Overview:

## Instructor

* **Released: 3/21/2025**

Looking to get up to speed with HTML (Hypertext Markup Language), the foundation for all websites and web applications? In this course, instructor Christina Truong covers the fundamental concepts and skills you need to know to start writing HTML, including terminology, syntax, how to create a document structure, and more. Along the way, learn how to define the semantic meaning of content to communicate between the web browser and the user. Additionally, explore how to embed various types of media, such as images, videos, and audio, and learn how to create a basic form to capture user input. By the end of this course, you’ll be prepared to start leveraging the power of HTML.

## Learning objectives

* Summarize the role and importance of HTML when creating websites.
* Define the syntax and terminology of HTML.
* Build the document structure of a webpage using semantic HTML elements and attributes.
* List ways to create media and format content to optimize for different screen sizes and devices.
* Create a basic webform for functionality and accessibility.

## Skills covered

### Why should I learn HTML?

Selecting transcript lines in this section will navigate to timestamp in the video

[**- Whether you're looking to build websites from scratch,**customize existing templates, or just want to understand how websites work, learning HTML is your first step into the world of web development. It's the foundation of the web. HTML is the language used to structure and give meaning to the content you see on every website you visit. In this course, you'll learn core concepts and techniques, including how to structure web pages, work with text and links, incorporate images and videos, and create interactive forms. By the end, you'll be able to build well-structured, accessible web pages that work seamlessly across devices and platforms. Hi, I'm Christina Truong, and I invite you to join me for my course on LinkedIn Learning where I'll cover the essentials of HTML.](https://www.linkedin.com/learning/html-essential-training-22425519/why-should-i-learn-html?resume=false&u=103733490)

## ****What You Should Know****

For this course, no prior knowledge of HTML is required, but it’s recommended to be familiar with essential computer tasks such as:

* Downloading and installing software
* Creating files and folders
* Working with multiple applications simultaneously

I'll also be demonstrating the exercises on a Mac computer, but it’s not a requirement. You can follow along on Windows as well. Just keep in mind there may be some differences between the operating systems.

### How to use the exercise files

Selecting transcript lines in this section will navigate to timestamp in the video

[- [Christina] This course includes three types **of hands-on work,**demonstrations and examples where I'll show how different HTML concepts work. You have the option to follow along on your computer or just watch as I go through the steps. There are also exercise that you'll complete with step-by-step solution walkthroughs and end of chapter challenges. This is where you'll work on a guided personal project with solution videos provided. For some of the shorter demonstrations, I'll be using an online tool called CodePen. All of the CodePen exercises in this course can be found in my HTML essential training collection. You're not required to sign up for the service to use it, but if you'd like to keep a personal copy, you'll need to create a free account. To save a copy, select the fork option in the bottom right hand corner. If you don't have an account, you can still make changes to these examples, but you won't be able to save the changes. The exercise files also include a links PDF containing all the links to the CodePens and resources mentioned throughout the course. And speaking of exercise files, they can be found underneath this video next to the instructor details. Click the show all button to download the ZIP file. Once it's downloaded, unzip the folder, and save the exercise files somewhere easy to find, like your desktop. Not all videos will have exercise files, but for the ones that do, they're marked by chapter and movie number. The folders marked Personal Site and Personal Site Generic indicate the end state of the project for that video's challenge. All other exercise files are in their corresponding chapter and movie number with the files marked either start or end to indicate the start and end state of the exercise. And lastly, you'll need a text editor to create and edit files. I'll be using Visual Studio Code. We'll go through the installation and setup together in chapter one, but feel free to download it now if you'd like. By the end of this course, you'll have an actual project framework, you'll create the HTML for a three page website to showcase your personal or professional online presence.](https://www.linkedin.com/learning/html-essential-training-22425519/how-to-use-the-exercise-files?autoSkip=true&resume=false&u=103733490)

### Languages of the web

Selecting transcript lines in this section will navigate to timestamp in the video

[**- For many of us,**it's hard to imagine a world without the web, and for some of us, we actually do remember the world without the web. Since it wasn't available to the public until the early '90s. In 1989, Tim Berners-Lee, the scientist at CERN, introduced hypertechs as a way for researchers to share information without sending emails. Even back then, nobody wanted to deal with an overflowing inbox. He also created the software for the first web server, which is a central area for storing files, and the first web client also referred to as a browser, which is used to view these files. This led to the creation of the World Wide Web. If you'd like to learn more about the history of the web, check out the links PDF in the exercise files, you'll find resources there to explore the background and see the very first website ever created. It's pretty basic compared to what we're used to seeing today. But remember, the web was originally created to share text-based documents, but since those early days, the web has developed into a lot more, including the introduction of more web-based languages. In this course, we'll focus on HTML, which is one part of a group of languages that control what you see in a browser. This is known as front-end or client-side languages and includes HTML, CSS and JavaScript. HTML stands for Hyper Text Markup Language. It's used to format content and create clickable hyperlinks that lets you navigate between different pages on the internet. CSS is a style sheet language used to define the visual presentation, layout, and overall look of the HTML elements. JavaScript is a scripting language used to add interactivity to webpages. Back-end or server-side languages are also used in web development to handle the functionality behind the scenes, such as managing databases or handling user authentication. Some examples of backend languages are PHP, Ruby, Java, which is totally different from JavaScript, and Python. Traditionally, front-end and back-end languages had clear differences. Front-end code runs in the browser, back-end code runs on a server, but with the introduction of JavaScript-based frameworks like Node.js, React, and Angular, those with knowledge of JavaScript can now create functionality that was once reserved for back-end languages. This blurring of the lines has contributed to the rise of the full-stack developer, which is someone who works with both front-end and back-end technologies. The specific languages and depth of knowledge in each area will vary. You might be wondering why it's necessary to learn how to write any of these languages when AI and website builders like Squarespace and Shopify exist. Well, I like to think of AI as a personal assistant. It's there to help you learn and solve problems, but it doesn't replace the learning. AI isn't always accurate, so having some knowledge helps you determine when it's useful and when it's not. As far as website builders go, I often do recommend them when you need something quick and simple. But if you need more customization or want a design that's unique, it's generally more limited. Also, many platforms have an option to add custom code, so if you have some technical skills, you can still use these services and also be able to add your personal touch. And maybe one day, you might work at one of these companies or start your own, developing the products for these services, then you'll definitely need to learn how these languages work. You could even be the person who comes up with the new ideas for these technologies in the future.](https://www.linkedin.com/learning/html-essential-training-22425519/languages-of-the-web?resume=false&u=103733490)

..........................................

### Anatomy of a website

Selecting transcript lines in this section will navigate to timestamp in the video

[**- [Instructor] A website is basically a collection of files**and folders that contain the website's, content, code, style sheets, images, and more. All of these files need to be structured in a specific way so they can pull information from each other. Let's take a closer look at how to organize your files. All projects start with a root folder, which is used to store all the files for the website. Folders are also referred to as directories. You can think of folders as the visual representation, while directory is a technical term for the file system. Both terms are used interchangeably. HTML files must be saved With the HTML file extension. You can write CSS and JavaScript within an HTML file, but they're usually separated into their own files with specific extensions, .CSS for CSS and .JS for JavaScript. There are also other types of files that can be added to a webpage, such as images, videos, and audio files. For basic websites, HTML files are usually stored in the root folder. All other files are usually grouped into related subfolders. For larger projects, more subfolders can be used for additional organization. Linking to different files is a big part of building a website, so following consistent naming conventions is important. Let's go over some tips. Avoid spaces in file names. It may cause errors when the files are uploaded to a web server. You can use a hyphen or underscore to separate multi-word names, but just note that Google's search engine optimization guide recommends hyphens over underscores. The reference link is included in the links.pdf file. Another convention is to use all lowercase letters. Some computers and web servers are case-sensitive and will interpret different letter casing as different files. To prevent any potential issues, stick to using all lowercase letters. Another tip is to use short but meaningful and descriptive names, which is not only helpful for organization, but is also good for SEO. Feel free to pause here to jot down some notes about best practices for naming files. These naming rules aren't required for the root folder itself, since only the files contained within are uploaded to the web server. But it's good practice to be consistent. This helps you stay organized and builds good habits for larger projects. When structuring a website, it can be designed as a single page, where content is grouped into discrete sections within the same HTML file. A multi-page website contains multiple HTML pages that are linked together. In this course, we'll use HTML to create the structure for a multi-page website that showcases your personal or professional profile.](https://www.linkedin.com/learning/html-essential-training-22425519/anatomy-of-a-website?resume=false&u=103733490)

...............................................

### The rules of the web

Selecting transcript lines in this section will navigate to timestamp in the video

[**- There are specific rules for writing HTML.**But where do these rules come from? Well, in addition to creating hypertext and the Web, Tim Berners-Lee also founded the Worldwide Web Consortium, or W3C for short. This organization was formed to establish and maintain the guidelines for writing HTML and CSS. In 2019, the W3C partnered with the Web Hypertext Application Technology Working Group to manage the HTML standards. The documentation for HTML is now available on their website. When I started learning HTML and CSS, I had planned to read the entire documentation, but I found out quickly that it is quite extensive. But the good news is, you don't have to read all of the publications to learn and keep up to date. In this course, I'll go over the core concepts for writing HTML, so you'll definitely get a comprehensive overview. But the scope of all there is to learn is too vast to be covered in a few hours. Also, you're not expected to memorize everything, so knowing how to find answers and having a few go-to resources is key. For example, caniuse.com is great for checking browser support for specific HTML elements and CSS properties. Just search for the feature you're looking for. The results will display a table showing which browsers and versions support that feature. There will also be additional notes and resources included in the results. The W3C also has an HTML validator. You can add a URL, upload a file, or paste your HTML directly into the validator to run a check. Any errors or warnings will be shown below. For documentation, I like to reference the Mozilla Developer Network. In my opinion, it's an easier read, but it also links back to the official specifications in case you want to check the original source. There are also a variety of tutorials and reference guides. You can find all sorts of articles, tutorials, and videos on pretty much any coding subject. But here's a little tip. Check the publish dates to make sure you're not using outdated documentation. And remember, exploring, getting lots of practice, and giving yourself time to learn is all part of the process.](https://www.linkedin.com/learning/html-essential-training-22425519/the-rules-of-the-web?resume=false&u=103733490)

.............................

### Terminology and syntax

Selecting transcript lines in this section will navigate to timestamp in the video

[**- Just like learning any spoken or written language,**the best way is to start by learning the rules of that language. When it comes to coding, we can think of syntax and terminology as basically the grammar, punctuation, and vocabulary rules for that coding language. HTML stands for Hypertext Markup Language and is used to add structure and meaning to content on webpages. You may have also heard of two variations of this term, XHTML and HTML5. XHTML refers to an older version based on the XML language. HTML5 came after XHTML and is the latest version. Each version is updated with new features to keep up with advancements in tech and the changing needs of users. These days, HTML specifications are updated in parts as needed instead of full numbered releases. This means we're unlikely to see in HTML6, but the language continues to evolve and improve. At the end of the day, it's all HTML. There are many types of HTML elements, which are used to mark up the content on a webpage. These elements not only control appearance, but also communicate the content's meaning to the browser, a concept known as semantics. Think about any word processing app or any type of writing tool you've used. The interface usually provides options to format text into headings, subheadings, paragraphs, and lists. You're also able to bold or italicize text. Some other options include embedding media such as images, adding links, and more. When writing HTML, instead of selecting an icon or button, we create these different types of elements using HTML tags. Tags are written by enclosing the element name within left and right angle brackets. The opening tag marks the beginning of the element. The closing tag marks the end of an element and must include a forward slash directly before the tag name. The content to be displayed in the browser is added between the opening and closing tag. The complete structure is the HTML element. Different elements are used for various types of content and each has its own specific tag name. Some are text-based, such as headings and paragraphs, which are defined using an h1 and p tag. Other elements like header and section are used to create page structure. HTML tags can also be written inside of other HTML tags. Most HTML tags are written in pairs, but there are some exceptions. These are called void elements, also referred to as self-closing. They do not require a closing tag. Instead of wrapping content, they are the content themselves, like the hr tag, which adds a divider line. Or they're used to embed web resources like image tags used for images. You may see the old XHTML syntax still being used today, which includes a forward slash after the tag name. While both formats work in the browser, it's best to use the current syntax. HTML tags are not case sensitive. They can be written in uppercase or lowercase letters, but for consistency and to improve readability, it's best practice to use lowercase letters. Attributes are used to add additional information or functionality to an HTML element. They're included in the opening tag after the tag name and separated by a space. They can be defined with or without a value. When used without values, they're called Boolean attributes. Multiple attributes can be added to an element in any order, but must be separated by a space. There are many types of attributes which will be discussed in more details throughout this course. Whether you are working on your own or with the team, there are times you may want to add notes to your code. You can do that with HTML comments. They can only be seen in the source code and will not be displayed in the browser. Comments are often used to hide code for later use, document your work, provide explanations, and to organize the code. To write a comment, start with left angle bracket followed by an exclamation mark and two dashes. To close the comment, use two dashes followed by a right angle bracket. Any type of characters or line breaks can be included in a comment as long as they remain between the opening and closing brackets. Understanding the fundamental HTML terminology and syntax rules will help you write clean, well-structured code.](https://www.linkedin.com/learning/html-essential-training-22425519/terminology-and-syntax?resume=false&u=103733490)

.................................

### Exercise: The development environment

Selecting transcript lines in this section will navigate to timestamp in the video

[**- Making websites requires using various tools,**and as you grow in your development journey, you'll figure out which ones will be essential to your toolkit while others may come and go as your needs change. But at its most basic, you'll only need two things, a web browser and a text editor. Well, technically three things. You'll also need a computer, of course. Web browsers come pre-installed on your computer, Safari on Apple devices, and Edge on Windows. You can also download other options like Firefox, Chrome or Opera. While each browser has its own unique features, they all have the same primary function, displaying webpages. Code must be written in a text editor. There are online text editors like Code Pen, Glitch, and JS Fiddle that could be used in your browser. These tools are great for testing or sharing code snippets, experimenting with ideas and collaborating with others. But for full projects, you'll want to use a desktop application, especially when working with multiple files. Most computers come with pre-installed text editors. usually Notepad on Windows and Text Edit on Macs. While they can be used for writing code, they don't have a lot of features that improve your workflow, such as syntax highlighting, auto completion, and various customization options. Text editors are also optimized for specific programming languages. Some popular options for writing HTML are Sublime Text, Notepad++, and Visual Studio Code. Throughout this course, I'll use Firefox and Visual Studio Code. For some of the smaller demos and exercises, we'll also work with Code Pen. Feel free to use any browser and text editor you prefer. Just note that their interfaces and features may be a little different. If you're new to using text editors, I recommend following along with VS Code. Before we can start writing HTML, we need to set up our dev environment. So head over to code.visualstudio.com and let's go through the setup process. Click on the download button. If you don't see an option for your operating system, click on the other platforms link. Once the download is complete, install it according to your operating system. For me, the zip file is saved to my downloads folder. For Windows, it will be a .exe file. Click on it to start the installer and follow the installation prompts. For Macs, double click the zip folder to open it. Then drag the icon to the applications folder. Double click the icon to open the app. If you're opening VS Code for the first time, you'll see this walkthrough setup page. This usually shows on new installations. We'll be going through some of these setup options, so we can skip this for now. Clicking the welcome link at the top of the page will take you to the main welcome page. This is what you'll see when you open the editor moving forward. The get started page can still be accessed under the walkthrough section. There's also some quick links for actions like creating a new file or opening a folder. I personally like to start with a blank editor, but if you want to keep this welcome page open on startup, leave the option at the bottom of the page checked. If not, uncheck it. Then close this page by clicking the X in the welcome tab. We'll start with a blank editor in the upcoming exercise. So leave VS Code open.](https://www.linkedin.com/learning/html-essential-training-22425519/exercise-the-development-environment?resume=false&u=103733490)

.........................

### Exercise: Customizing your text editor

Selecting transcript lines in this section will navigate to timestamp in the video

[- [Instructor] Like any tool, **a text editor works best when customized**to suit your preferences. Don't worry if you're unsure about what settings to use. I'll share some recommendations which you can adjust as you develop your own workflow. Let's start by adding the folder containing the exercise files to the editor. I have my copy saved to the desktop. If you need help finding the exercise files, refer to the How to use the Exercise Files video in the intro chapter. To add a folder, I like to use the drag and drop method. Just select the Exercise folder and drop it into the workspace. You may see a couple security popups. Select Allow to allow Visual Studio Code to access your desktop. For this popup, select the checkbox and Yes, I trust the authors. The main exercise folder and all the sub folders will now appear in the sidebar. You can also click the File icon on the side menu to open and close this view, which is called the Explorer View. Another way to add folders is to use the menu File, Open Folder..., and navigate to it from there. Now, let's open an example HTML file to see how different settings affect how the contents of the files appear in the editor. Click the arrow to the left of the ch1 folder to expand it. Then inside the 01\_06 folder, double click the example.html file to open it in the editor. The file name appears in the tab. The breadcrumb shows your file location, and position in the code, which will be more relevant when we write HTML. To access the settings, click the gear icon in the bottom-left corner and select Settings. A new tab will appear, showing the settings categories in the sidebar, beginning with commonly used. Go up to the Settings tab, right click or two finger tap if you are using a track pad to open a menu. Selecting any of the split options divides your Editor View so you can see two files at the same time. I'm going to choose Split Right then close the Settings tab on the left. Now let's take a look at some customizations. A common setting you may like to change is the font size. I'll increase mine to 16. Select a number and press Enter. I also like to change the Tab size, which is three options below. It's standard to use four spaces if you like a wider indentation in the code, and two if you like a smaller indentation. I'll set mine to two. Another setting I always turn on is word wrap. Without it, you'll have to scroll horizontally to see the content that goes beyond the editor's width. If you know what setting you're looking for, you can use the search bar. Let's look for word wrap. Under Editor: Word Wrap, select the dropdown menu. There are a few different options and you can hover over each one to see an explanation of what each setting does, but I'll keep it simple and choose on. Now the content will wrap to the width of the editor. I also prefer to disable the mini map. It may be useful for navigating through long pages of code, but I personally have never used this feature, so I would rather have the extra space to display the code. We can disable this option in the settings, but we can also right click directly on the mini map itself. Uncheck the option if you want to disable this feature. Another setting you'll probably want to personalize is the theme, which is the overall colors of the editor's UI and the code syntax highlighting, which uses colors to differentiate between the different parts of the code. In the search bar, let's look for color theme. Currently I'm using the default Dark Modern theme. To switch a theme, just pick an option from the dropdown menu. You can also click the gear icon and select Themes, Color Theme, or use the keyboard shortcut listed to the right to go straight to the dropdown menu. To install more themes and add additional features, we can use extensions. Click on the icon that looks like four blocks in the sidebar and search for themes. You can select an option to view more information about the extension. There's a theme pack that I like called Ayu, A-Y-U. It includes light and dark themes. To add an extension, just click Install and it will be added to your theme options. I'll choose the Mirage Border theme. I definitely spend way too much time looking for the perfect theme and I'm always switching them out, so this is a setting you'll probably want to come back to. You can even make your own themes. There's one more extension I'd like to add, which will come in handy for our exercises. It's called Live Server, and we'll use the version by Ritwick Dey. For now, let's just install it. I'll explain more about how to use it after we set up our project files. Clearing the search bar will show your list of installed extensions, which you can review or uninstall from here. These are just a few features to get you started. To see what else is available, check out the editor's documentation or take a look through the rest of the settings.](https://www.linkedin.com/learning/html-essential-training-22425519/exercise-customizing-your-text-editor?resume=false&u=103733490)

...........................

