New Incognito Window”). This will allow you to read this article while completing the following steps.

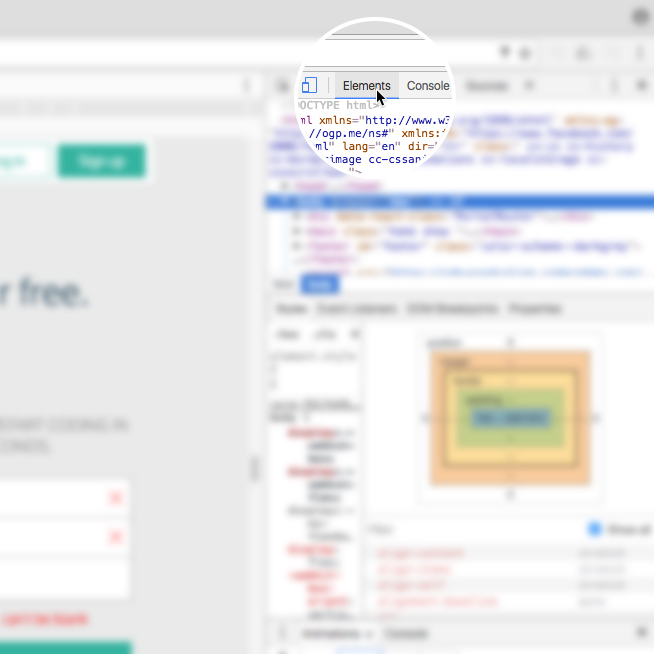
2. Navigate to [Codecademy’s homepage](https://codecademy.com/) (make sure you are logged out).

3. Right click (or Ctrl and click simultaneously) on any text on the page. (This article uses screenshots from a previous version of the Codecademy home page. Your home page may look different from the one shown in the screenshots.)

4. Select “Inspect” in the menu that appears.

5. DevTools should appear at the bottom of your page (it’s normal if its appears in another location, as its location can be changed).

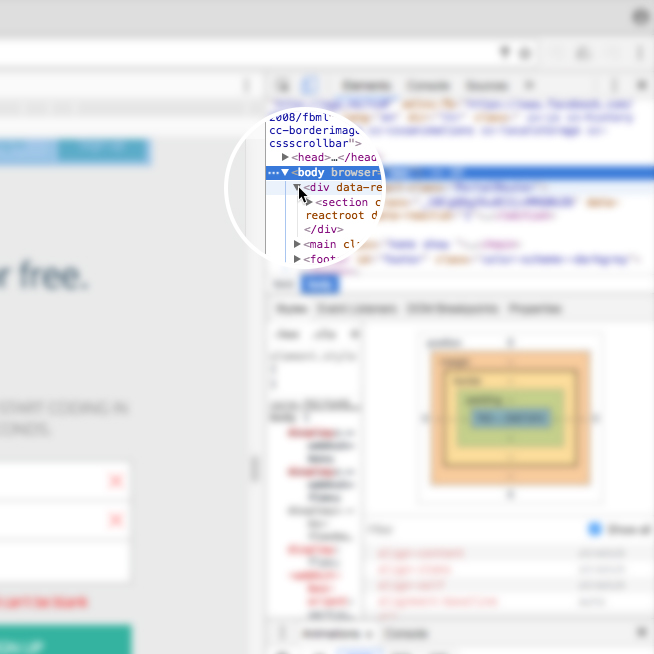
6. Click on the “Elements” tab of DevTools (if you’re not already on it).



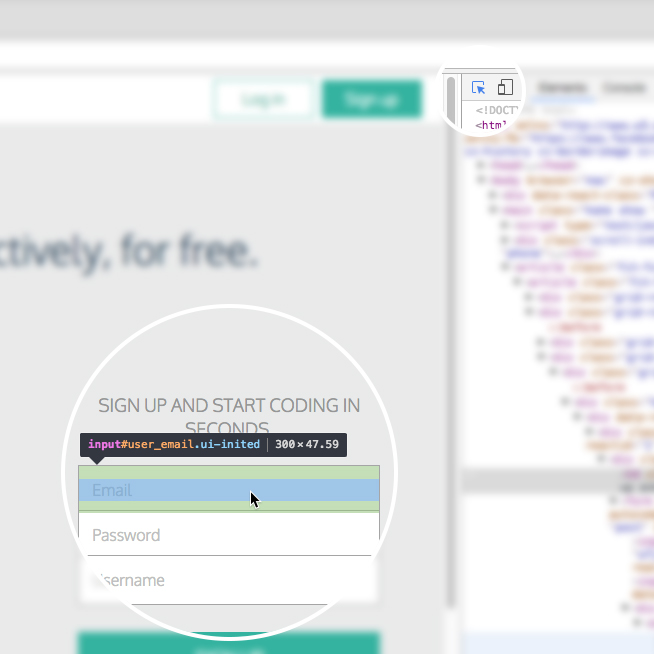
7. In the left pane, notice the interactive DOM (HTML elements) that contains the current content of the web page.

