HTML File Basics

[The screen is open to an IDE. The file "index.html" is open and is displaying HTML code.]

Female: Let's look at some basics that you should follow when creating an HTML page from beginning to finish. When you begin an HTML file, you should always follow a few simple naming rules. It's strongly suggested that you save your file names in all lowercase. This will cut down on mistakes that might be made later. The first character of the file names should be a letter. Don't start the name with a number. There shouldn't be any spaces in the file name. If you want to separate words, you can use a dash or an underscore, but no other characters. These rules apply to folders, image files, or other resource files that you can name when you create your website as well. Remember, the file with the name index.html will always be recognized as the homepage of your website. So when the user types in the domain name of your website like Amazon.com, the index page will always come up first as the homepage. Other sub-pages or child pages on your website. Don't have to have any special name. You can name them whatever you want, but try to remember to keep it short and simple and follow the naming conventions.

[The instructor opens a file called "top_product.html". The file is blank.]

I'm going to use VS Code, and their shortcut here, and type an exclamation mark and tab or enter.

[The instructor types the exclamation mark and a list of suggestions opens. The first suggestion is highlighted. When the instructor hits the tab. The file removes the exclamation mark and autofills the file with the tags <!DOCTYPE html>, <html lang="en">, a <header> with <meta> tags and a <title> tag, and a <body> tag.]

This brings in the beginning of our HTML page. Notice the Meta tags that come in. Let's talk about those. HTMLlets you add additional information about your document using Meta tags. The Meta tags can include name-value pairs describing properties of your HTML document. These tags are not visible on your webpage, but belong in the head section of your HTML. This is where we have information about our page, not the content that shows up on our page. I could add my own stating who the author of the document is.

[The instructor adds the text "<meta name="author" content ="Sue Smith">" in between the <head></head> tags.]

It's not required and it does not affect the look of the document in any way, but it is just additional information about our page if you want to include it. The next meta tag tells us what character set we're using in our document.

[The instructor highlights the code "<meta charset="UTF-8">"]

Utf-8 refers to the Unicode character set. Unicode can handle any international language characters. So it's a good one to use on your webpage. The next meta tag tells Microsoft to use the latest rendering engine edge.

[The instructor highlights the code "<meta http-equiv="X-UA-Compatible" content="IE=edge">"]

This one is only needed in a few instances. The last metatag is important and I would always include it on any HTML document.

[The instructor highlights the code "<meta name="viewport" content="width-device-width, initial-scale=1.0">"]

It allows developer, the developer to take control over the viewport or visible area of a web page, no matter what device is being used. Whether the user is seeing your web page on a phone, tablet, laptop, or desktop, the width equals device width sets the width of the page to follow the screen width of the device. And the initial scale equals 1.0, sets the initial zoom level when the page is first loaded in the browser. So this initial zoom at 1.0 or 100%. When used in conjunction with media queries, which we'll touch on later, it can help make pages be

responsive, or in other words, look good on any device. As you create your file, make sure you're using semantic elements for your content.

[The instructor opens a file called "staywarm.html". Code in HTML is written.]

Most tags describe what is inside them. And our semantic, for example, the nav tag has a navigation contained between this opening and closing tag.

[The instructor highlights the nav opening and closing tag. Inside is a list of links to other pages on the website.]

But some elements are not semantic like divs or span tags. These tags tell nothing about their content. They simply create a wrapping element around their content, usually for CSS presentation purposes, not because it gives their element meaning. So for example, I might not have a good tag that has a semantic fit with the banner information.

[The instructor highlights a div with the ID "banner-msg".]

But I know that I'll be doing CSS and all that content together later. So I place a div with the class of banner message around it. Also the span element is an example of a non-semantic tag.

[The instructor highlights a span tag with the class "emphasize".]

I might want the color of just that text "have an outdoor heat source" to be a different color. With CSS, it doesn't really have a semantic meaning to that content. Why use a nav tag when a div tag would work and your pages navigation or menu will look just the same on the page. The reason it's important is because screen readers and search engines use this extra information. And it's important to give our pages as much meaningful information about our content as we can. This makes it accessible to those navigating our page with that site. And also for search engines like Google to make more sense out of our page and make it appropriately. It's also just good practice to do it. Once we're done with our file, we should always validate the code. Validating your HTML CSS is important. Validating Is just like it sounds, making sure our code is valid or coded properly. W3c is the World Wide Web Consortium or international community that maintains and develops web standards. They have a validator that you should use to validate your HTML. It's at validator.w3.org.

[The instructor goes to validator.w3.org. The website has a banner saying "markup Validation Service". There are three tabs that say, in order, "Validate by URL", "Validate by File Upload", and "Validate by Direct Input".]

Here you can check your webpage HTML code by either placing the web address of your page or uploading your HTML file, or by pasting your HTML code in directly.

[The instructor selects the "Validate by Direct Input" and pastes in the code in an input field labeled "Enter the Markup to validate:". The instructor hits a button with text saying "Check".]

Then when you check it, it will then let you know if you have any errors.

Then you should correct any errors so that your page will validate with no errors. Here's what it would look like once it validates.

[Text is added in a green box that says "Document checking completed. No errors or warnings to show."]

Why is this important? First, it can help you debug your code. If your page is not rendering like you expected it to, validating can help show you where your errors are. Also it helps you learn good practice, having clean code, and help with site performance. It's a sign of professionalism. And those hiring you later might look at your portfolio or webpages and validate them to see if you have good code. Also, search engines like Google will rank your website much better if your code validates. So get into a good habit of always validating your HTML code. You can also validate CSS at jigsaw.w3.org/css-validator.

[End of video]