### Exercise: Setting up your dev environment

- [Instructor] Let's set up the project directoryfor the course challenges,which will be your personal website.Close the pages from the last exerciseso we can start with a blank workspace.And let's go back to the Explorer view in the sidebarby clicking the first icon.There's often more than one way to do the same thing.For example, we can create files and foldersfrom the Menu bar under Fileor work directly in your Finder or File Explorer.But we can also create new files and foldersright in the editor.Let's use this optionand create the project folder hereto keep all the course exercises together.When using the Live Server plugin,keeping all your files in the same root folderalso makes it easier to manage multiple files and folders.Click the folder icon.When naming the folder,I like to use something that represents the project,like a client's name or website domain.If that doesn't apply, use something descriptive.Since this will be a website about yourself,you can name it something like personal-site.Press Enter or Return to create the folder.Now, let's create an HTML file for our homepage.Make sure the personal-site folder is selectedand create the new file from here.Name the homepage index.html.This is requiredbecause web servers look for this file name by default.And press Enter or Return.When you visit a website by entering just the domain name,like websitename.com,the server will automatically load the index.HTML page.All HTML documents require some basic markup,which will be covered in a later video.For now, I'm going to show you a shortcutusing VS Code's built-in auto-complete tool called Emmet.Type an exclamation point.A dropdown will appear with optionsthat match to the typed characters.We're going to use just the single exclamation point.You can either click on it from hereor just press Tab.And all the required tags and attributesare added, all in one step.If you're using a different text editor, this may not workif it doesn't have the Emmet tool integrated.Inside the body tags on line nine, add some text.It can be anything.We just want to have something to see in the browser.When you add content to the page, a dot appears in the tab.This just lets you know that you haveunsaved changes in the file.Once you've added some text, go to File, Save,or use the keyboard shortcut commandor Control + S to save the file.Now, let's view this file in the browser.Go to your Finder or File Explorer,and in the exercise files,go to the personal-site folder we just created.Double-click the HTML fileand it will open in your default browser.We'll only see the text that was addedbetween the body tags.I'll resize my editor and browser windowso we can view them both side by side.Once the file is open in the browser,anytime you make and save a change to the file,you'll need to refresh the browser to see the updates.Now, let's see how the Live Server extensionwe installed earlier can help.You can launch it by clicking the Go Live linkin the bottom right corner,which will open the current page from your editor.You could also right-click on the pageand choose Open with Live Serveror use the keyboard shortcut.It will automatically open in your default browser.When opening files manually,the address bar displays the local file path.With Live Server,we'll see a URL with numbers.This is an IP address.It connects to your computer, which allows us to runand test our changes on a local server.Now, when you make a changeand save the file with Live Server,the browser automatically reloads the updates.You don't have to use a local server for HTML files,but it's useful to be able to use features like auto-reload.There had been many timeswhen I thought my updates weren't working,but it just turns out I forgot to refresh the browser.The plugin isn't required for the exercises in this course.You can open and refresh the files manually if you prefer,but just note that there will be one demothat needs a local server,but you'll still be able to watch the demoand see how it works.And that's pretty much the basicsof using a text editor and browserto write and view your documents.Each chapter challenge builds onto this project,so remember to save your changes.That way, if you close the editor,you can easily reopen the filesand pick up right where you left off.For smaller exercises, we'll use CodePen,which is a great tool for quick demonstrations.And we'll give you a chanceto work with different industry tools.

..........................

## Question 1 of 13

In the following example, which portion of this markup is the opening tag?

<p class="example">Quiz</p> <h1>Quiz</h1>

* 

</h1> and <p class="example">

* 

<h1> and class="example

* 

<h1> and <p>

* 

</h1> and </p>

Submit

2- formatting text :

### Paragraphs and headings

- The majority of content on a webpage is usually text,and now that we've learned the syntax of HTML,let's look at how to create some common text-based elements,starting with paragraphs and headings.To create a paragraph element, use the <p> tag to format oneor more sentences into relevant groupings.For headings, there are actually six tags.They range from <h1> to <h6>and are used to define a hierarchy.<h1> is used for the most significant heading on the page,followed by <h2> and so on.I'll use CodePen to demonstrate how these tags workand how they're displayed in the browser.All the links to the CodePen demos, course references,and extra resources can be found in the Links PDF filein your exercise files.You can follow along with the demos on your computeror just watch; whichever method works best for you.Before we get started, let's look at this editor layout.Since we'll only be writing HTML, I'll collapse the CSSand JS editor panels.Hover over the divider lines,and once you see the icon, drag it over to the right.Then click the boxes icon next to settings.Under change view, I'll choose the first option.This will put the editor panels on the left.This will give a little more roomto see the input and the output.Feel free to adjust the layout to your own liking.In the top of the HTML panel, there are three sentences.If you try to add line breaks or extra spaces,the browser won't recognize the formatting.As we can see in the output, nothing changes.Without HTML tags, all the sentences will displayon the same line and will wrap to the next lineif they can't fit within the width of the viewport,which is usually the visible area in a browser window,but on CodePen, it's just this output panel.The browser only recognizes one space to separate the words;any other white space or line breaks will be ignored.Let's add an opening and closing paragraph tagaround the first sentence and another pair of <p> tagsto group the following two sentences into one paragraph.Start with the left angle bracket, p, right angle bracket,and we'll close itwith another left angle bracket, forward slash, p,right angle bracket.And let's do the same for the second paragraph.HTML is displayed with basic default stylesfor formatting different types of text content.When the <p> tag is used to define the text,a space between the paragraphs is automatically addedto visually separate them.Even when you remove the space between them in the HTML,it doesn't change the default style in the browser.Let's remove the comments around the headings to seehow they're displayed in the browser.We could just delete the brackets and dashes at the topand the bottom of this block,but we can also use a shortcut.Select the entire block,then use the keyboard shortcut cmd + / for Macsor ctrl + / for PCs.For headings, the text is bolded and sized according tothe heading level.The <h1> is the largest,and each heading gets smaller as you go down the levels.If we were to write content that contains six headings,that doesn't necessarily mean all six tags need to be used.Instead, headings should define the organizationof the page content.Let's take a look at the next part of the example,but first, I'll comment out the previous code usingthe same cmd + or ctrl + / shortcut to clear the viewport.Then I'll un-comment this next block.In this part of the example, we can seehow multiple headings can be organized,and <h1> should only be used once per pageand has the highest ranking, so it's used for the titleof the whole article.The next two headings are contained in <h2> tagsbecause they're the next highest in the hierarchy.They're used to split the content into subsections.The following three headings are defined with <h3> tagsbecause they're also subsections,but just within the terminology and syntax portionof the page content.Though there are six headings in this example,we only used <h1>, <h2>, and <h3> tags.We didn't need to go deeper into the heading levelsto structure this content.We're basically organizing the headingsinto a table of contents.The outline is also used by screen reading software,so it's important for accessibility.The outline of the headings from this CodePen examplewould look something like this.The main title is the first item in the outline,followed by the two subsections,then the three subsections within the terminologyand syntax portion of the page.Many HTML elements are displayed with default styles,but it's important to use the appropriate tagfor its meaning rather than how it looks.For example, if you use an <h1> headingfor your main heading, don't skip straight to an <h3>for the subheading just because you want itto appear smaller than how the <h2> looks;that's what CSS is for.HTML defines the contentand CSS changes the styles without changing the semantics.

### Ordered, unordered, and description lists

- Something we use in our everyday lives are lists,like shopping lists or to-do lists.So, it's not surprising that we'll also need to use listsin HTML to organize our content.There are three types, ordered, unordered, and description.Ordered lists are for itemsthat follow a specific sequence,like directions or recipe steps.Unordered lists are for items that do not needto follow a specific order.And description lists are used to organize related termsand their descriptions.Let's take a look at the syntax.All list items are written as li tags,for ordered lists, the items are grouped using ol tags,and for unordered lists, use the ul tag.Only li tags can be nested inside both types of lists.If you want to include introductory texts,use a paragraph element outside of the ol or ul tag.Let's take a look at a CodePen example.In the first block of text,the content contains directionswhich need to be in a specific sequence,so an ordered list is used.When using the ol tag,each list item is shown as a numbered item.The way the numbers are displayedcan be changed by using attributes.If you need to change the starting numberof the first list item, use the start attribute.All attributes are added to the opening tag.Space start= then the value in quotes,which will be whatever number you wantthe list to start with.You can also use the type attributeto change the numbers to letters or Roman numerals.Start with type= then add the quotes.For the value, use a lowercase a for lowercase letters.An uppercase A will display uppercase letters.For Roman numerals, use the lowercaseor uppercase letter I as the value.To stick to the default style, just remove the attribute.The second list represents content you may seeon a shopping site,which can be formatted as an unordered list.The list items can be displayed in a certain orderby organizing them in the HTML,but using a ul tag doesn't conveyany specific meaning about the order.Let's remove the comment around the textto create an unordered list.Then we'll group all the list itemswith an opening and closing ul tag.Don't forget to include the forward slashin the closing tag.Next, we'll wrap each item in an li tag,but let me show you a shortcutthat works in most text editors.Start by typing just the tag name, li with no brackets,then press the Tab key.The editor's auto-complete feature will add the bracketsand the closing tag.This shortcut will work for any HTML tag.I'll add two more list items.Using tab spaces makes it easier to see nested elements.Most code editors automatically add this indentationwhen you press return or enter to create a new line.But when you start moving content around,you may need to adjust the formatting manually.Now, let's move the content to each li tag.You can highlight the contentand drag and drop it between the tags.You can also highlight the contentand right-click to choose cut, copy, or paste.I like to use keyboard shortcuts.Command or Ctrl + C to just copyand Command or Ctrl + X will copy and cut contentall in one step,then use Command or Ctrl + V to paste.Command Ctrl + X to cut and copy,Command or Ctrl + V to paste.Then I'll delete this extra space.If you want to change the display order of the list items,just move it in the HTML.For example, if I want the apparel item to appear first,I will just put it at the top of the list.The list items in an unordered listwill display with bullets.The style can be changed,but unlike ordered lists, it must be done with CSS.Secondary list can also be added to an existing list.For example, you may want to add subcategorieslike shirts and pants under the apparel item.We can do this by creating a nested list.Start by adding the ul tag inside of the list itembut after the text.I'll put the closing li tag on its own lineso we'll be able to see the nested list more clearly.Then add the li tags for the subcategoriesin the nested ul tag for shirts and pants.The nested unordered lists still displays the itemswith bullets, but they're displayedas outlined circles rather than filled ones.The third type of list is the description list,which is used to display terms and descriptions.The dl tag is used to define the entire description list,the dt tag defines the term,and the dd tag defines the descriptionor definition of the term.It's not as commonly used as ordered in unordered lists,but it can be useful for adding semantic meaningwhen displaying terms and their definitions.

### Quotations

- [Narrator] Headings, paragraphs and lists are commonHTML elements you'll likely use for the majorityof your content, but there are additional waysto format text-based content.What if you had a quote and you wantedto cite the source of that quote?You can do that with the blockquote and cite elementsand the cite attribute.Let's take a look at a code pen example.Quotations are defined using a block quote tag.However, the quote's text should be wrappedin paragraph tags.Any attribution must be placed outsideof the blockquote element.The cite tag can be usedbut only for the title of the work, like a book, websiteor movie, not for a person's name.To include a name, use a P tagand use cite to wrap just the title of the work.In this example, I've added the full citationin the comments below.Let's un-comment itand place it right after the blockquote.The cite element can be used for both the article title,and magazine name.The default browser stylefor the content is also italicized by default.Since quotation marks around article titlesare a formatting convention, rather than partof the actual title, I added them outsideof the cite tags.Earlier in the course,we talked about formatting nested HTML tagsby using indentation.In this example, the P tag on line two is nestedinside the blockquote tag, so it's indented once.Paragraphs are also categorizedas block elements, which span the width of the page.If you were to add a second paragraph,it would start on a new line.The cite element is an inline element, which meansthat it's written within the paragraph tagand the browser displays the text within the same lineas the rest of the paragraph text.For inline elements, we generally don't add indentationin the code, but just note that these are just conventionsthat we use to help us read our code better.It's not required syntax.And speaking of indentation, the contentsof the blockquote are displayed with an indentationbecause that's the default style.It's not because of any formatting in the HTML code itself.Let's delete this test paragraph.If you have a reference URL for the source of the quote,the cite attribute can be used by adding itto the opening blockquote tag.The URL of the source is the value,which I've also included in the comments.The cite attribute is not required,but it's recommended to provide additional contextor reference for the quoted content if available.The URL listed in the cite attribute is not displayed in thebrowser, so users won't be able to see or click on it.Instead, it provides metadata to browsers to helpwith assistive technologies and search engines/To make the source clickable, a link can be added.We'll get into more details about usinglinks later in the course.For short quotes within a sentenceor paragraph, use the Q element.The source URL can also be added with the cite attribute.By default, the Q element will automaticallyadd curly quotation marksaround the quoted text.Using the Q element is optional.You could also just manually add quotation marksaround the quote to format it,and your HTML will still be valid without using the Q tag.But if you're really keen about typography,keep in mind the quotation marks typed on your keyboardtypically appear as straight quotesrather than curly quotes.The style of quotation marks may also vary dependingon the font used.However, changing the default font style requires CSS.

### Exercise: Formatting text, part 1

- [Instructor] In this exercise, you'll get some practiceusing the HTML elements we discussed so farto format content using the appropriate HTML tags.You can pause the video right now to try it out yourselfor keep watching to do it together as a code along.Let's start with the first line.This looks like it's the title for a website,so it should be formatted with a heading tag.And since the title represents the pageand any other page in this website,let's make it the most important one.So this should be an h1.We'll start with our opening h1 tagand close it with a /h1.The next line is the titlefor the content of the specific page.And since we've already designated an h1 heading,we should move down the hierarchy and make this an h2.I'm going to use the auto complete shortcutfor the remainder of the tags.So we'll add an h2 and press tab.And then I'll move the content inside the h2 tags.The next two lines can be formatted with a paragraph tag.Since the content of the post dateis separate from the text introducing the list,they should be formatted as separate paragraphs.The text in the introductory paragraphsays the list is in no particular order.So let's format this as an unordered list.Start with the UL to contain the list items.Then I'll add three LI tags.Then I'll move the content in between the tags.Make sure to take some timeto fix the indentation if needed as you go along.It's good to get into the habitof keeping organized early on,before the code gets too unruly.Next is the quote.The example text states that it's meant to be a placeholderfor a longer quote, so let's use the block quote tag.Then we'll wrap the text in a P tag.For the source, we'll create another paragraph tagoutside of the block quote element.I'll move the content in first,and then we can add a site tag around the title of the work.And finally, for the last line of text,we can use another P tag.That's it for this exercise, but keep this pin available.We'll continue to build onto this examplein an upcoming exercise.

### Date and time

- Date and time-based contentis something that is commonly used on the web.For example, news sites include publish dateson their articles,and social media platformsdisplay timestamps for user posts.This can help us understandhow old or recent the content is.It can also be used to add functionality,such as calendar reminders,or to search for results based on a date range.In HTML, the time element is used to definea specific period in time.The content you want to display in the browseris added between the tags.This can be whatever format you prefer.Use the datetime attributeto format the date into machine-readable format.The values can be represented as time on a 24-hour clock,a calendar date, or a duration.To define the time, the values must follow these formats,hours and minutes, hours, minutes, and seconds,or hours, minutes, seconds, and milliseconds.To define the dates,the value can include the year and month,the year, month and day,or just the month and day.You can also use times and dates together.Here's the format for the year, month, and daycombined with the time in hours and minutes.There are quite a few more formatsthat can get pretty specific,especially when using time zones and durations.The full list of valid datetime values can be foundon the MDN docs.Remember, HTMLs used to communicate the semantic meaningof the content to the browser.For interactive features like setting reminders,other programming languages are requiredto add that functionality.

### Bold and italics

- [Instructor] Many text formatting elements are usedto define whole blocks of text,but sometimes you may need to format just a wordor a short phrase contained inside another element.We've already seen an example of this with the cite element,which is an example of an inline element.HTML has more ways to format inline text.For example, there are two tags that can be usedto convey importance or emphasis, strong and em.Here's how the strong tag might be usedand how it's displayed in the browser.This is an example of a warning message you may seewhen you're about to delete an account.Adding the strong tagaround the word warning will designate this wordas more important than the surrounding text.It will also be displayed as bolded.The em element is used to define textthat has stress emphasis to change the meaning of a word,phrase, or sentence.It's like how we emphasize certain words when we speakby pronouncing them with more emphasis.For example, putting the em tagaround the word love would be the same as saying,I love HTML versus I love HTML.Adding the emphasis changes the meaning.By default, the browser will display textusing em tags with italics.There are two more elements that are similar,the b and the i elements.In the browser, they're styled the same.The b tags are bolded and the i tags are italicized.They may look similar,but they serve different semantic purposes.Strong is for content of greater importance, while b is usedto draw attention without indicating a higher importance.For example, b can be used for keywords or a product name.Em represents stress emphasis while i is for textthat is different from normal prose,like common expressions from another language or thoughts.Despite the default styles,it's all about communicating meaning to the browser,screen reader, search engines, and other technologies.So use the appropriate HTML tag to convey the semantics.If you're looking to make text boldor italic purely for decorative purposes,or you want to change the default styles,that's what CSS is for.

### Character entity reference

- [Narrator] Sometimes you may need to use charactersthat are not on a standard keyboard,such as the copyright symbol.You can even include shapes like arrows, or a check mark.This website, HTML Arrows is one of the many resourcesthat you can refer to to see a listof all the special characters and symbols.These symbols can be copied and pasted directly into HTML,but sometimes you may need to use the character reference,which are the special codes shown below each symbol.Let's look at a code pen demo.If I were to write an article about HTML,it would include some code samples,but if I try to add an HTML tag to be displayed as text,the browser will interpret anything enclosed within theangle brackets as HTML.In this case, I can't use the actual brackets.Instead, the character reference can be usedto represent the brackets, also referred to as less thanand greater than signs.Let's go back to HTML Arrowsso we can take a closer look at the different formats.Scroll down the page until we see the less thanand greater than signs.Click on one of the signs to see some examples.The first and last formats, Unicodeand CSS code, aren't used with HTML,so we don't have to worry about them.The next one, Hex Code is a hexa-decimal, numericcharacter reference.It always starts with an ampersand, number sign, Xand ends with a semicolon.The HTML Code refersto a decimal numeric character reference.It starts with only the ampersand and number signand also ends with a semicolon.The HTML Entity refers to a named character reference.It starts with an ampersand, followed by a name.In this case, it's lt, which I'm assuming is shortfor less than.It also ends with a semicolon.You can use any of these three options,but named entities are generally easierto read and remember.I'll copy the named entity to add to my code pen example.If we go back to the main page, we can seethat the HTML entity for the right angle bracketor greater than sign is ampersand, gt, semicolon.Let's go back to the code pen exampleand replace the brackets with these entities.I'll start with the left bracket,since I've already copied it.Then I'll replace just the brackets being used in theexample and for the right bracket,it's the same except the named part, is gt.I'll copy this and replace the remainingright angle brackets.Now they're displayed as text within the paragraphs.Another instance where a character reference is useful isfor including characters that can't be typed directly,such as a non-breaking space.Normally, text automatically wraps onto the next linewhen it reaches the edge of its container.I'm going to add another paragraph.Then use another shortcut to add some dummy text.Lorem, L-O-R-E-M.Then press Tab.Now we have a block of text that wraps over multiple lines.Sometimes you don't want words to breakbetween certain phrases.Let's say I always want my firstand last name to stay on the same line.A non-breaking space can be usedto prevent the default behavior of breaking lineswherever there's a space between words.Instead of a regular space,I can use ampersand, NBSP, semicolon,to add a non-breaking space.Now, the browser knows not to break that spaceand will treat it like one word.The non-breaking space can also be used if your text needsformatting that requires more than one space.Character references are useful,but you're not required to use them.In many cases, you can just insert the special charactersor symbol directly into the code.However, in situations where you can't, knowing howto use character references becomes necessary.