8. Mouse over the HTML code — as you mouse over, notice that DevTools will highlight the corresponding HTML element on the web page.

9. Note that you can expand closed elements by clicking the arrow directly to the left of them.

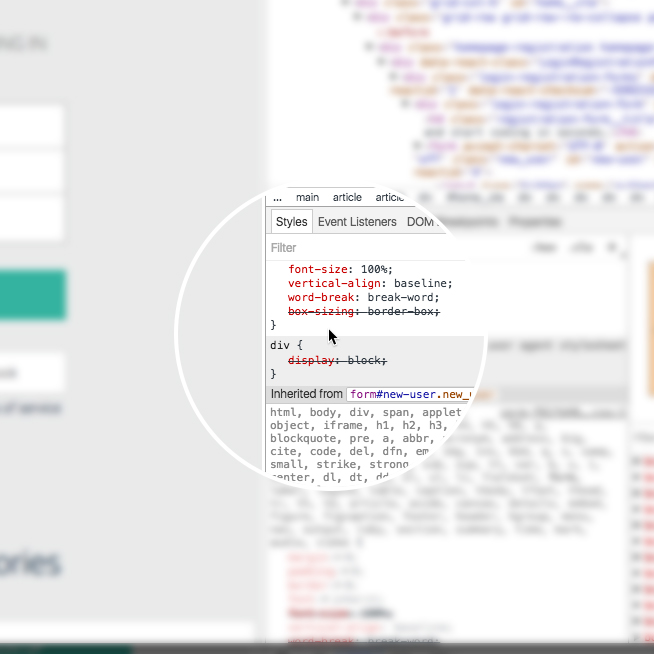


10. Alternatively, click the “Select element” icon (shown in the image below) in the top-left corner of the console and then click on an element within the web page — this is a much quicker way of accessing a specific element on the web page that you want to inspect.



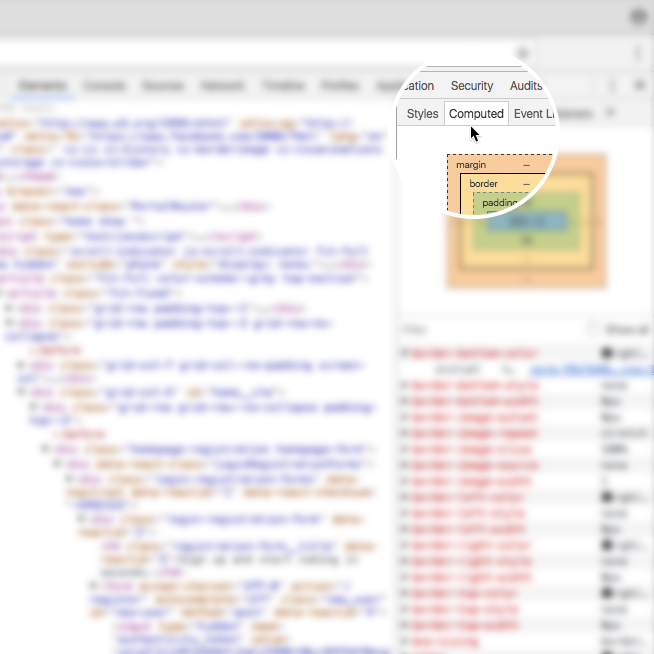
11. On the right hand side of DevTools, click on the tab named Styles (if you’re not already on it) — this tab displays all of the CSS styles associated with the element highlighted in the left side of DevTools.

12. Scroll down in the Styles tab, notice that some CSS styles are crossed out with a horizontal black line.



13. Remember, the Styles tab shows all styles applied to that element (rules can often be overwritten by more specific rules, which causes the horizontal black line through some CSS rules, denoting that that rule is not being used).

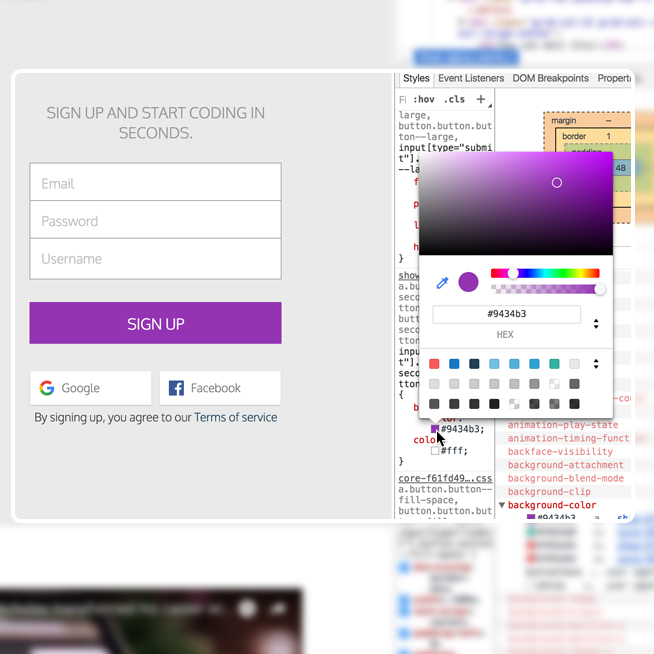
14. To instead view only the styles applied to that specific element, click on the Computed tab directly next to the Styles tab in the right pane. In this pane, you will see only the styles that are being applied to that element, also known as the computed styles. (If the Computed tab is not appearing for you, your browser may be sized too small. Expand the width of the browser until it appears.)



