Html Content

HTML Fundamentals

What is html?.

HTML is the language used to create the web pages you visit everyday. It provides a logical way to structure content for web pages.

Let's analyze the acronym "HTML", as it contains a lot of useful information. HTML stands for HyperText Markup Language.

Doctype:

A web browser must know what language a document is written in before they can process the contents of the document.

You can let web browsers know that you are using the HTML language by starting your HTML document with a document type declaration.

The declaration is the following:

<!DOCTYPE html>

HTML:

The <!DOCTYPE html> declaration is only the beginning, however. It only tells the browser that you plan on using HTML in the document, it doesn't actually add any HTML structure or content.

To begin adding HTML structure and content, we must first add opening and closing <html> tags, like so:

<html> </html>

Anything between <html> and </html> will be considered HTML code. Without these tags, it's possible that browsers could incorrectly interpret your HTML code and present HTML content in unexpected ways.

Head:

The <head> element will contain information about the page that isn't displayed directly on the actual web page:

<head> </head>

Title:

The browser displays the title of the page because the title can be specified directly inside of the <head> element, by using a <title> element.

<title></title>

Body:

Before we can add content that a browser will display, we have to add a body to the HTML file. Once the file has a body, many different types of content can be added within the body, like text, images, buttons, and much more.

<body></body>

HTML Content

Headings:

Headings in HTML can be likened to headings in other types of media. For example, in newspapers, large headings are typically used to capture a reader's attention. Other times, headings are used to describe content, like the title of a movie or an educational article.

<h1></h1> (alternative, h2, h3, h4, h5, h6)

Paragraphs:

Paragraphs are great for expanding the amount of content (text) on your web page. As you begin to add more text to your web page, however, keep in mind that large amounts of text in paragraph format can overwhelm web page visitors. For example, if multiple paragraphs on your web page each contain large amounts of text, your web page could become difficult to consume.

<p></p>

Unordered lists:

In HTML, you can use the unordered list for text you decide to format in bullet points. An unordered list outlines individual list items with a bullet point. You've probably used an unordered list when writing down a grocery list or school supplies list.

CSS Fundamentals

<ul>

<li>Car

<li>Airplane

<li>Boat

</ul>

The list was created using the <ul> element and all individual list items were added using <li> elements.

Ordered lists:

Ordered lists are like unordered lists, except that each list item is numbered. You can create the ordered list with the <ol> element and then add individual list items to the list using <li> elements.

<ol>

<li> Rice and beans

<li> Earthquake

<li> Cell phone

</ol>

Links:

You can add links to a web page by adding an anchor element <a> and including the text of the link in between the opening and closing tags. The anchor element in the example above is incomplete without the href attribute.

<a href=""></a>

Target and Blank:

For a link to open in a new window, the target attribute requires a value of \_blank. The target attribute can be added directly to the opening tag of the anchor element, just like the href attribute.

<a href="" target="\_blank"></a>

Images:

The <img> element lets you add images to a web page. This element is special because it does not have a closing tag, it only has an opening tag. This is because the <img> element is a self-closing element.

<img src="https://www.example.com/picture.jpg" />

Different types of linking:

HTML allows you to turn nearly any element into a link by wrapping that element with an anchor element. With this technique, it's possible to turn images into links by simply wrapping the <img> element with an <a> element.

<a href="" target="\_blank"><img src="#" alt=""/></a>

Line Breaks:

The line break element is one self-closing tag. You can use it anywhere within your HTML code and a line break will be shown in the browser.

<br/>

Comments:

Comments begin with <!-- and end with -->. Any characters in between will be treated as a comment.

Including comments in your code is helpful for many reasons:

1. They help you (and others) understand your code if you decide to come back and review it at a much later date.

2. They allow you to experiment with new code, without having to delete old code.

<!-- This is a comment that the browser won't show. -->

CSS Fundamentals

Style:

The <style> element allows you to write CSS code between its opening and closing tags. To use the <style> element, it must be placed inside of the head.

<style></style>

Linking the CSS file:

You can use the <link> element to link the HTML and CSS files together. The <link> element must be placed within the head of the HTML file. It is a self-closing tag and requires the following three attributes: href, type, rel.