### Exercise: Formatting text, part 2

- [Instructor] Let's go backto the text formatting exercisewe started in Video 02\_04,and add in some of the new HTML tagswe've learned about since then,such as the time element for the post date.Start by wrapping just the date with the time tag.Then add the datetime attribute.It's all one word with no space.For the date value, we can use the year, month, day format.2025-03-01.We learned about the strong and em tags,but in this example, there's really no needto add importance or emphasis to any texts.The headings are already bold by default,but we wouldn't want to use the strong tag for that anyway.We use heading tags for headings,not from making the text bold.The same logic can be appliedto the italicized style of the sighted work.In this case, it's better to keep using the site tagrather than use the em tag just to style it.For the last paragraph, let's add a copyright symbol.I'll just copy the symbol from the reference I sharedin the previous lesson and paste it into the code.We can also add a non-breaking spacebetween the name exampleto prevent them from wrapping onto separate lines.Since we can't copy or type out this character,we can use the HTML entity, "&nbsp;"Now if the text were to wrap,the full name will always be on the same line.And that's the end of this exercise.

### Code and line breaks

- [Instructor] So far, we've talked about different waysto format inline content, such as bold and italics.Let's look at this CodePen demo to explore a few more.One that I often use when writing about codeis the code element.In addition to using HTML entitiesto display the angle brackets as text,we can also use the code element to wrap the text.By default, the text will be displayedwith a monospaced font.If you need to display multiple lines of code,wrap the code block with the pre tag.This will also allow you to use additional spacingor tabs to show indentation in the code.The pre element represents pre-formatted textand is commonly used for displaying code,but it can be used to format non-code-related charactersas well.Another element that can be usedto add additional formatting is the br tagto add a line break in a block of text.It's often used to format content like poems or addresseswhere specific parts of the text needto start on a new line.It's also a void element, so no closing tag is required.Use the br tag where it makes sensefor the content to have a line break.If you just want a particular wordto go on another line strictly for style purposes,use CSS.HTML for content, CSS for presentation.

### HTML and typography

- When writing HTML, the main rule is to use tagsthat describe what the content means, not how it looks,but there are a few HTML tags that are usedfor specific typographical styles,like superscript or subscript.This may seem to contradictthe semantics over presentation rule,but these tags are okay to use as long as you use themto follow standard writing practices.For example, footnotes are notations marked with symbolsor numbers, which correspond to additional informationadded to the bottom of the page.On the web, they're commonly usedon sites requiring citationsor providing supplementary details.These notations are usually formatted as superscript,using the SUP tag.The text will be displayed with a smaller sizeand a raised baseline.There are also sub elements to display text as subscript,which will also display as smaller text,but with a lowered baseline.Let's talk about one more element that can be usedto further define your text content, the small element.It's used for fine print, like disclaimers,copyright notices, or secondary information.The text will appear smaller than the surrounding content.These tags are for inline texts,so you'll still need to wrap a paragraph tagaround the content first.You could also use CSS to change the font stylesfor this type of content,but both methods are valid options.

### Challenge: Write text-based HTML markup for course project

(upbeat music)- [Instructor] In this challengeand all remaining challenges,we'll build onto the same project fileswe created in chapter one.If you've closed your editor since then, just reopen it now.It's been a while since we worked with our project files,so let's review where we left off.In chapter one, we installedand customized our VS Code settings.We also created the project folder, personal site,and the index.html file.If you closed your editor, you will also needto relaunch Live Server as well.Just right-click on the pageand select Open With Live Server.I'll resize the editorand browser window so we could see them side by side.Now let's go over the instructions.Using the HTML tags we learned about in this chapter,add the following content.Since this is a personal website, start by adding your name.We'll add a personal titleor tagline later, so don't worry about that just yet.Next, add a title to introduce yourself,for example, About or About Me.Then add one or more paragraphs containing a bioor anything you'd like to say about yourself.Follow that up with a quote.Then add one last line to include a basic copyright note,and of course, format this contentusing the appropriate HTML tags.All of this content goes between the body tags.Just delete the test text we added in the last exercise.Your content should showcase your personalor professional online presence.But don't worry if you're experiencing writer's blockor haven't developed the copy yet.You can also use placeholder text.Lorem Ipsum is placeholder textthat we used in a previous demo.It's been an industry standard in graphic design,publishing, and web development.It uses nonsensical Latin-derived wordsto mimic the look of actual content.This gives context to your page layoutsfor when the actual copy is not ready yet.You can use this tool to generate some text.You can also use the shortcutthat we used earlier in the demo.Just type lorem and press Tab.Since then, many themed ipsum variations have been created.This site, meettheipsums.com, has gathered themall in one place.Take some time to work this out on your own.You also have the option to readthrough the solution using a text-based formatwith the solution text option.My version of the completed challengeas well as the generic version can also be foundin the corresponding Solution:Demo exercise folder.

### Solution: Demo

(lively music)- [Instructor] Let's walk through the solutionfor this challenge.We'll start with the first instruction, adding your name.Since it's meant to be a personal website,your name represents the most important heading on the page,so let's make it an h1.As you start adding more HTML into the page, you may noticethat in VS Code, when you hover over an element,a pop-up appears.It includes a description of the elementand a link to the MDN reference.So if you want to check if you're using the right element,this could be a helpful feature.Next in the introduction is your personal bio,an about me heading.We've already used an h1, so let's make this an h2.For the bio itself, I'll wrap the text blockin a p tag.I have the content pre-written,so I'm going to copy it from a text filethat I have off screen, so you don't haveto watch me type the whole thing out.Let's save what we have so far.The Live Server plugin is running,so the browser automatically reloaded when I saved the file.Some of my About Me content bio could be turned into links,which we'll learn about later in the course.So if you have references you'd like to include,just add the text for now.We'll learn how to add the HTML markupfor links in a future exercise.I'm also going to close my Explorer sidebarso we can have a little more room to see the code.For the quote, I'm going to start with a paragraphto introduce the content.This is totally optional, so you can include it if you wantto introduce the quote or just go straight to the quotewith a blockquote tag.The text of the quote itself is then formatted with a p tag.I'm going to use the same quotethat I used in the earlier demo.For citations, they must be added outsideof the blockquote, so I'll add another p tag,then add the name of the person quoted.For the citation, use the cite tag,but only around the title of the work.For my quote, there are two citations,the title of the article and the name of the magazine.If there is a URL for the source, it can be addedto the blockquote tag with the cite attribute.Add a space, cite = quotation marksand the URL.For the last item, the copyright note,I'll use another paragraph tag.Then I'll use the copyright symbol followed by the text.Save the file one more time.And that's it.We can see the default browser styles appliedto the various HTML tags, such as bolded textand different font sizes.If some of the formatting looks off,it's time to check your syntax.These issues can occur when a tag isn't closed properly.I'll show an example of this.Let's say I didn't close the h2 tag properly.Here's how it will look.The heading style continues into the following contentbecause the browser doesn't knowwhere to stop applying the stylewithout the proper closing tag.I'll go back and fix that.When writing code,especially if you're just starting out,I recommend writing out a few lines at a time.Then save your file and check for errorsor preview it in the browser as you go.This approach makes it easier to identify any mistakessince you only need to review a few lines of coderather than trying to troubleshoot an entire page.There's no harm in being extra careful.

**02\_13 Solution: Text**

Here is the final solution using placeholder content. For a further breakdown, continue reading below or review the **02\_12 Solution: Demo** video.

<body>

<h1>Your Name</h1>

<h2>About me</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum maiores aliquam, nihil repellendus in blanditiis, excepturi praesentium suscipit adipisci, repellat incidunt odit soluta cumque ipsum temporibus quos labore. Molestias, rem.</p>

<p>(Optional) Another paragraph. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Consequatur iusto officiis, eaque reiciendis molestias iure, dolorem similique sit illo quos adipisci earum velit at, architecto asperiores tempore nemo ipsum quaerat.</p>

<p>Optional intro paragraph for the quote:</p>

<blockquote cite="<https://source/goes/here>">

<p>Add a quote here. Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis perspiciatis voluptatem hic explicabo sunt quia voluptatum eos? Enim impedit, distinctio sit rerum, obcaecati porro, quaerat tempora ducimus eum dolorem ex?</p>

</blockquote>

<p>-Name of person being quoted, "<cite>Title of work</cite>"</p>

<p>© 2025 by Your Name.</p>

</body>

The first two lines represent the main heading and subheading of the page, so they can be defined using an <h1> and <h2> respectively.

<h1>Your Name</h1>

<h2>About Me</h2>

For the bio, use one or more <p> tags to wrap the content.

<h2>About Me</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum maiores aliquam, nihil repellendus in blanditiis, excepturi praesentium suscipit adipisci, repellat incidunt odit soluta cumque ipsum temporibus quos labore. Molestias, rem.</p>

<p>(Optional) Another paragraph. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Consequatur iusto officiis, eaque reiciendis molestias iure, dolorem similique sit illo quos adipisci earum velit at, architecto asperiores tempore nemo ipsum quaerat.</p>

For the quote, if you want to add an introduction, wrap it in a <p> tag. The <blockquote> element should only contain the quote text. However, the text itself should be wrapped in one or more <p> tags, depending on the length of the quote.

The citation should also be added outside of the <blockquote> element, using a <p> tag. If you have a title of a work for the source, it can be included within the paragraph, using the <cite> element. If you have a URL for the source, include it as a cite attribute in the opening <blockquote> tag.

<p>Optional intro paragraph for the quote:</p>

<blockquote cite="<https://source/goes/here>">

<p>Add a quote here. Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis perspiciatis voluptatem hic explicabo sunt quia voluptatum eos? Enim impedit, distinctio sit rerum, obcaecati porro, quaerat tempora ducimus eum dolorem ex?</p>

</blockquote>

<p>-Name of person being quoted, "<cite>Title of work</cite>"</p>

For the last line, it can be wrapped in a <p> tag.

<p>© 2025 by Your Name.</p>

And that’s it!

Save the file and view it in the browser to see the formatting. If some of the formatting looks off, for example, bolded text extending beyond the title, it's time to check your syntax. Review the **Solution: Demo** video to see a walkthrough for how to check for errors.



Up next

### **The basic HTML document**

4m 20s

Dismiss

## Question 1 of 14

What is the primary purpose of the <em> element?

* 

to make text bold

* 

to indicate a different language

* 

to draw visual attention

* 

to indicate stress emphasis

Submit

3 the html document and structure

### The basic HTML document

- Now that we have some practice writing HTML,let's take a step backand look at how to build an entire webpage,also referred to as the HTML document.There are many different elements,and some are used more frequently than others,but there are specific elements that all documentsmust include to function properly in the browser.When we first set up our project, we added a block of codebefore adding any content.Those were required HTML tags.Let's go over the syntaxand purpose of each of these essential HTML elements.The doctype is the first line of an HTML document.While it looks like an HTML tag, it's actually a declarationthat tells the browser which version of HTML is being used.This is an example of the doctypefor the current version of HTMLand can be written in uppercaseor lowercase letters.The HTML element is the first HTML tag.It follows the doctype, making it the root element.All other HTML tags must be contained within.The lang attribute is also includedto help screen reading technology determine the languageof the page content.A two or three character code defines the languageand is written in lowercase.In this example, the value en represents English.You can find a list of all the language codesin various online references.I've included a resource in the Links.pdf file.The next required tag is the head tag,which is used for information about the document itselfrather than the content.Nested inside is the title tag.This is used to define the title of the webpage.The title text appears in three places:the browser's toolbar, bookmarks, and search engine results.The amount of title content displayed varies among browsersand search engines.Each webpage can have a unique title.For example, your homepage may includeyour name and job title.For the other pages, you can add something specificabout the content, like contact or courses.This can help people distinguishbetween the different pages when they viewtheir browser tabs, bookmarks, or search results.Other types of information can beincluded using the meta tag.It's used with various attributesto provide different types of metadata.It's also avoid elements,so it doesn't require a closing tag.Meta tags can be added anywhere within the head section.This example shows two types of metadatathat should be added to every webpage.The charset attribute refers to the character setand is used to declare the page's character encoding.This helps browsers display special characters correctly.The standard value for most languages is UTF-8.While HTML works on its own, you'll likely want to add CSSto style your webpages.Using another meta tag with the name and content attributesensures that those styles will display properlyacross all devices.The name attribute uses the value viewportto define the area being viewed,which is usually the browser window.The content attribute is usedto set the information about the device's viewport."width=device-width" sets the HTML documentto the same width as the device."initial-scale=1.0" means the page won't appear zoomed inor out on page load.Every document should include these two types of meta tags,but there are additional attributesto include more information.The head section can also contain other tagslike the link tag for adding custom iconsor CSS and JavaScript files.This is documented on the MDN website.The link is included in the Links.pdf file.The last requirement for creating a basic HTML documentis the body element.This tag is added after the head tagand all content to be displayed in the browser viewportgoes within the body tags.When using CodePen, these tags aren't requiredbecause the tool has already included thembehind the scenes.This makes CodePen ideal for short demos and quick tests,but when you're creating HTML documents for a website,these tags must always be included.

### Structuring content, part 1

- [Instructor] No two websites are exactly the same,and the layout will vary based on content and purpose,but most websites have the same basic structure:a header at the top, a main content areain the middle, and a footer at the bottom.Historically page layouts were created using div elements.They have no semantic meaning,and were used as generic containersto group related content for styling.CSS was applied to these containers using an IDor class attribute.The naming conventions for the attribute valueswere often unofficially used to define these areas.HTML5 introduced new structural elementsreducing the need to use div tags.These days, the div elements should only be usedwhen no other semantic element is appropriate.The header element is used for introductory contentsuch as headings, logos, a tagline or slogan,a search feature or the main website navigation.A heading tag is commonly used to feature the company name,author name, or a main title.A logo image is also commonly used alongside the headingor sometimes in place of a heading.We'll explore how to add images later in the course.When using a tagline, in the past,I've often used a second heading for the text.However, there's ongoing debateabout the best way to handle taglines in HTMLto make them more meaningful.Since the H1 tag is almost always used for the main title,adding an H2 for the tagline,would suggests that it's a subtitle,which is usually for introducing content in a subsection.Remember, heading tags create a content hierarchythat forms an outline.If we were to create a table of contents,would we include the tagline in the outline?Probably not.A tagline or a slogan doesn't introduce content,but it is content that relates to the H1 heading.While you can use an H2,there are a few other ways to handle this.The tagline can be added in the same heading tagor as a separate paragraph.Another option is to include the H group element.This element has an interesting history.It was first introduced when HTML5 was being developed,but was later removed,but around 2022, it was brought back into the specs.It's used to group a heading and related contentthat represents a subheading, alternative title or tagline.The syntax is specific.It can only be used with one heading elementand one or more P elements.It can also be used without the header element.However, when adding other introductory content like a logo,all other elements must be placed outside of the H group.Then use the headerto group all the introductory content together.When using a header element, an H group is not required.Even a heading tag is not required.The HTML specification states, "A header element"is intended to usually contain a heading,"an H1 to H6 element"or an H group element, but this is not required."The header element can also be used to wrap"a section's table of contents, a search form,"or any relevant logos."Basically, a header can include various typesof introductory content,but doesn't require any specific content.It's flexible and depends on what you want to include.Another thing to note is that headeris different from the head element.Head is for metadata and information about the document.Header is for introductory content.After the header comes the main element.This represents the main content and is unique to the page.Repeated content like navigation, logosor copyright information should not be included here.The main tag can also only be used once per page.At the end of the page is the footer element.This generally contains extra,but related content such as copyright information, credits,links to related documents, a site mapor contact information.With the exception of main,it's possible to use the same structural elementsmore than once on the same page.Let's look at an example.Here is a global header on line oneand a global footer on line 17,which can be used for content that is relevantto the entire websiteand is repeated across multiple webpages.A second set of header and footer elementsseen on line six and line 11can be used for the same semantic purposes,but specifically for the content within the main tag.When following syntax rules,there are specific requirements,but deciding how to use different elementsto describe the meaning of the contentis not always so clear cut.In this example, the article's publication dateand author info is used as introductory content,but this information can also be switched as well,depending how you want to display the content.It could also be displayedas supplementary information in the footer.When starting out with HTML,proper semantics can be confusing,but try not to get too hung up on it.Use what makes sense to you.

### Structuring content, part 2

- [Instructor] Once you've created a basic page layoutwith the <header>, <main>, and <footer> tags,you can use other structural HTML tagsto further organize your content.Elements like <section>, <article>,and <aside> can be used to divide the contentinto more specific groupings.The <section> element is used to define a standalone blockof related content.Each section should include a heading.Unlike <div>, the <section> Element has semantic meaning,but it should be used onlywhen there isn't a more specific HTML elementthat is better suited for your content.It's commonly used for major blocks of content,like different topics in a long article,or distinct parts of a webpage.The <article> element represents contentthat is self-containedand independent from the surrounding content.Think of it as something that would make sense on its own,that can be reused on different pages,or distributed in a feed,like a magazine article,or a product description, or user comments.Each <article> element should also include a heading.<section> and <article> can also be used together.Let's look at an example.In the first example,the contents of the full articleis defined within an <article> element,since this whole piece can be considered standalone content.But the <section> element can be usedfor each topic within the articleto group the related content.The <article> element can also be usedwithin a <section> element.In the second example,<section> is used to group different categoriesof news stories such as featured and most popular.The <article> tag is then usedfor each story excerpt within the category sections.The <aside> element is used for content that is separatebut indirectly related to the main content.It's often used for things you might seeon the side of a webpage like ads,or a author bio if you're on a blog,or highlighted sections from within the main content,like pull quotes from an article.Let's go back to the CodePen exampleand add an <aside> element.Let's say I want to take a catchy line from the articleto be used as a pull quote.The <aside> tag can be placed within the articleto include the pull quote.It's important to use structural HTML elementsto assign meaning to the content,which also helps optimize accessibility.But there is no default styles for these elements,so adding an <aside> elementwon't automatically make it displayon the side of the webpage.We'll need CSS for that.