#### Step 3: Modifying CSS Styles with DevTools

DevTools is also useful for modifying existing CSS rules and previewing those changes directly on the page you’re viewing.

To try it out, click again on the Styles tab in the right pane of DevTools (feel free to use the Codecademy website again). Scroll down to a CSS rule (one that is notcrossed out with a black line), click on the value of any applied CSS rule, change the value, and press “Enter” (or “return”) on your keyboard. You should see the change automatically update on the page.



There are a few things to keep in mind when using DevTools to modify a web page:

1. When you modify or change a CSS rule, you may be affecting more than one element.

2. DevTools provides easy-to-use tools when you modify certain CSS rules. (For example, when modifying color values, DevTools will provide you with a color picker to help you select a color.)

3. DevTools is only a sandbox tool, meaning that any changes you make to the web page will not be saved, so make sure to write down any changes you’d like to make when using DevTools for your own web page!

#### Step 4: Add CSS Styles with DevTools

In addition to modifying existing CSS rules, you can add new CSS rules as well. Let’s continue using DevTools on the Codecademy homepage.

1. Locate some text on the home page (i.e. find a heading, paragraph, link, etc.).

2. Right click on the text and click “Inspect” in the menu that appears. DevTools will highlight the corresponding HTML element in its left pane.

3. Take a look at the Styles tab and click on the + icon in the top-right corner of the right pane &ndash; notice that this creates a new, empty CSS rule for that element.

4. Within the element’s selector, click and add a new CSS declaration. The following is an example (feel free to add your own declaration):

background-color: red;

You should see the background color of the text you selected change to red. You can also continue to add your own CSS styling as you wish.

In the future, try this technique on your own website(s) as you build them from the ground up. Building with DevTools can result in a more efficient workflow, as it can help you avoid repeatedly saving and viewing changes manually.

#### Step 5: Modify HTML with DevTools

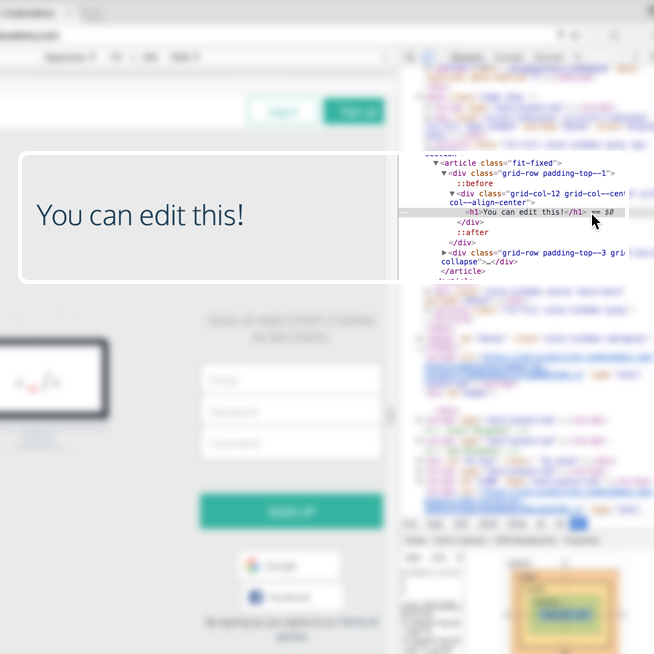
DevTools also lets you directly modify the HTML content of a web page. Let’s try this out one more time on the Codecademy homepage.

1. Again, right click on a piece of text on the homepage and click “Inspect” in the menu that appears.

2. DevTools will automatically highlight the HTML code in the left pane associated with the content that you inspected on the web page.

3. In DevTools, double click on the text between the opening and closing tags of the text you right-clicked on.

4. Change the heading to say something else, like your name, or “Codecademy”, and press Enter.



At this point, you should see the text change!

You can also add HTML of your own as well. Let’s add an <h2> element directly below the element you just modified.

1. Right click on the element you just modified, a menu should appear. Click on “Edit as HTML.” (You can also delete elements using this menu.)

2. A large text field should appear. Directly edit the HTML by adding an <h2> element below with the text of your choice.

3. To complete/view your changes, click on any other element in the left window pane or press Command and Enter at the same time (on a Mac keyboard).

What happens to the web page? Remember, these are sandboxed changes, so your changes will not be saved, nor permanently affect the website you are applying changes to.

#### Review

The Chrome web browser provides you with robust web developer tools known as DevTools. With DevTools, you can view a web page’s existing DOM elements and associated styles, as well as modify and preview changes you make to the web page, resulting in an efficient workflow. If you’re interested in learning more about DevTools, visit the official documentation at <https://developer.chrome.com/devtools>.

Happy coding!