When linking an HTML file and a CSS file together, the <link> element will look like the following:

<link href="/style.css" type="text/css" rel="stylesheet">

Element Selectors:

To style an HTML element using CSS, you must first select that element in the CSS file. For example, to style a <p> element, the syntax to select it using CSS is:

p{

}

all paragraph elements are selected using a CSS selector. The selector (in this case) is p. Note that the CSS selector essentially matches the HTML tag for that element, but without the angle brackets.

Multiple Element Selectors:

Styling with CSS would be very inefficient if you were forced to manually style the same property across many elements.

Fortunately, you can select multiple elements at once so that you can save time styling a shared property.

h1, h2, p {

color: Green;

}

Comments:

Just like HTML, you can also leave comments in your CSS file. CSS comments begin with /\* and end with \*/, like so:

/\* This is a comment in CSS! \*/

/\*

h1 {

color: Red;

}

\*/

In the example above, a valid CSS rule has been "commented out." This practice is useful when you want to experiment with new code without having to delete old code.

Styling with CSS

Foreground and Background:

In CSS, these two design aspects can be styled with the following two properties:

1. color - this property styles an element's foreground color.

2. background-color - this property styles an element's background color.

h1 {

color: Red;

background-color: Blue;

}

Named Colors:

In CSS, these colors are technically known as named colors. There are a total of 147 named colors, such as Red, Blue, Cyan, Crimson, Etc

RGB Colors:

Although named colors provide 147 different options, this is a small amount when you consider the flexibility of CSS. To take advantage of the full spectrum of colors that CSS supports, you have the option of using RGB colors.

To use RGB colors, you can use the rgb() value when styling a color.

h1 {

color: rgb(123, 20, 233);

background-color: rgb(99, 21, 127);

}

HEX Colors:

When read from left to right, each group of two characters responds to a value for red, green and blue, respectively. In the example above, 09 refers to the value for red, AA refers to the value for green, and 34 refers to the value for blue. All hex color codes begin with a # character.

h1 {

color: #09AA34;

}

HSL:

HSL stands for Hue, Saturation, and Lightness. You can use HSL colors in your CSS like this:

h1 {

color: hsl(182, 20%, 50%);

}

Notice that using HSL is very similar to using RGB.

Alpha Value: a

Opacity is a measure of how transparent a color is. To modify opacity in RGB colors, CSS offers the rgba() value. Note the slight difference in rgb() and rgba(). The extra a character in the rgba() value is known as the alpha value. It represents the opacity of a color. The alpha value can be a number between 0 or 1, inclusive.

h1 {

color: rgba(123, 88, 9, 0.5);

}

Font family:

When setting typefaces on a web page, keep the following points in mind:

1. The font specified in a stylesheet must be installed on a user's computer in order for that font to display when a user visit the web page.

2. How exactly does the browser know what typeface to use when displaying the web page? The default typeface for all HTML elements is Times New Roman.

h1 {

font-family: Garamond;

}

3. When the name of a typeface consists of more than one word, it must be enclosed in double quotes (otherwise it will not be recognized), like so:

h1 {

font-family: "Courier New";

}

Fallback Fonts:

Most computers have a small set of typefaces pre-installed. This small set includes serif fonts and sans-serif fonts, like Times New Roman and Arial, respectively.

When the stylesheet specifies a font not installed on a user's computer, the pre-installed fonts can be used as fallback fonts for users.

h1 {

font-family: Garamond, Times, serif;

}

Font size 1:

Often times, different sections of a web page and are highlighted by modifying the font size. To change the size of text on your web page, you can use the font-size property.

p {

font-size: 18px;

}

Font size 2:

There are three units of measurement for font size: px, em, and %:

1. Px: Represents the unit of pixels.

2. Em: Ems are a relative unit of measurement. They change the size of text relative to the parent element's size of text.