### Exercise: Structuring content

- [Instructor] In this exercise,we'll use VS Code since CodePen automatically addsthe required tags to the backend tool.By using VS Code, we can practice buildinga complete webpage from scratch.In the 03\_04 folder, I've created an HTML filewith some content for a fictional website.Open the 03\_04\_start.html file to get started.The 03\_04\_end.html file is the solution file for reference.If you still have your project openfrom the previous challenge, that's fine.You can leave it open or close it and reopen the fileswhen we start another project challenge, that's up to you.We'll start by adding the required HTML tags first.Then we'll analyze the contentto choose the appropriate tags based on its meaning.Feel free to pause the video hereif you'd like to work this out on your ownor continue for a step-by-step walkthrough.For the required tags, DOC type, HTML, Head and Body,we could type them out one by one,but the benefits of using a good text editorare the features.We've already used VS Code's built-in autocomplete toolin the chapter one exercise, so let's use it again.Let's add the tags to the top of the page,then we'll copy and paste the text into the tags.Type an exclamation point, then the tab key.And now all the required tagsand attributes have been added.The difference between using this shortcut nowversus the first exercise is now you understandwhat each tag means and its purpose.Now let's add in the basic structural tagswithin the body element: header, main, and footer.When using VS Code's autocomplete tool,you can also press enter or returnto have it auto complete to the first matchthat you see in the dropdown menu.Now let's look at the content to see where they fit.We have the site name 24/7 HTML, followed by a tagline.Since this is introductory content,let's add the appropriate tags to format it.Inside the header tag,add an H1 for the site name and a p for the tagline.Make sure to maintain the indentation for nested tags.I could add an H group hereto group the heading and its related content,but since I don't have any other introductory content,I'll leave it out.Now let's move the text to the appropriate tags.Command or Control + X to cut and copy the text.Then Command or Control + V to paste.And Command or Control + S to save.(keyboard clicks)I like to save my files often,so whenever I add a chunk of code, I'll save the file.Before I started using a keyboard cover,the S key on my keyboard would be completely rubbed offfrom saving my file so often.Once you've added that in, right-click on the pageand choose Open with Live Server.Now let's look at the next bit of text.(keyboard clicks)We can see that there are three categories of content:Latest, Featured, and Most Read Article.Each category also contains an excerpt of each article,so let's use the section tag for each category blockand an article tag for each excerpt.Inside the main tag, add a section tag.Within this tag, add an H2 for the section title,which will be the category.Then underneath the H2, add an article for the excerpt.Within this tag, add an H3 for the title of the articleand a p for the excerpt text.Now let's move the content for the first sectioninto these tags.I'll copy and cut the Latest Article title,add it into the H2,then the article title, and add it to the H3,then the excerpt and add it to the p tag.Now we can repeat this two more times.You may find it easier to move the content firstthen add the tags, use whichever method works best for you.I'll continue to add the tags first,then move the content in.(keyboard clicks)Since I have two sections left,I'm going to copy this whole block and put it underneath.This is also why I like to use keyboard shortcutsbecause there's a lot of copying, pasting,and moving stuff around.Once you've moved your content, save the file.Now we can see the proper hierarchy in the browser,we used the H1 for the website title,H2 for the sections title,and H3 for the title in the article tag.Now this next part is totally up to you.If you prefer your code to look more compact,you can leave it as is,but you may want to add some spacing and notes as well.For some, this makes the code easier to read.For example, we can add a space between the major blocksof content like the header, main, and footer.We can also add a spacebetween each section in the main area.It's also common to use comments to organize the HTML.For example, we can add a commentlike global header and global footerto visually mark these sections.Start with a left angle bracket, exclamation mark,dash, dash, and the editor will automatically addthe closing markup.Then add the comment in between: global header.(keyboard clicks)We can do the same with the footer.I'll just copy this, add it above the footer,and then change this to say footer.Now, if you were to add a secondary headeror footer to the main content area,these organizational notes can help you spotwhere different sections start and end.As you gain experience writing HTML,you'll develop your own preferencesfor organizing your code.These are just some suggestions and common conventions.We have one more line to add.This contains some basic copyright messaging,so we can add this to the footer in a p tag.And that's it, you've successfully createdsemantic, well-structured HTML markup.

### HTML tables

- In the '90s and early 2000s,tables were commonly used to create page layouts.But since then, new improvements to HTML and CSShave made this practice outdated,though you may still see it being used on old websites.Today, tables are mainly used for organizing datain rows and columns,like a price list or a comparison chart.One exception is HTML emailswhere tables are still needed for layouts.Let's take a look at the different partsthat make up a basic HTML table.All forms start with a table tag.The caption element is optional,but can be added to provide a brief descriptionof the table's contents.The rows in a table are created using the tr tag.The th tag is used for table headers.They're usually added as the first row to create a headerfor each column.To add data to the rows, use a td tag.Let's open an example in CodePento see what a basic table looks like.By default, the data is displayed in rows and columns.Each piece of table data appears in its own boxcalled a table cell,but there is no further default formatting,like borders or background colors.So, it's a little difficult to see these table cells.In older versions of HTML, there were attributesthat could be used to style tables,but these attributes are now deprecatedand should no longer be used.Instead, tables can be styled with CSS,which I've included in this example.Remove the forward slash and asterisk at the topand bottom of the CSS block.This is the syntax for CSS comments,which is used in the same way as HTML comments.I've used CSS here to add styleslike borders and text formatting,to display the content more clearlyand make the information easier to read.Let's get back to the HTML.In this example, the table headers are added across the topto define a heading for each column.But you can also define a heading for each rowby adding a th tag as the first item in the row.I'm going to name this one heading, row 2,just so we know where in the table it's supposed to display.I'll do the same for the third row.(computer keys clacking)By adding heading to each row,we've added a third column of content.So, now the column headings are off by one cell,so let's add an empty th tag,before the first heading in the first rowto create a blank cell.(computer keys clacking)This will shift the second and third cells in the top rowto realign the column headings to the corresponding data.Keep this pen open.We'll be using it again for the next exercise.For most scenarios,a basic table like this can be used, but there aremore advanced table formatting options available,when you need to display complex dataor create HTML emails.To learn more, check out the documentation on MDN.It also includes information about deprecated attributes,which is helpful if you need to update old code.

### Exercise: Create an HTML table

- [Instructor] Let's create somethingthat should be displayed in a table format, a sizing chart.We'll use the same pen from the last demoso the CSS can be reused.This will make it easier to see the rowsand columns we're about to create.This is what it will look like in the end.This image is also saved in the 03\_06 exercise folder.You can pause the video now if you'd like to work it outon your own first, or just keep watchingand do it together as a code along.I'm going to start by scrolling the HTML editor area upso we can't see the existing table code from the last demo.Now let's start building the size guidewith a new table tag.Then we'll add a caption tagto describe the contents of the table.I'll call it "Size Guide," with inches in parentheses.Usually you would have more size options,but to keep it simple, we'll create three columnsfor the small, medium, and large sizes.We'll start with the TR tag to make the first row.Then inside the TR tag,add three TH tags to add the headings for the sizes.We'll use S, M, and L, for small, medium, and large.Now, we have the headings across the top for the sizes,but we'll also need to add headings for the rowsfor the width and length data.Let's create two more rows with the TR tag.Then in each row, we'll add a TH tag,one for width and one for length.Now that we've added these headings for the row,we'll need to shiftthe top row of headings over by one cell.Go back to the top row and add an empty TH tagas the first cell.Okay, the headings are set.We can now add in the measurements for each size.I'm just going to use some random numbers hereto make it look realistic for this demo.For the size data, we'll need three TD tagsand values to represent the width for the small,medium, and large sizes.I'm going to go with 20, 21, and 22 inches.And we'll do the same for the length values.We'll need three TD tags here as well,including the length sizes.I'll go with 27, 28, and 29.And there you have it, a completed size chart.

### Exercise: Create an HTML table

- [Instructor] Let's create somethingthat should be displayed in a table format, a sizing chart.We'll use the same pen from the last demoso the CSS can be reused.This will make it easier to see the rowsand columns we're about to create.This is what it will look like in the end.This image is also saved in the 03\_06 exercise folder.You can pause the video now if you'd like to work it outon your own first, or just keep watchingand do it together as a code along.I'm going to start by scrolling the HTML editor area upso we can't see the existing table code from the last demo.Now let's start building the size guidewith a new table tag.Then we'll add a caption tagto describe the contents of the table.I'll call it "Size Guide," with inches in parentheses.Usually you would have more size options,but to keep it simple, we'll create three columnsfor the small, medium, and large sizes.We'll start with the TR tag to make the first row.Then inside the TR tag,add three TH tags to add the headings for the sizes.We'll use S, M, and L, for small, medium, and large.Now, we have the headings across the top for the sizes,but we'll also need to add headings for the rowsfor the width and length data.Let's create two more rows with the TR tag.Then in each row, we'll add a TH tag,one for width and one for length.Now that we've added these headings for the row,we'll need to shiftthe top row of headings over by one cell.Go back to the top row and add an empty TH tagas the first cell.Okay, the headings are set.We can now add in the measurements for each size.I'm just going to use some random numbers hereto make it look realistic for this demo.For the size data, we'll need three TD tagsand values to represent the width for the small,medium, and large sizes.I'm going to go with 20, 21, and 22 inches.And we'll do the same for the length values.We'll need three TD tags here as well,including the length sizes.I'll go with 27, 28, and 29.And there you have it, a completed size chart.

### The Document Object Model

- [Instructor] Throughout this course,we've talked about nesting elementssuch as lists and list itemsand structural page elements like main, article, or section.This nesting actually creates relationshipsbetween the elements known as the document object model,or DOM for short.The DOM is represented as a tree structure,and each node in the tree correspondsto a specific part of the document.The terminology used to describe these relationshipsis pretty much the same as the terms we useto describe relationships in a family tree.Ancestor elements are container elementsand are positioned one or more levels above another element.All the elements nested within are the descendants.A parent element refersto the immediate container of another element.In other words,it's directly one level above its child element.Sibling elements share the same parent.Understanding these relationshipsis essential for writing CSS and JavaScript.Both languages target elements based on their positionin the DOM to apply styles and add functionality.

### Challenge: Add structural elements to project

(bright electronic music)- [Instructor] In this challenge,we'll focus on organizing contentusing appropriate Structural HTML Tags.Here's what we'll do.Update the <title> tag to reflect the page contentin the index.html file.Then add the structural HTML tag's <header>, <main>,and <footer> to organize the content.Number 3, add a tagline to your introductory content area.For additional practice with structural tags,we'll also create a new HTML file.Name this new file blog.html,then add the basic required tags.For third step,use the same <header> and <footer> info as index.html.And finally, for the main content area,create two example blog posts.Each post should include a title,publication date, and a paragraph of texts.Use the appropriate structuraland text formatting HTML tags.Since we haven't created linksbetween the two pages yet,you can view the blog page in two wayswhen using Live Server.One option is to add the file name after the root folderand the backslash in the browser's address bar.For example, after personal-site, which is the root folder,just rename this to blog.html.Another option is when you open up the new page,just right-click from here and select Open with Live Server.It will open in a new tab,so you'll have your index page in one taband your blog page in another tab.If you use this option, just make sure to close any tabsthat you're not using to avoid accumulatinga bunch of extra tabs,and if you're viewing the files without Live Server,just go to your Finder or File Explorerand double-click the file from there.When you're ready, proceed to the Solution Demofor the video walkthrough,or Solution Text for the tech solution.

### Challenge: Add structural elements to project

(bright electronic music)- [Instructor] In this challenge,we'll focus on organizing contentusing appropriate Structural HTML Tags.Here's what we'll do.Update the <title> tag to reflect the page contentin the index.html file.Then add the structural HTML tag's <header>, <main>,and <footer> to organize the content.Number 3, add a tagline to your introductory content area.For additional practice with structural tags,we'll also create a new HTML file.Name this new file blog.html,then add the basic required tags.For third step,use the same <header> and <footer> info as index.html.And finally, for the main content area,create two example blog posts.Each post should include a title,publication date, and a paragraph of texts.Use the appropriate structuraland text formatting HTML tags.Since we haven't created linksbetween the two pages yet,you can view the blog page in two wayswhen using Live Server.One option is to add the file name after the root folderand the backslash in the browser's address bar.For example, after personal-site, which is the root folder,just rename this to blog.html.Another option is when you open up the new page,just right-click from here and select Open with Live Server.It will open in a new tab,so you'll have your index page in one taband your blog page in another tab.If you use this option, just make sure to close any tabsthat you're not using to avoid accumulatinga bunch of extra tabs,and if you're viewing the files without Live Server,just go to your Finder or File Explorerand double-click the file from there.When you're ready, proceed to the Solution Demofor the video walkthrough,or Solution Text for the tech solution.

## 03\_10 Solution: Text

Here is the final solution. For a further breakdown, continue reading below.

**file: index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Your Name, Optional Title or Tagline</title>

</head>

<body>

<header>

<hgroup>

<h1>Your Name</h1>

<p>Title or tagline goes here</p>

</hgroup>

</header>

<main>

<h2>About me</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum maiores aliquam, nihil repellendus in blanditiis, excepturi praesentium suscipit adipisci, repellat incidunt odit soluta cumque ipsum temporibus quos labore. Molestias, rem.</p>

<p>(Optional) Another paragraph. Lorem, ipsum dolor sit amet consectetur adipisicing elit. Consequatur iusto officiis, eaque reiciendis molestias iure, dolorem similique sit illo quos adipisci earum velit at, architecto asperiores tempore nemo ipsum quaerat.</p>

<p>Optional intro paragraph for the quote:</p>

<blockquote cite="https://source/goes/here">

<p>Add a quote here. Lorem ipsum dolor sit amet consectetur adipisicing elit. Quis perspiciatis voluptatem hic explicabo sunt quia voluptatum eos? Enim impedit, distinctio sit rerum, obcaecati porro, quaerat tempora ducimus eum dolorem ex?</p>

</blockquote>

<p>-Name of person being quoted, "<cite>Title of work</cite>"</p>

</main>

<footer>

<p>© 2025 by Your Name.</p>

</footer>

</body>

</html>

**file: blog.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Blog | Your Name, Optional Title or Tagline</title>

</head>

<body>

<header>

<hgroup>

<h1>Your Name</h1>

<p>Title or tagline goes here</p>

</hgroup>

</header>

<main>

<article>

<header>

<h2>Blog Title</h2>

<p>Published: <time datetime="2025-03-01">01 Mar 2025</time></p>

</header>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolorum neque cum quam atque hic deleniti vero tempora libero? Reprehenderit, inventore. Illo, quaerat debitis. Cum, porro accusamus alias necessitatibus laboriosam commodi.</p>

</article>

<article>

<header>

<h2>Another Blog Title</h2>

<time datetime="2025-02-14">14 Feb 2025</time>

</header>

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Nostrum aut ratione delectus ab esse labore rem assumenda provident deleniti, at ipsam a, praesentium fugit nam illo accusantium ipsa animi perferendis!</p>

</article>

</main>

<footer>

<p>© 2025 by Your Name.</p>

</footer>

</body>

</html>

The <title> text has been updated to reflect the page content for both HTML files.

<title>Your Name, Optional Title or Tagline</title>

<title>Blog | Your Name, Optional Title or Tagline</title>

The <header> and <footer> content is the same for both pages, since this content is used globally for the whole website. In addition, the <hgroup> element was added to further organize the name and tagline as related content. Other introductory text will be added within the <header> but outside of the <hgroup>.

<header>

<hgroup>

<h1>Your Name</h1>

<p>Title or tagline goes here</p>

</hgroup>

</header>

<footer>

<p>© 2025 by Your Name.</p>

</footer>

For the index.html file, the <main> element contains the remaining content that is not included in the <header> or <footer>. For the blog.html file, the <main> element contains the content for the two blog posts.

Each blog post uses an <article> element, since each post represents content that is self-contained and independent from the surrounding content. Each post also contains its own <header> element for the post-specific introductory content: the title and publish date.

<main>

<article>

<header>

<h2>Blog Title</h2>

<p>Published: <time datetime="2025-03-01">01 Mar 2025</time></p>

</header>

<p>Lorem, ipsum dolor sit amet consectetur adipisicing elit. Dolorum neque cum quam atque hic deleniti vero tempora libero? Reprehenderit, inventore. Illo, quaerat debitis. Cum, porro accusamus alias necessitatibus laboriosam commodi.</p>

</article>

<article>

<header>

<h2>Another Blog Title</h2>

<time datetime="2025-02-14">14 Feb 2025</time>

</header>

<p>Lorem ipsum dolor sit amet consectetur, adipisicing elit. Nostrum aut ratione delectus ab esse labore rem assumenda provident deleniti, at ipsam a, praesentium fugit nam illo accusantium ipsa animi perferendis!</p>

</article>

</main>



Up next

### **The hyperlink**

1m 17s

Dismiss

## Question 1 of 8

Where does the page title defined in the <title> tag appear?

* 

only in the browser

* 

in search results and the browser toolbar

* 

only in browser bookmarks

* 

in the browser toolbar, bookmarks, and search results

link and navigation

### The hyperlink

- Earlier in the course,we talked a bit about scientist Tim Berners-Lee,the creator of the Worldwide Web.He wanted to find a better wayto not only organize the information,but to be able to share itwith thousands of researchers at CERN.By the end of 1990,he had developed several key technologies,HTTP, which stands for Hypertext Transfer Protocol,and URLs, or Uniform Resource Locators.HTTP is basically a set of rulesfor transferring data across the web.And URLs are the addresses of the Internet.Just like you need an address to find a house,you need a URL to find a specific website or webpage.Tim Berners-Lee also created a web server to host the files,a browser to view the files,and last but not least, HTML to create the webpages.While all these components are essential,one of the most important elementsof the Worldwide Web is hyperlinks.Hyperlinks allow users to move between different pagesand resources on the Internet by clicking on linksembedded in webpages and written in HTML.This revolutionized how informationis accessed and shared onlineand forms the foundation of the Internet we use today.

### Relative and absolute URLs

- [Instructor] Hyperlinks are createdusing the anchor element, represented by the <a> tag.The content added between the tags becomes clickable.An href attribute is also includedto specify the link's destination.Links can point to project files or external websitesusing either relative or absolute URLs.In a previous lesson,we talked about websites as a collection of files.To link to these files within your project directory,use a relative file path.The location is relative to the file you're linking from.Let's look at an example directory.All the HTML pages are in the same directory,the first level of the main project folder,clothing-store, the root folder.When linking to pages within the same directory,the href value is just the file name.Another way to link to files in the same directoryis by adding a period and forward slashbefore the file name.This method explicitly statesthat the file is in the current directory,while using just the file name implicitly assumesthey're in the same location.Both options are valid,but the ./ syntax is less commonly used.I personally use just the file namewhen referring to files within the same directory.To link to files within a subdirectory,the folder structure must be reflected in the file path.For example, to link to a page inside the apparel folderfrom apparel.html,the starting point is where the HTML file is located,but the endpoint is in a sub-folder.Start the relative file path with the folder nameto direct the browser into the sub-folder.Then add a forward slash,followed by the file name and extension without any spaces.To create links from a sub-folderto a file in the parent folder,the starting point is now inside the sub-folder.For example, to create links in shoes.htmlto navigate to the homepage,apparel page, or contact page,use two dots followed by a forward slash and a file name.The two dots tells the browserto go up one level in the directory structureinstead of looking for a specific folder name.You can use the same syntax to link to any type of file,not just HTML files.Relative URLs are typically usedfor internal links within your own website files,while absolute URLs are generally used for external links.An absolute URL mustinclude the protocol HTTP or HTTPSand the domain name.Web servers automatically navigate to index.html files,so you don't need to include them in the URL.For all other pages and web resources,you'll need to use the complete directory structure,including the file name.

### Types of links

- [Instructor] In the previous video,we covered how to create linksusing relative and absolute URLsto link to internal and external resources.Now let's dive deeper into other types of linkable contentand how to create more accessible links.For accessibility, saying, "Click here," is vagueand doesn't really explain where the link will go.Instead, make your link text more descriptive.For example, if you have a phrase like,"Learn more about our courses here,"instead of making "here" the link,you can make "our courses" a link.This way, you don't even need to say, "here."This approach is more descriptive and provides more context.Hyperlinks can also be used for in-page linking,a technique that creates linksto specific parts of the document.An in-page link will take you directly to that sectioninstead of manually scrolling through the entire document.This method is often used for navigation within a page.Some common examples are FAQs, frequently asked questions,and back to top links.To create an in-page link, add an ID attributeto the element you would like the link to go to,the target element.Then in the a tag, set the href value to the hash symbolfollowed by the same value used in the id.This connects the link to the specific part of the pagewhere the element is.You can also link to a specific spot on another pageby adding a hash symbol and id valueto the end of the page URL.Let's look at an example in CodePen.For in-page links to work effectively,the page content needs to extendbeyond the browser's visible area.When clicked, the link will automatically scrollthe corresponding section to the top of the viewport.If the page isn't long enough,the browser will scroll as close as possibleto the target content.There are some longer blocks of code here in the HTML,so we need to scroll down to see all of the content.In most text editors, you can collapse indented codeby clicking on the arrow icon next to the line numbers.I'll collapse the p tags containing the answersand the extra sections of the content.The arrow now points to the right,indicating there's a collapsed block of code.By collapsing code blocks,it makes it easier to see where tags open and closeon pages with longer blocks of code.If we need to troubleshoot,we can now clearly see where tags open and closeand the href and id valueswithout scrolling up and down the page.Hyperlinks can also be used to link to specific actions.For example, the href attribute can be usedto link to an email address.When the link is selected,it will open the person's default email program.You can also link to a phone number.When selected, most phone numbers will auto dial the number.On a computer, if you have a program that can make callslike FaceTime or Skype,it will open in those applications.These special types of links require specific prefixes:mailto for email addressesand tel for phone numbers, both followed by a colon.For phone numbers,you can include just the area code and number.To add a country code,start with a plus sign followed by the country code.Links can also include additional attributesto add functionality to provide extra information.A few common attributes are target, download, and title.The target attribute is used to specifywhere the linked document will openwhen a user clicks on it.It's commonly used with the value \_blankto open an external websitein a new browser tab or windowdepending on the user's personal settings.This keeps your website open in the original windowrather than navigating away from it.The download attribute instructs the browserto treat a link as a downloadable filerather than opening it directly in the browser window.However, the behavior will varydepending on the browser and user settings.When a user clicks a download link,they might see a prompt before the download begins.The file might save automaticallyor it could open in an external appor within the browser itself.The download attribute works without a value,but if you want to specify a different file namefor the downloaded file,you can add a value to the attribute.For example, your organization's internal files might followa specific naming conventions,but you want your customers to download the filewith a more user-friendly file name.The title attribute can be usedto add supporting information, such as a short description.By default, this text will appear as a tooltipwhen users hover over the link.Just note that peopleusing keyboard navigation or touchscreenscan't access the title informationsince the tooltip only displays on mouse hover,so make sure essential information is visiblein the link text itself.Understanding how to use different types of linkscan help us make better websitesand using clear and descriptive link text ensuresthat our websites will work well for all usersno matter how they access the internet.

### Navigation

- [Instructor] The nav element represents navigation links.Like the header,this content usually stays consistent across webpagesto help users navigate the website more easily.A main menu navigation is typically added in the header.The navigation can be placed beforeor after other header content,but positioning it first improves accessibilitysince users will be able to accessthe navigation options right away.It can also be included in the main sectionor article element,if you have a navigation that's relevantto the page content,A nav may also be included in the footer sectionfor secondary navigation links,like policies or terms of service.Let's look at the structure for three common nav examples,a basic menu, a navigation with a main and submenu,and a breadcrumb navigation.The first list shows a basic main menu,which links to three pages,the homepage, portfolio page, and contact page.If you had a one-page website,you could still use the same structure,but with in-page linksto direct users to different sections on the page.The second list is an example of a navigation menuthat also contains a submenuby nesting another list within the list item.The main navigation contains two links,apparel and accessories.The submenu is nested inside the apparel list item.The third list is an exampleof a breadcrumb-style navigation.For this type of navigation,ordered lists are used because, in this case,order does matter.The first list item represents the top level link, apparel.The second represents the subitem, shoes.The current item, athletic, doesn't require a linkbecause this represents the pagethat you are currently viewing.I've added CSS to remove the default list styles,so we can view the navigationcloser to how it's commonly styled on websitesto give us a little more context.While you can do a lot more with CSSto change the look of navigation lists,make sure that it always starts with organizedand semantic HTML.

### Challenge: Add links and navigation

(upbeat music)- [Instructor] In this challenge, we'll update our projectby adding a new page for contact information.We'll also create a site navigation menu.Here are the steps to follow.Create a new file and name it "contact.html."Then add the required HTML tagsand the header and footer content,which will be the same as the other pages.Step three, update the page titleto reflect the content page.Then add a site navigation menuto link all three pages together.And lastly, add a list of ways to contact youor places you can be found online,such as your social media accounts,a newsletter, or YouTube channel.You can also add a sentencebefore the list to describe the content.Take some time now to work it out on your ownor proceed to the next video for the solution.

### Solution: Demo

(upbeat music)- [Christina] Let's go through the solution,starting with creating a new page.In the Explorer sidebar, click the file icon.Let's name this file contact.html,and make sure it's added to your personal site folder.Then we'll add the basic tags.Exclamation point, tab.Next, I'll copy the header and footerfrom one of the other pagesand paste it into the contact page.For the page title,I'll copy it from the other page as well.I'll update the beginning part to Contact first,then paste in the rest of it.For the site navigation, start by addingthe <nav> element as the first blockof the content in the header.For the navigation links, use an unordered list,starting with a <ul> tag.Then add the <li> tag.Inside the list item, add an anchor tag for the link.I'll duplicate this line twiceto create links for all three pages.The first link will go to the homepage.The href value will be index.html.All the HTML files are in the same directory,so we only need to use the file name.The link text to be addedbetween the anchor tags will be Home.The second href value will be blog.html.For the link text, I'll just use Blog.And for the third href, we'll link it to the contact page.And for the link text, I'll just use Contact.Now, let's copy this whole navigation blockand add it to the other pages.Make sure to save the changes on each pageso that we can test it out in the browser.Now we can easily navigate between the three pages.Let's leave it on the contact pagesince we'll be adding more content here.For the main content area, add list of ways to contact you.Start by adding the <main> tag.Before creating the list,I'll add some introductory text.Since only <li> tags can be used in a list,I'll use a paragraph tag.Here are places you can find me or support my work.For the list items, order doesn't matter,so I'll use an unordered list again.I'll be adding four list items,so I'll start by adding the markup first.<li> for the list item,<a> for the link.For the content, I'll be adding linksto my YouTube, online shop,newsletter, and LinkedIn profile.Just so you don't have to watch me type it all out,I'm going to copy the content from a pre-written text file.You can also just pause the video hereto take a moment to add in your own links.Since these links go to external websites,we should set them to open in a new window or tab.This keeps users on your sitewhile letting them explore the external links.Add the target attribute with the valueof \_blank to each link.If it makes it easier for you to read,you can also just add some line breaksto make it easier to see where the <li>'s open and close.Don't forget to add the target attributeto each link as well.And that's it. That's the end of this challenge.

## 04\_07 Solution: Text

Here is the final solution. For a further breakdown, continue reading below.

file: contact.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Contact | Your Name, Optional Title or Tagline</title>

</head>

<body>

<header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

<hgroup>

<h1>Your Name</h1>

<p>Title or tagline goes here</p>

</hgroup>

</header>

<main>

<p>Introduce your list:</p>

<ul>

<li><a href="/add/url" target="\_blank">LinkedIn</a></li>

<li><a href="/add/url" target="\_blank">Newsletter</a></li>

<li><a href="/add/url" target="\_blank">YouTube</a></li>

<li><a href="/add/url" target="\_blank">Social Media</a></li>

</ul>

</main>

<footer>

<p>© 2025 by Your Name.</p>

</footer>

</body>

</html>

The new contact.html page should have the same <header> and <footer> content as the index.html and blog.html page but with an updated <title> to reflect the page content.

<title>Contact | Your Name, Optional Title or Tagline</title>

The same site navigation should be added to all three pages. This will allow the user to be able to click the links to navigate the site. The navigation can be placed before or after other header content, but positioning it first improves accessibility since users can access the navigation options right away.

<header>

<nav>

<ul>

<li><a href="index.html">Home</a></li>

<li><a href="blog.html">Blog</a></li>

<li><a href="contact.html">Contact</a></li>

</ul>

</nav>

<hgroup>

<h1>Your Name</h1>

<p>Title or tagline goes here</p>

</hgroup>

</header>

For the main content area, the items in the contact list are not required to be in a specific order so an unordered list can be used. To include introductory text, use a <p> tag, since only <li> tags can be used in an HTML list. The target attribute should also be used to open external links in a new window or tab. That will prevent users from navigating away from your site when they click a link.

<main>

<p>Introduce your list:</p>

<ul>

<li><a href="/add/url" target="\_blank">LinkedIn</a></li>

<li><a href="/add/url" target="\_blank">Newsletter</a></li>

<li><a href="/add/url" target="\_blank">YouTube</a></li>

<li><a href="/add/url" target="\_blank">Social Media</a></li>

</ul>

</main>

5 images

### Image formats

- [Instructor] Before we talk abouthow to add images with HTML,let's take a look at the different image formatsused on the web.These formats fall into two categories,raster and vector images.Raster image files are made up of pixels,which are basically tiny squares.Each pixel has its own color,and together they form the image.The more pixels an image has,the higher its quality and detail.Digital photographs are usually raster filesand are commonly used for editing photos and graphics.Vector images use mathematical equations to define lines,curves, and shapes instead of pixels.Because of this,they can be scaled up or down without losing quality.Vector files work best for digital illustrations,complex graphics, and logos,but they're not suitable for photographs.All web image formats will be discussing in this videoare raster types, except SVG, which is a vector type.Digital image files are always squareor rectangular in shape.To create the illusion of other shapes,transparent backgrounds are used.The JPEG format displays over 16 million colors,making it ideal for photographs.It supports rich color variations, tones, gradients,and lots of detail.Both .jpeg and .jpg can be used as the file extension.JPEGs do not support transparency.The GIF format only supports 256 colors,so it works best for images with minimal color variation.GIFs can also be animated,which is one of their most popular uses.But since they support a limited color palette,photographic animations may appear grainy or pixelated.However, this color limitation generally resultsin smaller file sizes.GIFs also support transparency,so this format can be used for imagesthat require a transparent background.The PNG format can display millions of colors,so it works well for photosand images with lots of color variation.It also supports transparency,so if you need a high quality image with transparency,PNG is a better option than the GIF format.The file size of PNGs may vary.Photos are sometimes larger than JPEGs,but simple graphics with basic colorsand shapes often create smaller files.WebP is a newer image format developed by Googleand supports photographic images, animation,and transparency.The quality level is pretty much the same as JPEGs and PNGs.It also supports transparency.Since WebP images were specifically designed for the web,their compression techniques produce smaller files.Of all the image formats used for the web,SVGs are the only vector-based option.They use mathematical expressionsto define shapes rather than pixels.SVGs don't have support for photographs,so they work best for two dimensional graphics.Their file sizes tend to be smaller,especially for graphics with simple shapes and colors.SVGs are based on XML,which is a markup language similar to HTML.This means you can also create SVG imagesby writing your own code.Let's take a look at an example in CodePen.Basic shapes are created with rectangle and circle elementswhile more advanced shapes can be createdwith the path element.There are various attributes used to position the elements,add colors, change their sizes, and more.When creating SVGs with code,you can treat them like HTML elements.They can be styled with CSS and enhanced with JavaScript.You can also open any SVG files in a text editorto view its underlying code.There are quite a few SVG elements and attributes,so it can get quite extensive,but coding them isn't the only option.SVGs can also be created using vector graphic softwares,such as Illustrator or Affinity Designer.Different image formats can be used in the same projectdepending on the specific type of image and your needs.JPEGs and PNGs are common starting points,but it's good to know what other options are available.

### Embedding images

- So far, we've gone over different waysto format text-based content,but images can also be used as content.For example, shopping sites use them to showcase products,while personal websites might feature a profile pictureor thumbnails for your projects.Images can also be used as design elementsto add visual interests,such as a banner image at the top of a page.To add images to an HTML page, use the image tag.It's a void element, so a closing tag is not required.The source attribute must be includedto point to the location of the image file.A relative URL points to the fileswithin your project directory.An absolute URL points to an external resource.The syntax for relative and absolute URLswork the same whether you're creating hyperlinksor embedding web resources.When building a website,sometimes you don't haveall the content you need right away.Like text generators,many online placeholder image services, like Lorem Picsum,provide temporary imagesthat you can add to your page for context.Since these images are hosted on their servers,they must be included using an absolute URL.Following the instructions on their website,just add their URL with a width and height value at the endto generate a random image of that size.They also have a variety of optionsfor choosing images from their library.You can choose a specific image by adding an ID.You can also use the same random image every time.There are also gray scale and blur options,and some advanced usage instructionsto combine all of these options.Another option is Placeholder.Instead of displaying a photo,this placeholder surfacecan be used to display the dimensionsof the image to be added to that part of the page.This can be useful for templates, wire frames,or design guides,so you know what size the image needs to bewhen it's ready to be added to the site.This service has some formatting options as well.The basic usage is the same as Lorem Picsum.Just use their URL, plus the dimensions.With Placehold, you can also specify the image format,background and text colors,add your own text, font examples, or create retina images.These services are meant to be usedas temporary placeholders,so you should replace these images eventually,but also note that if the service goes down,the images will no longer be available.An alt attribute should also be included in the image tagto provide alternative texts for screen readers.This text should be a brief description of the image,and it's a simple way to make your website more accessible.The alt value will also appear in the browserwhen an image fails to load,usually with a broken image icon.This may happen when images are blocked,especially in emails,or when the source URL is incorrectdue to a typo or wrong file path.This helps users understandwhat images should have been displayed.It's best to have as much content ready as possiblebefore building your website.Sometimes you may design a beautiful pagewith placeholder content,only to find out that your actual contentdoesn't fit as expected,but it's not always possible,so if you do use placeholder content,try to use something that matches your intended contentas closely as possible.If you don't have your own images to use,you can find high-quality photosfrom a variety of stock photography sites.Some services are free,while others require a one-time paymentor a subscription fee.You can find a list of recommendationsin the Links PDF file.Here's a little pro tip.Even if an image is free,be sure to check the copyright,usage and attribution requirements.It's usually found in a text file with your download,or in the website's usage guidelines.

### Width and height attributes

- [Instructor] Now that we've covered the basic requirementsand syntax for the image take, let's explorehow to use additional attributesto enhance its functionality.This example file 05\_03\_startis located in the chapter five 05\_03 folder.I'll be using the same filefor all the demos in this chapter.There are also various image files included in this folder,which will be used throughout the chapter demonstrations.Each demo's completed code is saved in its own folder,but these folders only include the HTML file,not the images.You can use these filesto view the completed code off screen.We're using VS Code for these examplessince CodePen's free account doesn't allowfor the uploading of your own image files,and we'll be using quite a few.I'm also going to zoom my browser out a bitand collapse the explorer sidebarso we can see this full image while keeping the editorand browser both in view.Okay, now that we've got the file situation sorted out,let's get back to working with images.When using the image tag with the src attribute,the embedded image displays at the same dimensionsas the file itself.The file used in this example is 800 by 500 pixels,so it will be displayed in the browserwith the same dimensions.Also, just note that there's a bitof CSS in the head section.I've added some basic styles to preparefor a later demonstration,but it doesn't affect how the images are currently displayedother than the fact that it's center aligned.I've already opened my filein the browser using Live Server.Also note that I've added the dimensions to the imagesand file names to make it easierto see which file we're currently working with.But you can also find the dimensionsin the information panelwhen selecting a file in Finder or Explorer.Even when an image is already displayedat the size you want, the width and height attributescan be added to an image tag.The values are defined with numbers only,but are interpreted as pixels.I'll use the same values as the dimensions of the image.When using multiple attributes in the same take,the order doesn't matter as long as they're separatedby a single space.The browser ignores extra white spaceor line breaks, so when you're using multiple attributes,you may want to put each attributeor related group on its own line to make it easier to read.If we take a look at the update in the browser,it didn't result in any visible changesto our current example, but the widthand height attributes can be useful for image loading.Images initially take up no space on the page,so when they finish loading, they appearto jump into position, pushing the surrounding contentand creating layout shifts.This is more noticeable when using larger image filesor if the page is loading on a slow connection.Adding the width and height attributes allows the browserto reserve the specified amount of spacebefore images load, preventing layout shifts.It's not required, but it can improve the user experience.When you set either the width or heightto a value different from the original file dimensions,the browser will display the image at that specified size.Let's change the width to 400.Then I'll save the file to see the updates.Now the image appears distorted.That's because these dimensionsdon't have the same aspect ratio as the original file.The aspect ratio is the proportional relationshipbetween width and height.It's usually written as two numbers separated by a colon,like 16:9 or 4:3.When changing the size of an image,the ratio must be consistent.If you use only the width attribute,whether it sets the original sizeor a different value,the height will automatically adjust proportionally.The same is true if I were to only set the height value.The width will also resize automatically.Although we're using the 800 pixel wide image file,it now displays at a 400 pixel width.Since we've defined a size with the width attribute,this overrides the default display behavior.The height is also adjusted automaticallyto maintain the aspect ratioof the original file dimensions.We'll be using the same file for the remainderof the demonstrations in this chapter.So if you'd like to continue to follow along,leave it open in the editor and browser.

### Optimizing images

- There are many ways to resize images with HTML attributesand with CSS, but you'll often needto resize the actual image file first.Photos taken with a camera are typically optimizedfor print, making them much largerthan what's needed for the web.But how do you know what size to make the images?Let's go over some key factors to consider.If we look in the exercise folderand select the original file named dogs-original.png,we can see in the file informationthat the dimensions are just over 4,900 by 3,000 pixels,which is significantly largerthan the image file currently being used in the example.Photos can be modified using professional software,like Photoshop or Affinity Photo.There are also free alternatives, such as GIMP and Pixlr.How to use these tools will be discussed in more detail.For now, let's talk about a couple of thingsto consider when sizingand optimizing images for the web:resolution and pixel density.Resolution is the number of pixelsthat can be displayed on a screen,usually expressed as width by height.For example, a resolution of 1920 by 1080 meansthat the screen can display 1,920 pixels horizontallyand 1,080 pixels vertically.Resolution is also independent of a screen's physical size.Two screens with the same dimensionscan have different resolutions.The screen with the higher resolutionhas more pixels contained within the same space.This is called pixel densityand refers to the number of pixelsthat are packed into a screen.It's measured in pixels per inch or PPI for short.The higher the pixel density, the higher the resolution.While PPI varies among manufacturers,high pixel density screens typically haveat least twice the number of pixels per inch,both horizontally and verticallyas compared to standard screens.You may have heard of the term Retina,which is Apple's trademark termfor their high pixel density screens, but sometimes Retinaor Retina-like is used to describe similar displays.It's just easier to say than high-density pixel screen.When the PPI is higher, the pixels are smaller,which makes images and other graphics, iconsand fonts appear sharper and more detailed.When using images for websites, the files should be resizedto the smallest dimensions needed for the page layout.Since high pixel density screenshave roughly doubled the PPI of standard screens,it's best to use images with at least twice the resolutionof the intended display size.Then use HTML or CSS to resize the dimensionsbased on the screen size and resolution.Let's go back to the editor and update the HTML.In this example, we set the image display sizeto 400 by 250 pixels.Even though the height is automatically resized,I'll just add the attribute back in for clarity.To see the difference in quality,let's compare two file sizes.This one is already twice the size of the dimension setwith the width and height attributes.Let's add another image filethat matches these display dimensions.I'm going to copy the existing HTML code and paste it below.Then change the src value to point to the 400by 250 image file.Now both images are displayed in the browserand take up the same physical space.It might be hard to see in this video,but if you have the example open on your computer,you'll notice that the 800 by 500 pixel image is more clearand detailed since it has doubled the amountof pixels packed into the same amount of space.Normally, we wouldn't use width and height attributesto change the size of the image,but this was just for a quick demo.In the upcoming videos, we'll go over more HTML techniquesfor handling images across various screen sizesand pixel densities.