3. %: Percentages are also a relative unit of measurement. The default size of text in web browsers is 16 pixels, or 16px.

p {

font-size: 18px;

}

p {

font-size: 1.3em;

}

p {

font-size: 150%;

}

Line height:

Text on a web page must also be easy to read. When text is styled to appear larger, the vertical spacing between lines of text can decrease, creating text that is difficult to read, particularly in paragraphs. To avoid this problem, you can modify the spacing between lines of text with the line-height property.

p {

line-height: 1.5em;

}

Word spacing:

The default amount of space between words is usually 0.25em. In the example above, the word spacing is set to 0.3em, which represents an increase of only .05em in word spacing.

h1 {

word-spacing: 0.3em;

}

Letter spacing:

The technical term for adjusting the spacing between letters is called "kerning". Kerning can be adjusted with the letter-spacing property in CSS.

h1 {

letter-spacing: 0.3em;

}

Font weight:

It's common to bold important headings or keywords. In CSS, the font-weight property turns bold on or off.

p {

font-weight: bold;

}

Font style:

The italic value causes text to appear in italics. The font-style property also has a normal value, for the same reasons discussed in the previous exercise.

h3 {

font-style: italic;

}

Text Transformation:

Text can also be styled to appear in either all uppercase or lowercase with the text-transform property. The code in the example above formats all <h1> elements to appear in uppercase, regardless of the case used for the heading within the HTML code. Alternatively, the lowercase value could be used to format text in all lowercase.

h1 {

text-transform: uppercase;

}

Text alignment:

To move, or align, text, we can use the text-align property. The text-align property can be set to one of the following three values:

1. left - aligns text to the left hand side of the browser.

2. center - centers text.

3. right - aligns text to the right hand side of the browser.

h1 {

text-align: left;

}

Organizing HTML and CSS

IDs: HTML:

With the proper labels, we can style individual HTML elements! Specifically, we can label HTML elements with a unique identifier, or ID. We can then style that specific element in the stylesheet. To label an element with an ID, we can use the id attribute on an HTML element.

<h1 id="Praise">Praise</h1>

ID selectors: CSS

To style a specific element labeled with an ID, you can use an ID selector in the stylesheet. All ID selectors begin with the octothorpe character: #. The value of the ID immediately follows the octothorpe. Once you've correctly targeted the element, you can proceed to style it as usual.

#Praise {

background-color: #56ABFF;

}

Classes: HTML

CSS offers a solution to this limitation. For multiple elements that should share the same styling, classes can be used to label them. To label an element with a class, we can use the class attribute on an HTML element.

<h1 class="Soccer">The best soccer player in Europe</h1>

Class Selectors 1: CSS

we can style elements belonging to the same class at once. How exactly do you select them in CSS, though?. To style elements of the same class, you can use a class selector in the stylesheet.

.Praise {

font-family: Georgia, Times, serif;

color: #A3B4C5;

text-transform: uppercase;

}

Class Selectors 2: CSS

A heading and a paragraph (both with a class of breaking) may need to share the same typeface, but the paragraph may require a styling better suited for paragraphs, as in the following example.

.breaking {

font-family: Georgia, Times, serif;

}

p.breaking {

line-height: 1.3em;

}

Multiple selectors: CSS

CSS does not limit you to selectors that target a single element or class. The same syntax can be used to style multiple classes at once.

h1, p {

font-family: Garamond, serif;

}

.first, .last {

font-size: 20px;

}

Multiple classes: HTML

It's also possible to label HTML elements with more than one class. How is this functionality useful?. When those same elements must also be differentiated, however, labeling with an additional class is helpful.

<h1 class="Relative Space">Depth</h1>

<h1 class="Relative Stars">firmament</h1>

. Relative {

font-family: Georgia, serif;

}

.Space {

font-color: #0902CC;

}

.Stars {

font-color: #B097DD;

}

The div:

HTML offers an element that is the backbone of code organization: the div, represented by <div> in HTML. You can think of the div as a box, or container, that groups elements that belong together. For example, a <div> can be used to group together all of the elements that make up the navigation for a web page, or any other section of a page.

<div>

<h1>Sonic the Hedgehog</h1>

<p> ... </p>

</div>

Divs and Classes:

This does not mean that labeling individual elements with classes is no longer useful. Even when divs are labeled with classes, there will be many other times when an individual element will need to be labeled with a class.

<div class="container">

<h1 class="title">Sonic the Hedgehog</h1>

<p> ... </p>

</div>

div.container {

background-color: rgb(252, 255, 205);

font-family: Roboto, Helvetica, sans-serif;

}

h1.title {

color: #0D1A2F;

}

When a div is styled, all elements inside of the div will inherit the styling applied to the div. This example illustrates how easy it is to style sections of a web page using div.