### Developer tools and resizing images

- To create responsive images for your website,you'll need multiple versionsof each image file at different sizes.And a way to test them across different screen sizes.In this demo, we'll take a look at two types of tools,Pixlr, an online editing tool for resizing imagesand browser developer toolsfor testing different screen resolutionsand pixel densities.Developer tools across browserswill have similar functionality,but their specific features and interface may vary.For this demo, I'll be using Firefox's developer tools.To access the Firefox dev tools,right click anywhere on any webpage and select inspect.By default, the tool will open on the bottom of the browser.You can also use a keyboard shortcut Commandoption I on a Mac or Ctrl Shift I on a PC.Once it's open, click the iconthat looks like a phone and a tablet.This is located on the top right of the menu barand it will open the responsive design mode.We won't be using any of the tools down here in the console,so click the X icon to close it.In the options at the top,the responsive dropdown contains a listof specific devices that can be emulated.Selecting an option will resize the test areato the same dimensions as the device.You can also choose a specific resolutionby updating the numbers in the input boxes.Another option is to hover over the right bottomor bottom right edge of the viewport areato manually drag the edges to resize it.The phone icon is used to rotate the viewport.The DPR dropdown contains the device pixel ratio emulator.This lets us test how images will load on a standard displaywith the DPR set to one or high density displayswith the DPR set to two or three.Emulator tools may not be 100% accuratebut they're helpful when you don't have accessto all these different types of devices.Let's keep the developer tools open.We'll continue to use this fileand this tool throughout the restof the chapter demonstrations.Now let's explore how to resize image files.If you're already familiar with image editing software,feel free to skip ahead to the next video.Pixlr is a photo editor tool,and while it isn't as full featuredas other professional software,the basic functions work in the same way.Also, resizing images is pretty basic,so for this purpose, it's a good free optionand it's available in the browserso there's no download needed.I'll start by choosing the pixel editor option.From here, choose open image to select a file to resize.You can choose any image filesince this is just for demonstration.Once the image is added to the workspace,go to the image menu.We'll see two common options,image size and canvas size.The canvas usually refers to the areawhere you can create and edit your imagesand we don't want to change that,so let's choose image size instead.If you want to resize the image proportionally,then just change the width or height.That way the other value will automatically resizeto maintain the aspect ratio.Once you modify the size, select apply,then go to file, export, quick export image as PNG.This file will then be saved to your computer.More advanced editing tools will allow youto export your files into different image formats,but PNG is fine for most cases.And that's it, that's how you resize an image file.Depending on the project, you may need to createa bunch of different versions of the same image.Web design and development can be fun and interestingbut it also involves some repetitive tasks.That's just the nature of the business.If you are using a content management system like WordPressor Joomla, there are plugins that can be addedto automatically resize the images when you upload them.Website builders like Squarespaceoften integrate this feature into their platform as well,though they often still have formatting guidelines.For the project challenges, you can use your own photosor stock photography.If you do, you'll likely need to resize the image files.You can also use the images from the demos as well,but just keep in mind, these are just placeholdersand should only be used for the course exercises.

### Responsive images, part 1

- In the early days of the internet,most people viewed websites on desktopsor laptop computerswith little variation in screen sizes and resolutions.Today, we have a wide variety of internet-enabled devices,from phones and tablets to laptops,desktops, TVs, and even refrigerators.In 2010, designer Ethan Marcotteintroduced Responsive Web Design to address these changes.This design philosophy focuses on creating websitesthat are able to adaptto different screen sizes and devices.Responsive images are an important partof responsive web design.There are several HTML methods for creating adaptiveand responsive images,each designed for specific scenarios.If you're limited to only using a single image file,then using one that's at least twice the sizeof the intended display dimensionswill support both standard and high-pixel density screens.While this technique works,and it's a valid option,the standard screen doesn't actually needto display the larger file.Instead of referencing just one file,the source set attribute can be added to an image elementto list multiple image files.The browser can then choose from the listand load the most appropriate versionbased on specific conditions, like pixel density.With srcset, you can have one versionfor standard screensand another for high-pixel density screens.Let's go back to the HTML file.With the srcset attribute,we can create a list of image filesfor the browser to choose from.However, the basic image attributesshould still be included.The source attribute serves as a fallback optionfor older browsersthat may not support these newer features.To maintain functionality while minimizing page load,let's use the 400x250 file as the fallbackand remove the image tag with the 800-pixel-wide file.The width and height attributes are optional,but we can leave it for now.Now let's add the source set attribute.I'll put it before src.In VS Code,when you type an attribute or file name,it will show a list of matching options.You can tab through the options and press enter or returnor tab to use the auto-complete feature.Next, we'll define the list of images.The first one will be the dogs-400x250 image filefor standard screens.Don't forget to include the file extension, .png.Then we'll add a comma to include a second file,the 800x500 pixel file.A pixel density descriptor is added after each image file,but if you leave it out,the browser assumes it's for a 1x standard screen,so we can leave the first one blankand add a descriptor after the second image.This is for displays with twice the PPI.Add a space after the file name, then 2x.Save the fileso we can test it with the browser tools.When the DPR is set to 1,we'll see the 400x250 imagesince this image is designated for standard displaysas listed in the srcset attribute.When the DPR is set to 2,we'll see the 800x500 pixel imagebecause now we're emulating a 2x screen.The browser has selected the image file markedwith the 2x pixel density descriptor.If we select DPR 3,the browser will still use the 800x500 imagesince it's the highest resolution option available.With just the src attribute,we were only able to use one image file.But with srcset and a pixel density descriptor,we can use multiple filesand allow the browser to display the most appropriate onefor the user's screen.With this technique,the display dimensions remain the sameacross different screen sizes,but we can swap out the image filebased on pixel density.Standard screens will show the low-resolution imagewhile high density screens show the high-resolution version.

### Responsive images, part 2

- [Instructor] In the last demo,we used the source set attribute to swap image filesbased on the pixel density of the screen.With this technique, the images display sizestays the same regardless of the browser viewport width.You can also use source set to changeboth the image file and display sizeand have the change be basedon the browser's viewport width.This approach requires using a width descriptorand the sizes attribute.A width descriptor specifies the width of the image file,instead of designating a pixel density.It's measured in pixels, but uses a W unit.The sizes attribute provides guidelinesfor browsers to choose the most suitable image filebased on a media condition and display size.The first part of the value sets the condition,which describes a characteristicof a device like the viewport width.The second part of the value specifies the display size.If that sounds confusing, I totally get it.It took me a little time to wrap my head around it too.So before we update the example,let's go over what we're trying to accomplish, step by step.The browser matches the viewport widthto the closest available image widthspecified in the source set value.It will choose an image file that is equal toor less than the viewport width.For example, when the viewport width is 400 pixels or less,it will display the 400 pixel wide image.If the viewport is wider, even just by one pixel,the browser will select the next closest image size.But what if we wanted to use the 400 pixel imageat wider viewport widths?The sizes attribute handles this part.The media conditions are used to set break points.Instead of switching to a larger image immediatelyafter 400 pixels,we can choose where to make the switch, like at 600 pixels.We can add as many break points as needed.In this example, there's also a third value,which is just the display size.This specifies the image width for viewportsthat fall outside of the media conditions.Let's go back to the editor and update the HTML.I'll keep this image with the pixel density descriptor,to use for comparison.For the new image, I'll copy the block and paste it below,then add the width descriptors.For the first image, the width descriptor will be 400W.Second image will be 800W instead of 2X.We'll also add a third image for larger viewports.Add a comma, then the file name.I'll use the dogs-1600x1000 pixel file.Then I'll add a space and the width descriptor, 1600W.Next, we'll add the sizes attribute.I'll put it after source set.The first part of the value is the media condition.Add a pair of parentheses.Then add the condition, max width: 600 pixels.The space after the colon is optional.Outside of the condition, add a space.This one is required.Then we'll add the display size.I'll use 400 pixels to make sure it will selectthat image file.Then we'll add a comma to add a second condition.We'll use max width again,but this time with a value of 1,000 pixels.Then add a space and 800 for the display size.Let's add one more commaand just the display size, 1600 pixels.This will be for the viewportsthat are wider than 1,000 pixels.One last thing before we check the updates,remove the width and height attributesto avoid conflicts since we're setting the display widthusing the sizes attributes instead.Okay, now let's save the fileand see how it looks in the browser.Let's switch back to DPR-1.When the viewport width is 600 pixels or less,both image takes will display the 400 pixel file.But if the screen is wider than 600 pixels,the first image remains the same.But when using the width descriptor in sizes attribute,the display and image file has changedbased on the conditions.If the viewport width is smaller than the widthof the display size of the image,you'll have to scroll to see the full image.On the other hand, if the viewport is largerthan the width of the display size,there will be extra space.Let's see what happens when we change the widthsin the sizes attribute.For the first condition, changing the display widthto anything larger would cause the browserto skip the 400 width imageand look for the next larger one.We don't want that to happen, so I'll put it back to 400.For the second condition,since we're using an 800 pixel image,we can set the size to any value between 401 and 800 pixels.I'll choose 600.Now, when the viewport is between 601 and 1,000 pixels,the browser will still choose the 800 pixel image,but display it at 600 pixels.Though this technique uses widths,it still takes pixel density into account.At DPR-1, the 800 pixel image is selected.At DPR-2, the 1,600 pixel image is selected.Though we can change the widths in the sizes attributeto control how it displays in the browser,it's limited because it's a fixed width.It's more effective to use this techniquefor swapping the image files.Then use CSS to change how it's displayed in the browser.In the head section of our HTML file,there are two CSS blocks.The first block just centers the imageand removes some extra spacing around the viewport.This is what it would look like without this CSS.The second block contains styles that will set the widthof the image to 100% of the viewportand the height to auto to maintain the aspect ratio.It's using what is called a class selector.To apply this style,we'll add a class attribute to the image tag.I'll add it as the last attribute,class = responsive.Now let's set the DPR to one to seehow it cycles through the images.Adding CSS only changed how the image displaysby setting the width to span 100% of the viewport.The conditions for the image switching stayed the samewith the display width helpingto determine which image file to userather than how wide to display it.While we generally avoid sizing up small images,they can be upscaled a bit with minimal quality loss,which can save some bandwidth.This is a great example of using HTMLfor structuring contentand CSS to style the content.

### Responsive images, part 2

- [Instructor] In the last demo,we used the source set attribute to swap image filesbased on the pixel density of the screen.With this technique, the images display sizestays the same regardless of the browser viewport width.You can also use source set to changeboth the image file and display sizeand have the change be basedon the browser's viewport width.This approach requires using a width descriptorand the sizes attribute.A width descriptor specifies the width of the image file,instead of designating a pixel density.It's measured in pixels, but uses a W unit.The sizes attribute provides guidelinesfor browsers to choose the most suitable image filebased on a media condition and display size.The first part of the value sets the condition,which describes a characteristicof a device like the viewport width.The second part of the value specifies the display size.If that sounds confusing, I totally get it.It took me a little time to wrap my head around it too.So before we update the example,let's go over what we're trying to accomplish, step by step.The browser matches the viewport widthto the closest available image widthspecified in the source set value.It will choose an image file that is equal toor less than the viewport width.For example, when the viewport width is 400 pixels or less,it will display the 400 pixel wide image.If the viewport is wider, even just by one pixel,the browser will select the next closest image size.But what if we wanted to use the 400 pixel imageat wider viewport widths?The sizes attribute handles this part.The media conditions are used to set break points.Instead of switching to a larger image immediatelyafter 400 pixels,we can choose where to make the switch, like at 600 pixels.We can add as many break points as needed.In this example, there's also a third value,which is just the display size.This specifies the image width for viewportsthat fall outside of the media conditions.Let's go back to the editor and update the HTML.I'll keep this image with the pixel density descriptor,to use for comparison.For the new image, I'll copy the block and paste it below,then add the width descriptors.For the first image, the width descriptor will be 400W.Second image will be 800W instead of 2X.We'll also add a third image for larger viewports.Add a comma, then the file name.I'll use the dogs-1600x1000 pixel file.Then I'll add a space and the width descriptor, 1600W.Next, we'll add the sizes attribute.I'll put it after source set.The first part of the value is the media condition.Add a pair of parentheses.Then add the condition, max width: 600 pixels.The space after the colon is optional.Outside of the condition, add a space.This one is required.Then we'll add the display size.I'll use 400 pixels to make sure it will selectthat image file.Then we'll add a comma to add a second condition.We'll use max width again,but this time with a value of 1,000 pixels.Then add a space and 800 for the display size.Let's add one more commaand just the display size, 1600 pixels.This will be for the viewportsthat are wider than 1,000 pixels.One last thing before we check the updates,remove the width and height attributesto avoid conflicts since we're setting the display widthusing the sizes attributes instead.Okay, now let's save the fileand see how it looks in the browser.Let's switch back to DPR-1.When the viewport width is 600 pixels or less,both image takes will display the 400 pixel file.But if the screen is wider than 600 pixels,the first image remains the same.But when using the width descriptor in sizes attribute,the display and image file has changedbased on the conditions.If the viewport width is smaller than the widthof the display size of the image,you'll have to scroll to see the full image.On the other hand, if the viewport is largerthan the width of the display size,there will be extra space.Let's see what happens when we change the widthsin the sizes attribute.For the first condition, changing the display widthto anything larger would cause the browserto skip the 400 width imageand look for the next larger one.We don't want that to happen, so I'll put it back to 400.For the second condition,since we're using an 800 pixel image,we can set the size to any value between 401 and 800 pixels.I'll choose 600.Now, when the viewport is between 601 and 1,000 pixels,the browser will still choose the 800 pixel image,but display it at 600 pixels.Though this technique uses widths,it still takes pixel density into account.At DPR-1, the 800 pixel image is selected.At DPR-2, the 1,600 pixel image is selected.Though we can change the widths in the sizes attributeto control how it displays in the browser,it's limited because it's a fixed width.It's more effective to use this techniquefor swapping the image files.Then use CSS to change how it's displayed in the browser.In the head section of our HTML file,there are two CSS blocks.The first block just centers the imageand removes some extra spacing around the viewport.This is what it would look like without this CSS.The second block contains styles that will set the widthof the image to 100% of the viewportand the height to auto to maintain the aspect ratio.It's using what is called a class selector.To apply this style,we'll add a class attribute to the image tag.I'll add it as the last attribute,class = responsive.Now let's set the DPR to one to seehow it cycles through the images.Adding CSS only changed how the image displaysby setting the width to span 100% of the viewport.The conditions for the image switching stayed the samewith the display width helpingto determine which image file to userather than how wide to display it.While we generally avoid sizing up small images,they can be upscaled a bit with minimal quality loss,which can save some bandwidth.This is a great example of using HTMLfor structuring contentand CSS to style the content.

### Challenge: Add responsive images

(upbeat music)- [Presenter] This challenge has only one task.Add an image to your homepage.It can be a profile pick of yourself,an image that represents you,or something that relates to the bio you've written.If you're new to photo editing,try following the steps from our earlier lessonto resize your image files.You can also use a placeholder image service, stock photo,or one of the sample images from our exercises.Just remember, the sample filesare for practice within the course only,not for external projects.Create an images folder for your image filesto group them separately from the HTML files.Use the responsive techniquefor supporting different resolutions and pixel densities.And here's a little bonus.Add additional links to your content.We created links for the navigation in your contact list,but if there are other places you can add linkslike the source of your quote or some content in the bio,feel free to add them now.When you're ready, proceedto the next video for the solution.

### Solution: Demo

(upbeat music)- [Instructor] For this challenge,I have two versions of the same image filethat I've resized for standardand high pixel density screens.I've already added it to my project folder,but even though there are only two files,I'll create an images folder to organize them.This time, I'll do it from Finder.Inside of the personal site folder,right click select new folder,and name it "images".Then, I'll move the files into this folder.Back in the editor,I'll add the image tag within the main element,just before the heading,starting with the basic requirements.When you use the auto complete shortcut for the image tag,it will automatically add the source and alt attribute.For the source value,I'll start with the folder name "images"followed by a forward slash,and use the smaller file beach dash 600 png.For the alt value,I'll add a description of the image,which is a beach along Lake Ontarioin the city of Scarborough called Scarborough Bluffs.Then I'll save the file and check my homepage.Now that the image has been added,let's update the image tag to add the responsive techniquefor displaying specific image filesbased on the pixel density of the screen.I'll add the source set attribute,starting with the 600 pixel image for standard screens.I'll start with the images folder again, forward slash,and use the same file name.Then I'll add a comma to add the second image.This one is the 1200 pixel wide png file.For the second image,I'll add a space and the two x descriptor.And now I'm going to save the file.High pixel density screens will now use the larger image,though both images will displayat a width of 600 pixels in the browser.The change may not be noticeable in the video,but I can see that my image is sharper.Now that it's loading the high res photo.For the bonus step,you can add additional links to your content as needed.For example, I can add a link to the citation in my quote.I'll start by copying the URLand then add the A tag around the article name.Put my URL in hereand move the closing tag to the end of the article name.I'll also add the target attribute with a value of blank.It's an external link,so I want it to open in a new tab or window.Other places I could add a link or referencesto external websites in my bio,such as my LinkedIn profile, courses, and YouTube channel.For demonstration purposes,I'll just use the number sign as a placeholder for the URL.And that's the end of this challenge.If you have any questions about how to useany of the other responsive image techniques,feel free to review the exercises from this chapter.

## 05\_11 Solution: Text

Here is the final solution. For a further breakdown, continue reading below.

file: index.html

<main>

<img srcset="images/beach-600.png, images/beach-1200.png 2x"

src="images/beach-600.png"

alt="Scarborough Bluffs beach on a cloudy day">

<h2>About me</h2>

<p>Lorem ipsum dolor sit amet consectetur adipisicing elit. Dolorum maiores aliquam, nihil repellendus in blanditiis, excepturi praesentium suscipit adipisci, repellat incidunt odit soluta cumque ipsum temporibus quos labore. Molestias, rem.</p>

...

</main>

This solution uses the responsive technique for displaying image files based on screen resolution. The browser selects the appropriate image based on the device's pixel density, while maintaining the same display size—in this case, 600px wide.

This requires two image files, one for standard screens (beach-600.png) and one for high pixel density screens that should be approximately twice the size or more (beach-1200.png).

The srcset attribute is used in addition to the src and alt attributes to list multiple image files. A 2x descriptor is added to the larger image to indicate to the browser that this file should be used for screens with a higher resolution and high DPI.

For the bonus step, use the anchor tag (<a>) with the target attribute to wrap any text that you want to link to an external source such as the quoted source.

<p>-Name of person being quoted, "<cite><a href="https://source/goes/here" target="\_blank">Title of work</a></cite>"</p>

And that’s it! If you have questions about how to use any of the other techniques, just review the exercises from this chapter.

6 more media elenet:

### Video

- The introduction of video and audio elementswas a big step forward in the development of HTML.The ability to easily embed videoand audio directly into webpages,replaced older plugin based solutionslike Flash and Silverlight,helping to drive the growthof streaming media online with platformslike YouTube and SoundCloud as some of the early adopters.Let's explore how to implement the video element,its key features and syntax.There are several file formatsthat can be used with the video element.MP4 is supported across all major browsersand offers good compression and quality.WebM is an open source formatthat delivers high quality videoand is optimized for web use.OGG is an older open source formatthat is still a viable option, but is less commonly used.MP4 provide the best browser compatibility,though any of these formats will work.You can use video editing softwareor conversion tools to switch between formats.All video elements start with an opening and closing tag.Just like the image element, the source attribute is usedto link to the media file.It's also recommendedto include a fallback message for older browsers.This typically includes a message and a download link,so users can watch the video in a mediaplayer outside of the browser.You can add additional supportby providing multiple video formatsusing the source element.It's a void element,and instead of adding the source attribute to the video tag,it's added to each source tag instead,along with a type attribute to specify the video format.Place these elements before the fallback message.The browser will use the first supported videoformat from the list.Let's see how a basic video element looks.Using the 0601 start HTML file.There's only one video file for the demonstration,so the source attribute is added directly to the video tag.To test your fallback message, just add a typoto the video tag so it won't render properly on the page.This message will only display in older browsersthat don't support the video element.Let's put it back to the correct syntax.Right now we can only see the video,but no player controls by default.Videos can be used to add visual interests.For example, a clothing store might use videoto showcase their products or for ad campaigns.In this scenario, you may not needto include any playback controls,but most of the time they're typically recommendedfor a better user experience.Back in the editor let's add the controls boolean attributeto display the default browser controls.The media player varies across browsers.Firefox provides play pause buttons, volume control,full screen mode, and a pop-out optionthat appears when hovering over the video.Chrome offers some additional features,playback speed and picture in picture,which is basically the sameas Firefox's pop-out feature.For more advanced customization beyond the default controls,or to create browser specific features,JavaScript can be used.Let's add a few more boolean attributes.Autoplaying can be used to set the videoto automatically start when the page loads.This feature would be usefulfor the clothing store scenario,but it's usually better to let users control playback.Just note that some browser settingscan be set to block autoplay,and in those cases,this attribute may not override those settings.Loop automatically repeats the video playbackafter it ends.And muted, starts the video with the sound muted.When muted, the audio icon is shown with a line through it.Click it to turn the volume onand click it again to mute it.These attributes serve specific purposes,but before using them, consider your goalsand how they will affect the user experience.To display a custom preview image for the video,use the poster attribute.This requires a value to specify the path to the image file,which is also included in the exercise folder.This image displaysbefore your video starts,so the autoplay attribute must be removed to be ableto see the preview image.Let's save the file.By default, the videos display area matchesthe files dimensions,but when using the poster attribute,some browsers will default to the sizeof the image file instead.For example, Firefox maintains the video dimensions,while Chrome defaults to the image size.Like the image tag, the width and height attributescan be used to define the size.You can also use CSS as well,but for now, let's use the attributesto set the width and heightto match the video file dimensions 640 by 360.There's no change in Firefox since it was already defaultedto the videos dimensions,but now we can see the change in Chrome.We've covered the basic usage of the video elementand how to add extra features using attributes,but there are more attributesthat can be used with the video element.The complete list is available in the MDN docs.Keep this example open.We'll be using it again in the next video.

### Adding captions and subtitles

- [Instructor] When using video and audio elements,additional content such as captions, subtitles,and chapters can be added using the <track> element.Captions are transcripts of the audio content.They're necessary for people with hearing impairmentsand helpful for various situations like a noisy coffee shopor in quiet settings like librarieswhere playing audio would disturb others.Subtitles are translated versions of the audio content,allowing viewers to understand the contentin different languages.Chapters are markers that divide the content into sections,allowing users to navigatethrough different parts of the media.The <track> element is also a void element,and it's nested within the video tag.Use the source attributeto specify the file location.Track files also use the WebVTT formatand have a VTT file extension.If you're using the source element,the track must be added after.Let's take a look at what's included in a VTT file.It starts with a file headerto indicate the format of the file, WebVTT.Next are the queue blocks, which consist of a time codeto specify when the text should appear.The format can be either hours,minutes, seconds, milliseconds,or minutes, seconds, milliseconds.The time units are separated by colons except milliseconds,which uses a period and three values.The start and end times are separated by two dashes,followed by an angle bracket with spaces before and after.Following the time code is the text that will be displayedas a subtitle or caption.An optional queue identifier can be added,usually a number, represented here with one, two, and three.It's added before the time code to serve as a referencefor JavaScript or CSS.A blank line should also be included between queue blocks.Let's go back to the example.Using the same file from the previous video,let's add a track tag.I'll add it before the fallback message.If you use the auto complete feature to add track,it will automatically add a closing tag,but this is a void element, so make sure to delete it.Then add the source attribute to link to the VTT file,which is included in the same folder.The file is now linked to the video,but the captions won't displayuntil we add the default Boolean attribute.This will enable the trackand show the contents of the VTT file in the video player.Now when you press play, the captions will be displayedto match the audio in the video.Let's go over some more attributesthat can be used with the track element.The kind attribute specifieshow the text track will be used.It can be set to one of these values,captions, subtitles, chapters,which we discussed earlier, or metadata.Metadata provides additional information about the mediasuch as descriptions or annotations.The chapters and metadata valuesdon't automatically display content on the screen.Instead, they can be accessed through JavaScriptto create features like chapter navigationor to display contextual informationwhen users hover over specific parts of the media.When using the kind attribute set to subtitles,the srclang attribute must be includedto specify the language.It uses the same language subtypes as the lang attribute.The srclang attribute is optional for the other kind values.We can add the kind and srclang attributes to the example,but they won't change how the captions appear.We're just providing additional informationabout the track to the browser.Kind should be set to captions and srclang to EN.One final note about the track element.While you can open HTML files directly from your computerinto the browser,the track element requires a server environmentto function properly due to security restrictionsand server communication requirements.However, the live server plugin we've been usingcreates a local server,allowing us to test VTT files with the track element.Without the server environment, the video will still play,but the track content will not be displayed.To learn more about the track elementand setting up a server,check out the extra resources in the linked PDF file.

### Audio

- [Instructor] The video element can be usedto play audio files,but the audio element is a better choice.It's specifically designed for audioand provides clear semantic meaning,making it more effectivefor both accessibility and search engines.Both elements share similar functionality,so the syntax is pretty much the samewith a few differences.Let's take a closer look.Like the video element,the audio element requires the source attributeto link to the audio file.The controls attribute must also be addedto display the player controls.Without it, the audio element won't display at all.Since it's audio only,the controls are the only visual interfacedisplayed in the browser.Also like the video element,audio should include fallback content as well.You can also include multiple audio formatsby using the source element nested inside the audio text.The source attribute is also added hereinstead of the audio tag.There are three primary audio formats that can be used.MP3 is the most widely supported formatand is compatible with all major browsers.It provides good audio qualitywhile maintaining relatively small file sizes.The WAV format also offers high-quality audio,but tends to have larger file sizes than MP3s.It has widespread browser support,though some legacy browserslike Internet Explorer don't support it.But IE usage is pretty low these days,so it's not that much of an issue anymore.The OGG format is an older open-source option,but also provides good quality and smaller file sizes.Let's look at an example using the 06\_03\_start HTML file.This shows the basic audio implementationusing the source and controls attributesalong with a fallback message.The styles of the default players vary across browsers,but they all have the same basic functions:play, pause, and volume control.Chrome has an additional feature: playback speed.The audio element supports the same attributesas the video elementexcept for the width, height, and poster attributessince audio has no visual component beyond the control bar.You can add the autoplay and loop attributesto automatically play the audio on page loadand loop the audio file when it reaches the end.But just like video, use these features thoughtfullyto ensure the best user experience.The track element can also be used with audio,but its implementation requires additional scriptingbecause the only visual interface is the control bar.Using the video element is the simplest solution,but if you want to use the audio elementfor better semantics,you'll need JavaScript to display captions during playback,either through existing librariesor creating your own custom scripts.

### Embedding media with iframes

- If you've ever used the embed option on YouTube, Spotify,or some other media platformto add a video, song, or playlist to a webpage,you've likely encountered the iframe element.It's an effective wayto display content from another source.With iframe, you can embed contentfrom one webpage into another webpage.Let's look at the structure for a basic iframe setup.The syntax should look familiar by nowsince it uses many of the same attributeswe've used for other media elements.An iframe requires an opening and closing tag.The source attribute points to the URLof the content you want to embed.The width and height attributesdetermine the size of the embedded content.And the title attributeprovides a text description of the embedded content,which is helpful for accessibility.Let's go to YouTube and take a lookat what their embed code looks like.You can look at any video,but I'll use the same one from the previous examples,except this time it's the version uploaded to YouTube.To find the code, click the Share button,then click the Embed option.Here we can see the basic markup we just covered,the width, height, source, and title,but there are quite a few more attributes includedto allow for additional controls, behaviors,or security settings.For the most part,it's fine to use a third-party's embed code as is,but understanding the basic structuremeans you can make your own edits,such as the width and the height.This is YouTube's default size,but now you know where you can change it.If you'd like to know moreabout some of these other attributes,it's all documented, of course.Check the Links PDF file for more resources.With iframe, you can embed any type of contentfrom another website,not just audio and video platforms.Let's say I have a websiteabout tourist attractions in Toronto.I can go to Google Maps and search for an addressor landmark to find its location.For example, let me look up the CN Tower.I can then embed a map of the CN Towerdirectly into my website,allowing visitors to view and interact with the mapwithout leaving the page.Go to Share, Embed a Map.The code will appear here.The preview shown in this modal windowrepresents what will be embedded on your website.There are many third-party servicesthat let you embed content,like forms, calendars, and interactive widgets,into your webpage.While knowing how to write HTML from scratch is valuable,using existing tools can help you add more advanced featuresand simplify your workflow.

### Challenge: Add media to project

(upbeat music)- [Presenter] In this challenge, let's add some media.This is an open-ended challenge.You can decide what media type to useand how you want to incorporate it.Here are some options.Use your own video or audio fileswith the video or audio elements,or embed media from a third party servicelike YouTube, Dailymotion, SoundCloud,or Spotify using the iframe element.For the solution walkthrough, I'll demonstrate howto embed a YouTube video into one of the blog posts.

### Solution: Demo

(upbeat music)- [Instructor] For this challenge,I'll add a YouTube video using the iframe elementto the first blog post.I've updated the post content to align with the contentof the video to make it look more realistic.You can use any video you'd like.It doesn't even have to be YouTube or a video.It can be Google Maps, SoundCloud, Spotify,or any third party media service.Finding the embed codefor most third party services is generally straightforward.It's usually under a share type link.Then look for an embed option.Copy the whole block and add it to your HTML.Back in the editor, I'll paste it underneath the paragraph.And let's see how it looks in the page.I could change the size by changing the widthand height values, but most of the time it's better to do itwith CSS, so I'll just leave it as is.And that's pretty much it.If you have any questions about how to useany of the other techniques,just review the exercises from this chapter.

## ****06\_07 Solution: Text****

Here is the final solution. For a further breakdown, continue reading below.

**file: blog.html**

<article>

<header>

<h2>How to use Emmet and VS Code</h2>

<p>Published: <time datetime="2022-12-28">28 Dec 2022</time></p>

</header>

<p>Learn how to use Emmet to streamline the process of writing HTML and CSS using different operators to create nested and sibling HTML elements, and how to specify CSS properties and values using an abbreviated syntax.</p>

<iframe width="560" height="315" src="<https://www.youtube.com/embed/nZp8cWskle0?si=OwQqnlEKQIocGGBI>" title="YouTube video player" frameborder="0" allow="accelerometer; autoplay; clipboard-write; encrypted-media; gyroscope; picture-in-picture; web-share" referrerpolicy="strict-origin-when-cross-origin" allowfullscreen></iframe>

</article>

This solution uses a YouTube video embed, which is added to the first blog post. The post content was updated as well to align with the content of the video.

Finding the embed code for most third-party services is generally straightforward. It’s usually under a “Share” link, which will lead to an “Embed” option. Copy the entire block and paste it into your webpage.

You can change the size by revising the width and height values, but most of the time it’s better to do it with CSS.

That's pretty much it! If you have questions about how to use any of the other techniques, just review the exercises from this chapter.

7 webforms

### What are web forms?

- Web forms are one of the main ways usersinteract with websites.When you fill out a quiz, sign up for a newsletter,or enter payment details to make an online purchase,you are using forms to input your data.Forms can be basic and contain a couple fields,like a name or email for newsletter subscriptions.They can also be complex, such as online shopping formsthat require mailing addresses,billing information, and payment details.A form's complexity depends on how much informationneeds to be collected.While most forms require processing on a web serverthrough server-side programming,the user interface elements,the parts that we see in the browser,like text fields and submit buttons,those are created with HTML.On web development teams, front end developerstypically create the interface elements,while backend developers manage the functionalityand the data processing.There are also many services for creating formssuch as Typeform or Google Forms.And while these tools have their place,there are benefits to learninghow to create forms with HTML.Some tools allow for customizationsand understanding how forms work can be useful,even if you're not coding them yourself.

### HTML form basics

- [Instructor] HTML forms contain various controls,which are the elements that allow users to input dataand interact with web forms,like text fields for typing,check boxes for selecting options,and buttons for submitting the forms.We'll explore how to implement various form controlsby building an account signup form.When designing user interfaces,I like to start by sketching it out.It provides a helpful blueprint,even if the final design may change.All forms begin with the <form> tag and two attributes.Action specifies where the data will be sent for processingwhen the form is submitted.The value must be a valid relative, or absolute URL.Method specifies how the form data will be sentto the web server and uses one of two values, GET or POST.The GET method sends data as part of the URL,making it visible in the address bar.This makes it useful for bookmarking or for sharing links.For example, if you're on a clothing store websiteand use the search bar to look for black pants,the data is added to the end of the URL.You can then bookmark it or share the link.The POST method sends data in the HTTP request body,which basically means it keeps the data hiddenand it's more secure.It can also handle sending large amounts of data,so this method is best for sensitive informationor for extensive forms.This is just a high level overview,since we won't be getting into form processing.For more information, check the extra resourcesin the Links.pdf file.Next, we'll need a way for users to add data.The <input> element is commonly usedbecause it supports many different types of data.We'll start by taking a look at the text-based inputs.A name attribute is also included to identifyand reference each piece of data when the form is submitted.Since this is only used for processing,the name value isn't visible in the browser.If we leave the form as is,users will only see blank input boxes.Each <input> needs to be paired with a <label>to display a caption.A <button> element is also needed for usersto submit the form for processing.The button text is added between the tags.For this form, we could use something like submitor more specific like register or create account.We'll cover how to add the checkboxfor the newsletter signup in a later video.For now, let's look at this markup in an example file.The 07\_02\_start.html file can be foundin the 07\_02 folder.I've created a submit.html fileincluded in the exercise folder as well,and it's referenced in the action attribute.While this won't process the form data,since that requires server-side programming,it will redirect to a form submitted pagethat we'll use for basic validation testing.Here's how the form currently looks in the browserwith default styles.They're all inline elements,so they appear side-by-side in a single row.This is a basic form structurethat's technically valid HTML,but we can enhance it with additional elementsand attributes to improve its structure and functionality.A form can be structuredusing any of the HTML tags we've discussed so far.For example, a heading tag can be used for the form title.For the <label>, <input> pairs,they're usually styled together,so lists, paragraphs or <div> elementsare commonly used to group them.Forms also have specific structural elementsthat can be used to add additional organization.The <fieldset> elementgroups related form controls together.The <legend> element is included within the <fieldset>and provides a caption or title.You'll still likely need to use a <div> or something similarto group the <label>, <input> pairs for styling,since the <fieldset> taggroups all the related pairs together.These elements could be used with our current form example,but since we only have two fields and a submit button,it's also fine to just leave it out.But there are situations where <fieldset>and <legend> is required.We'll get into that a little later.Let's go back to the example and update the structure.I'll use a <div> to group each <label>, <input> pair,since this is for presentation rather than for semantics.(keyboard tapping)I'll also use an <h2> for the form title,since websites generally reserve the <h1>for its main title.For the heading, I'll add Create an Account.By adding the <div>, the <label>, <input> pairsare now on their own line,but the default styles are quite minimal.Let's add some CSS to make it easier to viewand look more like a form you would see on a website.In the <head> section, un-comment the <link> tagto include the forms.css file.Now let's take a look at how the form works.When I click inside an input field, I can start typing.However, clicking on the label doesn't trigger any response.While the labels appear next to their inputsto connect them visually,we'll also need to establish this connection in the code.This will make the labels clickable,which improves usability,especially on small devices and touchscreens.One option is to place the <input> inside the <label>to create an implicit association.Another option is to create an explicit associationby using attributes.Let's update the example using this option.Add a for attribute to the <label> tagand use a value to describe the label.I'll go with email.Then add an ID attribute in the corresponding <input>,which I'll move to the next lineto make the code easier to read.The values of both the for and ID attributesmust be the same to create the association,so I'll use email again.Then I'll do the same for the password <label> and <input>.Start with the for, give it a value,then add an ID to the <input> using the same value.Using the implicit label association simplifies the markup,but an explicit association is generally preferredsince it creates clearer relationshipsand makes CSS styling more flexibleby keeping the elements separate.Now we can click either the input fieldor its label to focus on the associated input.In the next video, we'll go over input types in more detail.

### Input types

- [Narrator] Input elements are an important partof creating forms since they can handledifferent types of data.By adding the type attribute,we can specify each input's data type,which determines how the input functions.If a type attribute is not specified,the default data type is text,which is a single line text field.This would be appropriate to usefor something like a name or username.The email data type creates a single line text fieldas well, but it's used for inputs that are specificallyfor email addresses.Unlike the text type, which accepts any characters,the email type includes built in form validation.It doesn't verify that the email address actually exists,but it does check that the format matches eithername@domain or name@domain.tld.TLD refers to a top level domain,which is the part that comes after the dot.Dot.com is the most common,but there are many other standard TLDs such as .org or.edu.You may also see unconventional TLDs, like .me or .fyi.For password inputs, use the password type.It also looks like a single line text field,but the value is obscured as you begin typingto provide some security.if someone is peeking over your shoulder.The characters will be displayed as an asterisk or a dot,depending on the browser and operating system.You can also use the input tag as a submit buttonby setting the type to submit.The default text is also submit,but you can change it with the value attribute.When using the input element with the submit type,you can only modify the button textwith the value attribute.With the button element,the content is placed between the tags.You can use just textor combine it with other elements like an icon imagethat you can put next to your button text.Both approaches are valid,but the button element allows for more flexibility.Let's update the example with the type attributes.For the email input, add type equals email.For the password input, we'll use the password type.It's okay if the IDand name attributes also have the same valuesince they're used for different purposes.Submit is the default behaviorfor button elements used within a form,but we should still explicitly declare itby adding type equals submit in the opening tag.Adding the type attributes won't change how the form looks,but it will change how it functions.Let's try filling out the form.For the email, I'm going to purposely fill it outusing the wrong format.Hello, space, email.com.For the password, as I type the data,we can see that the text is obscured.Now I'll try to submit the form.Since the email address doesn't follow the proper format,an error message appearsand the form won't submit until it's corrected,so let's go back and fix it.I'll put the at symbol in and try to submit it again.Now that it has passed the email validation,the form will submit and take us to the success page.Click the back button to get back to the form.There's one more thing I want to demonstrate.If I leave all the fields blankand then try to submit the form,it will pass validation.By default, a blank input is also accepted,no matter the type.In the next video, we'll talk about howto add additional form validation.

### Form validation

- In the last demo,we saw that forms accept blank fields by default,since they often include both required and optional fields.For example, an address form will require a street address,city, country, and postal or ZIP code,while a unit number is optional;it's only needed for those who live in apartments.The required Boolean attribute lets you definewhich fields must be filled outand which are optional and can be left blank.The placeholder attribute doesn't validate data,but it does provide userswith a reference for the expected format or content.The attribute value appearsinside the input field and usually includesexample text or brief instructions.Let's add these attributes to the example.For this form, both inputs need to be filled out,so the required attribute should be added to each one.For the email input,I'll also use the placeholder attributeto include an example of the expected format.For the value, I'll add the example email@example.com.For the password input,I can also use the placeholder attribute here as wellto include some instructions like eight character minimum.After you save the file,you should see the placeholder text inside the input box,but it does disappear once you start typing.That's why it should never be usedas a substitute for a label.Labels should always stay visibleand be associated with the corresponding input elements.Now, when I try to submit a blank form,an error message will appearand the form will not submit.For the email type,by default, the form will accept an addresswith or without a TLD,so that means something like email@examplewill be considered valid.Also, let's try addinga password that's less than eight characters.Then submit the form.The data passed the validation.Because the placeholder attribute can only add hints,it doesn't actually run any checks on the data.Let's say you wanted to makethe email input match with the TLD.To add additional validation,the pattern attribute lets you specifya regular expression that the input values must match.Regular expressions are rulesthat define sequences of characters.In this example, these sequencescan be used to match to any upper or lowercase lettersor numbers, or just any upper or lowercase letters.Since the default email validation accepts addresseswithout a TLD, we can use the pattern attributeto require one.The pattern requires matching any letteror number, followed by an @ symbol, then a domain name,and finally a dot with a tld.To learn more about regular expressions,check the extra resources in the Links PDF file.There's also a text filein the 07\_04 exercise folder containing this pattern,so you don't have to worryabout trying to manually copy it from the video.For the password field,we can use the minlength attributeto set a number of characters required for the input.The value must be a non-negative number.Let's update the example.For the email input,the regular expressionused for the pattern value requires precise syntax.One missing symbol, and it won't work.So I suggest copying it from the text filein the 07\_04 Exercise folder.All right, let's add this to the email input,and then for the password input,we'll add the minlength attribute with a value of eight.Now, let's try adding an emailand password that does not follow the correct patternto see what will happen when we try to submit the form.The email has not passed the validation,which is what we want, so that's good.I'll add a .com to the address.Then submit it again.Now, the email validation has passed as expected,but I only used six characters for the password,and the minlength is set to eight,so I'll add two more charactersand try to submit the form again.Now it's passed all the requirements.Using just a few HTML attributes,we were able to add basic form validation,but for more advanced validation features,JavaScript or server-side programming will be needed.

### Checkbox and radio inputs

- So far, we've used the input element for text-based data,but inputs can also be usedto collect information using checkboxes and radio buttonswhere the user selects a choice rather than inputs text.A checkbox can be used to present one or more options.The user can check or uncheck individual optionsand select one or more options from the same group.It also needs a label to display a captionand the foreign ID attributes to create the association.Labels are usually placed after the checkbox.A radio button is used to present multiple optionswhere users can select only one choice.Once a radio button is selected, it can only be deselectedby choosing another radio button in the same group.It also requires a label.Let's go back to the example.The initial design included a single checkbox.Since we're already collecting an email address,the user can just check the boxif they want to opt in to the newsletter.Let's add a div just after the password grouping.Then we'll add the input and label inside.When using the shortcut for the input tag,it automatically adds the type to equal to text,so let's update that and change that to checkbox.Then I'll add the label underneath.For the for attribute value, I'll set it to newsletter,then add a corresponding ID to the input.Also remember, every input needs a name attributeto reference the data when the form is submitted,so let's add that too and give it a value of newsletter.Now, let's add some label content.I'll use, "Yes, send me weekly tips."You can add anything you'd like here.When using the checkbox type,we'll need to add another attribute, the value attribute,to include additional data when the form is submitted.This is added to the input element.In this example, we have name set to newsletter.If the value attribute isn't declared,the default value is on,so the data submitted will read newsletter=on.If we do add a value attributeand set it to something else like subscribe,the form will submit the name-value pairas newsletter=subscribe.I'm going to put these two values together on the same line.The checkbox can also be preselected by default.To do this, add the checked boolean attribute to the input.When you save the file,you should see the checkbox is now preselected.If the user does not want to sign up for the newsletter,they can uncheck the box before submitting the form.Now let's talk about how to use multiple checkboxes.We'll look at another example filein the 07\_05 folder, multiple-inputs.For our newsletter example, we can add optionsthat let users select which topicsthey'd like to receive emails about,coding, tools and software, or productivity.Since these options are related,they should be grouped using the field setand legend elements.The markup for the input and label elements is the sameas when using a single checkbox.The type is set to checkbox,the ID and for attributes are used to connectthe input and labels,the name attribute is used to reference the data,and the value attribute is added to specify each topic.When the form is submitted, the selected name-value pairswill be sent as topic=coding, topic=tools,and topic=productivity.Unlike checkboxes, a radio button can only be deselectedby choosing another option in the same group.That's why it's used for presenting multiple optionswhere only one choice can be selected at a time.For the newsletter, instead of sending weekly tips,we can allow the user to choose the frequency.The markup is pretty much the same as the checkbox example,except, of course, the type has to be set to radio.In this example, we've provided two options,weekly or monthly.One more thing to note.In the checkbox example,we used the same name value for the related options,but it wasn't required.But with radio buttons,the name values must be the same across related inputsto create a radio group.Once the group is defined, selecting any radio buttonautomatically deselects the previously selected button.Let's update our original example.While it's generally good to get practice writing code,in the interest of keeping this demo moving along,let's copy these two blocks from the multiple-inputs fileand add it to our original form.Copy the whole thingand add it underneath the newsletter signupand just before the button.I'll fix the formatting and remove these comments,then save the file.Now the user has the optionto choose one or more topics of interest for the newsletter.For the frequency, they can only choose one option.Since we're allowing the users to pick their frequency,I should probably update the signup wordingand remove the weekly part.Send me tips.This example was originally meant to be a formfor creating an account, so these extra newsletter optionsmight fit better in a dedicated newsletter signup form.But adding this to the demo gives us a chance to seehow all these elements can be used together.We've covered the basics of how to use various input types,but there are more input types and attributes available.Learning all the ins and outs of formscould really be its own course.The full list of inputs and attributesare available on the MDN website.

### Additional form elements

- [Instructor] Now that we've explored the many waysto use the input element,let's look at a couple more options, text area and select.The text area element is used for multi-line textand allows for larger blocks of freeform user inputs,like comments, reviews, or feedback.It only accepts one input type,so the type attribute is not required.A name attribute should be includedto reference the data when the form is submitted.The calls attribute can be usedto set the width using a numberbased on the average width of text characters.The rows attributedetermines the number of visible text rows.You can use these attributes, but they're not requiredsince CSS is typically used for more precise controlover the size of the text area.Dropdown controls are another way for usersto select from several optionswithout taking up too much space in the form layout.The select element is used to create a list of choices,while the option element is usedto add each item in the list.The value attributes are not required.It can be added when you want to send custom valueswhen the form is submitted.Without a value attribute,the content between the option tags is used as the value.Let's look at an example.Since these form elements don't quite fitwith our create account exampleand don't require any linked files,I've created a code pen demo insteadjust to view these form controls.The select dropdown is usefulbecause despite containing five options in the list,it only takes up the space of a single line of text.Unlike the text input, the text area allowsfor multiple lines of text, including line breaks.Both select and text area can also be usedwith labels to add a caption.There's a lot more to learn about forms.Selecting an option shows the documentationand all available attributes for each element.As mentioned in a previous lesson,forms could be an entire course on their own.But like everything we've discussed,you don't need to memorizeor know how to use every single element.Focus on learning the foundationsand keeping reliable resources handy.

### Challenge: Add a contact form

(upbeat music)- [Instructor] For this challenge,add a contact form to your contact page.The form should include introductory contentfor the form, like Contact meor Send a message, a field for the person's name,a field for the person's email,and an area for the person's message.And don't forget the Submit button.Don't worry about makingthe form functional on the server side.Just focus on including all the necessary tagsand attributes to create a well-structured form.You can add this form beforeor after the existing content on the contact page.

### Solution: Demo

(upbeat music)- Let's get into the solutionfor your final project challenge.Every form starts with the form element and the actionand method attributes.I'm going to add mine before the list.Since we're not focusing on making this form functionalon the server side, it's okayto leave these attributes blank.Next, we'll add a title for the form.I'll use an H2 and my title will be, Say Hello.You can also say something like, contact meor send me a message, whatever you'd like.Next, we'll add the labelsand inputs for the name and email.I'll start with a div to group each label input pair.Then add the label for namewith also a label of name.Then I'll add the input tag.We can leave it as a type of textand just add a couple more attributes.To connect the input to the label,we'll use the ID attribute with the same valueas the for attribute, which is name.To reference the form data when it's submitted.We'll add the name attribute,which will also be given a value of name, though we needto add several attributes for different uses.They often have the same valuessince they're describing the same input.Let's add one more, the required Boolean attribute,so this field can't be left blankwhen the form is submitted.For the email address field.I'll copy this block and paste it below.For the input, let's change the valuesof all the attributes to email.If you're on a Mac, here's the shortcut.Double click the value, then hold down the option key.Then double click on the next two values,and you can make the change all at once.I'm not quite sure what the equivalent is on a PC,but I say, try using the alt key.We can also use the pattern attributefor additional email validation.I'll use the same regular expression we usedin the 07\_04 exercise.Let's add that to the email input.If you haven't already, let's save the file.Next, we'll add the message box.Let's start this one with a div as well.Then I'll add a label tag with a for value of message.For the label text, I'll go with leave a message.Instead of another input field,we'll use the text area tag for the text box.Set both the name and ID attributes to message.Since the main purpose of this contact formis to receive messages, let's add the required attributeto the text area tag as well.And finally, we need a way to submit the form.So let's add a submit button.Although the input element can be used, I preferto use the button element since it offers more flexibility.The display text goes between the tags.I'll use send message.A button's default type is submit when used in a form,but let's include it anyway to explicitly define itas a submit button.And that's it for your form, but we're not quite done yet.We've created a well-structuredand semantically sound website with HTML.However, the current appearance is limitedto the browser's default styling.So let's see how this content can lookwith a little bit of CSS.I've created a styles.css file for this project,which can be found in the 07\_08 folder.You can just drag it into your project folderor make a copy.I'm going to add this just to the root folder.Then in the head section, we'll need to add a void link tag.The shortcut will automatically add the rail attribute,which we want to leave as is.For the href value, just use the file name styles.csssince it's in the root folder.Save the page to see the updates.Now let's copy this whole lineand make sure we add it to the other two pages as well.I'll add it to the blog page, save that file,and add it to the index and save it as well.And now we can click through our websiteand see the updates with the CSS.It looks like there's an error on my blog page.In the second post, the style for the paragraphis being included as the style for the time-heading.Let's do some troubleshooting.Here it is in the HTML.Let's check the markup.There's the header and there's the paragraph.So it looks like, I accidentally added the paragraphto the header section, and it should be outside of it.Let me move that and save the file.That's how it's supposed to look.Even with all my talk of being organizedand double checking, I still make errors.The trick is being able to spot it and fix it.If your styles look a bit differentfrom my project, that's okay.It might be something like the error I just fixed,but CSS is also very precise.It's all based on selecting a particular elementand applying a style to it.For example, if you used a section elementfor your blog post instead of an article element like I did,the styles won't apply since the styles in the CSS fileare specifically looking for article elements.I won't dive much further into CSS here,but this should give you a good sense of how it can be usedto transform the content beyond the default HTML styles.And now you're ready to complete the final chapterof this course.

## 07\_09 Solution: Text

Here is the final solution. For a further breakdown, continue reading below.

**file: contact.html**

<form action="" method="">

<h2>Say hello!</h2>

<div>

<label for="name">Name</label>

<input type="text" id="name" name="name" required>

</div>

<div>

<label for="email">Email</label>

<input type="email" id="email" name="email" required

pattern="[a-zA-Z0-9]+@[a-zA-Z]+\\.[a-zA-Z]+">

</div>

<div>

<label for="message">Leave a message</label>

<textarea name="message" id="message" required></textarea>

</div>

<button type="submit">Send message</button>

</form>

**file: index.html, contact.html, blog.html**

<link rel="stylesheet" href="styles.css">

Every form starts with the <form> element and the action and method attributes. Since we’re not focusing on making this form functional on the server side, it’s okay to leave these attributes blank.

<form action="" method="">

</form>

For the form title, use the <h2> tag since the <h1> is already used.

<form action="" method="">

<h2>Say hello!</h2>

</form>

The label and input pairs are grouped in a <div> and all the required attributes are added. For additional email validation, the pattern attribute was added using the same regular expression from the **07\_04** exercise.

<div>

<label for="name">Name</label>

<input type="text" id="name" name="name" required>

</div>

<div>

<label for="email">Email</label>

<input type="email" id="email" name="email" required

pattern="[a-zA-Z0-9]+@[a-zA-Z]+\\.[a-zA-Z]+">

</div>

For the “leave a message” box, a <textarea> tag is used to create a multi-line text field.

<div>

<label for="message">Leave a message</label>

<textarea name="message" id="message" required></textarea>

</div>

While the <input> element can be used to create a submit button, the <button> element offers more flexibility.

<button type="submit">Send message</button>

And that's it for your form! We've created a well-structured and semantically sound website with HTML. However, the current appearance is limited to the browser's default styling.

To get an idea of how this content can look with a bit of CSS, I've created a **styles.css** file for this project, which can be found in the **07\_08** folder.

* Add the file to your project’s root folder.
* Then add a <link> void element anywhere in the head section of the page
* For the href value, use the file name, **styles.css**
* Copy the whole line and add it to the **index** and **blog** pages as well.

<link rel="stylesheet" href="styles.css">

If your styles look a bit different from my project, that's okay. CSS is very precise—it's all based on selecting a particular element and applying a style to it. For example, if you used a <section> element for your blog post instead of an <article> element like I did, the styles won't apply since the styles in this CSS file are specifically for <article> elements.

I won't dive much further into CSS here, but this should give you a good sense of how it can be used to transform the content beyond the default HTML styles.

8 accessibility and internationalization

### What is accessibility?

- Throughout this course,I've mentioned ways to make HTML more accessible,and we've covered different topics.Now let's take a step backand explore what does this really mean.Accessibility means making websites usableby as many people as possible.While it mostly focuseson supporting people with disabilities,this practice benefits everyone.Even if you don't have a disability,you might be browsing on a mobile devicewith a smaller screen or using a slow internet connection.Accessible techniques can improve the experiencein these situations as well.The core principle is simple:everyone should have access to the web,and in some places it's actually the law.The W3C keeps a list of web accessibility lawsand policies by country.The legal requirements vary by region,but they usually apply to government agencies and sectorssuch as banking, telecommunications, libraries,educational institutions, and transportation.The W3C has also createdthe Web Content Accessibility Guidelines,which is a detailed guideexplaining how to make websites accessible.This guide is extensiveand has a lot of detailed information,so I recommend starting at the WCAG At a Glance article.You don't need to learn all the criteria all at once,but it's good to be awareof the four major areas of concern,Perceivable, Operable, Understandable, and Robust.Another great resourceis the Web Accessibility in Mind website at webaim.org.They offer training sessions, articles and toolslike Inaccessibility Evaluation and Color Contrast Checker.Web accessibility is a broad fieldand many people are able to specialize in this area.I've added additional resources to the Links.PDF fileif you'd like to learn more.

### Accessible Rich Internet Applications (ARIA) basics

- Informational websites tend to havea pretty straightforward structurebecause content is generally static.It doesn't change very often.Dynamic websites and web applications, however,tend to be more complexbecause the content changes more often.For example, on social media platforms,content is constantly updated in responseto various user actions, like posting or liking a post.As the web became more advanced and interactive,new accessibility challenges popped up.Adding more complex featuresand dynamic content using JavaScript librariessometimes resulted in less semantic HTML code,making it harder for screen readers to understand.There is a technology that can help, ARIA,which stands for Accessible Rich Internet Applications.It's a special set of attributes that can be addedto HTML elements to provide additional semanticsand improve accessibility.The first rule of ARIA use is to determinewhether you actually need it at all.According to the W3C documentation,if you can, use a native HTML elementor attribute with the semanticsand behavior you require already built in,instead of repurposing an elementand adding an ARIA role, state or propertyto make it accessible.When it isn't possible, ARIA attributes can be used.There are three main types that are documented here,but as I've mentioned before,I find the MDN Web Docs to be easier to read,so let's move over there.ARIA attributes fall into three categories,roles, states, and properties.Like other HTML attributes,each one has its own specific usage and values.You can see them all listed on this page.You can also go directlyto a specific attribute from the side menu.Let's do a high-level overview of these categoriesand look at some examples.ARIA roles define an element's purpose and function.They act as labels that tell assistive technologieswhat each element does on the page.Let's take a look at the button role as an example.It identifies an element as a buttonto assistive technologies like screen readers.However, adding the role only communicatesthat it's a button.It doesn't provide any functionality.When using the button or input element,the functionality is built in.So if you must use the button role,you'll need JavaScript to add the functionality.ARIA properties provide additional informationand semantics about HTML elements.For example, the ARIA required propertywith the value set to true is used with a form inputto indicate that it must be filled outbefore the form can be submitted.This helps users who rely on screen readerswho may not see visual indicators like asterisk,which are commonly used to mark a required field.Similar to the button role,functionality is not built into this property.JavaScript must be used to prevent the formfrom being submitted if the required field is empty.ARIA States are similar to propertiesin the sense that it provides additional information,but unlike properties,these elements often change based on user interaction.ARIA states define the current conditions of an element.The ARIA checked attribute is an example of an ARIA state,and it's used to indicate whether a checkbox is checkedusing a value of true or unchecked with a value of false.ARIA techniques don't affectthe visual appearance of a webpage.They only provide information to browser accessibility APIs,which screen readers use to interpret the content.And remember, ARIA is designedto help make webpages more accessible,but using it incorrectly can actually reduce accessibility,so whenever possible, use semantic HTML instead.

### Supporting languages

- [Instructor] In Chapter three's discussionof required HTML tags, we talked about the lang attribute,used to specify the language of the page content.This helps assistive technologiesuse the correct pronunciation.This is the snippet we've been using in our projects.The en value is an example of a language sub tag,which are two or three character codesand written in lowercase letters.As we can see in these examples,the subtags may include basic language codesas well as specific dialects.The W3C provides detailed informationon how to use language tagsand where to find specific language codes,which are managed by the IANA registry.The data is displayed as a text file.The language subtags are listed under Type: language.The subtag value is what you would useas the lang attribute value.It's quite extensive,so if you know what you're looking for,I would suggest using the Find feature in your browser.Command or Control + F to open the search box,and add in your search.There's also a language subtag lookup toollisted in the documentation, which you can also use.When you do a search, for example, English,the results will show the language subtag for basic English,as well as the dialects.Language subtags define the basic languageand are required when using the lang attribute.There are two other optional types of subtagsthat can be used to add more specific details.A region subtag defines the regional dialectof the base language,and the script subtag defines the writing systemused for that language.The region subtag uses either two uppercase lettersfor a country code, or three numbersfor a non-country region.For example, "fr-CA" represents French as spoken in Canada,and "es-013" represents Spanishas spoken in Central America.The subtag is documented in the IANA registryunder the region type.The script subtag is always four characters long,with the first letter capitalized.For example, French in Braille is "fr-Brai".The script subtag is documented under the script type.When using the lang attribute,it's best to keep it as simple as possible.Language subtags are required,but only use the optional subtagsif it's necessary for adding useful information.The dir attribute is another way to support languagesby specifying the text direction and telling the browsershow the text should flow on the page.It can be used with the following values:ltr for languages that are writtenfrom left to right, like English,and this is the default,so you don't need to specify thisfor left to right languages.But if your document has a mix of languages,then you may want to use it.The rtl value is used for languagesthat are written from right to left, like Arabic.A third option is auto, which lets the browser decide.It looks at the text and finds the first characterthat clearly shows which direction the text should go.Then it applies that directionto all the text in that element.Use this when you're not surewhich direction the text should go,such as when text comes from user input or outside sources.By implementing language attributes and text direction,we can create web content that is more accessibleand inclusive for usersacross different languages and writing systems.

### Next steps

- Congratulations on completingthe HTML Essential Training course.You deserve a round of applause and maybe a little break.When you're ready to resume your studiesand continue on your web development journey,here's some suggestions for next steps.Get in some more practice.Take what you've learned about HTMLand see what you can come up with,or just go through all the exercises againand see how much you remember.This will help reinforce your skills.You can also learn how to style your HTML documentswith my CSS Essential Training course.HTML is the foundation of the web,but it's also the beginning.As you progress, you'll want to learn CSS for stylingand JavaScript for interactivityto become a well-rounded web developer.Thanks for watching the course.