**HTML5**

HTML5 is the new standard for HTML. The previous version of HTML came in 1999. The web has changed a lot since then. HTML5 is still a work in progress. However, most modern browsers have some HTML5 support.

Learn more about [HTML5 Semantic Tags at the W3 Schools](https://www.w3schools.com/html/html5_semantic_elements.asp).

**HTML5 Development Rules**

* New features should be based on HTML, CSS, DOM, and JavaScript
* Reduce the need for external plugins (like Flash)
* Better error handling
* More markup to replace scripting
* HTML5 should be device independent
* The development process should be visible to the public

**New Features of HTML5**

* New content specific elements, like article, footer, header, nav, section
* New form controls, like calendar, date, time, email, url, search
* The video and audio elements for media playback
* The canvas element for drawing
* Better support for local offline storage

**HTML5 DOCTYPE and charset**

* <!DOCTYPE HTML>
* <html lang="en">
* <meta charset="UTF-8">

**Obsolete Elements**

* frameset, frame and noframes
* acronym
* presentational elements such as font, big, center and strike
* presentation attributes such as bgcolor, align, cellspacing, cellpadding

**Elements That Have Changed**

**small**

now means 'the small print' not small text size

**b**

is now 'stylistically offset this from the text around it', not bold

**i**

means 'in an alternate voice or mood', not italicize

**cite**

now is the title of a work

**a**

element may now be wrapped around multiple block elements

**HTML5 Semantic Elements**

HTML5 has added several new semantic elements that help structure your page.

* provide a new way for browsers to understand your content
* works as a kind of outlining tool for your site
* may be helpful with SEO and accessibility (in the future if not yet)
* not all elements supported by all browsers
* included here are those supported by most major browsers

**Semantic Block Elements**

<section></section>

* is used for content that can be grouped thematically.
* all content contained by <section>is related</section>
* should not be used just for styles (stick with a <div>tag</div> for that)
* indicates major topics for outlining

<header></header>

* typically contains the headline or grouping of headlines for a page
* may also contain other supplemental information like logos and navigational aids
* may also be used within a section

<footer></footer>

* used for content *about* a page, such as who wrote it, links to related information and copyrights.

<nav></nav>

* used to contain *major* navigation links for a page
* often included infor the page

<article></article>

* used for content that is self-contained and could be consumed independent of the page as a whole, such as a blog entry
* An article should make sense on its own and it should be possible to distribute it independently from the rest of the site.

**Examples of Articles**

* forum post
* newspaper article
* blog entry
* user comment

<aside></aside>

* a portion of a page that is related to the content around it, but also separate from that content, such as a sidebar or pull-quotes.
* if you can remove the content without affecting understanding of the page, use <aside></aside>
* typically formatted as a sidebar

<address></address>

* holds contact information for the author of a page or section
* often displayed in italics

**Semantic Inline Elements**

<figure></figure>

* specifies self-contained flow content such as images or code
* content should be relevant to the main content, but if removed, should not affect the flow of the document.

<figcaption></figcaption>

* contains a caption for the <figure>tag</figure>
* should be first or last element in <figure>tag</figure>

<mark></mark>

* defines marked text
* use for text you want to be highlighted

**Sample HTML5 Layout and Code**



<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8">

<title>Title</title>

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

</head>

<body>

<header>

<h1>Header in h1</h1>

</header>

<nav>

<h2>Navigation</h2>

<ul>

<li><a href="#">Menu Option 1</a></li>

<li><a href="#">Menu Option 2</a></li>

<li><a href="#">Menu Option 3</a></li>

</ul>

</nav>

<main>

<section>

<h2>Section</h2>

<article>

<h2>Article #1</h2>

<p>This is the first article. This is <mark>highlighted</mark> text. An article can be almost ant text/image combination that contains related content.</p>

</article>

<article>

<h2>Article #2</h2>

<p>This is the second article. Articles could be blog posts, comments, reviews, etc.</p>

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.</p>

</article>

</section>

<aside>

<h2>Aside</h2>

<figure>

<img src="https://bhcwebdev.com/articles/images/HTML5PageLayout.jpg" alt="My Image Alt Text">

<figcaption>HTML5 Layout Image</figcaption>

</figure>

</aside>

</main>

<footer>Footer - Copyright &copy;</footer>

</body>

</html>

Edit in JSFiddle

HTML

CSS

Result

\* {

margin: 0;

box-sizing: border-box;

}

header {

color: WhiteSmoke;

background-color: #c00;

border: thin solid white;

padding: .5em;

}

nav {

color: WhiteSmoke;

background-color: Orange;

border: thin solid white;

padding: .5em;

}

main {

display: table;

width: 100%;

}

section {

color: WhiteSmoke;

background-color: gold;

border: thin solid white;

padding: 5px;

display: table-cell;

}

article {

color: WhiteSmoke;

background-color: YellowGreen;

border: thin solid green;

border-radius: 5px;

padding: .5em;

margin: .5em;

}

aside {

color: WhiteSmoke;

background-color: blue;

border: thin solid white;

padding: .5em;

display: table-cell;

}

img {

max-width: 80%;

}

footer {

color: WhiteSmoke;

background-color: purple;

border: thin solid white;

clear: both;

text-align: center;

height: 2em;

line-height: 2em;

}

**CSS Style Sheets**

Cascading Style Sheets (CSS) provide a way for web developer's to have more control over the appearance of their web pages. CSS uses style tags and attributes to format text and specify layout for your Web pages. Remember the principles of layout and composition when designing your web pages.

Video produced by [GCFLearnFree.org](https://edu.gcfglobal.org/en/)

There are three ways to add style to your web pages through CSS.

* in-line style
* embedded (internal) style sheet
* external style sheets

In-line style works by adding a style attribute to an existing tag. In-line style is not standards compliant, so we will not be using it in this class. However, if you were to come across some HTML code that has a style attribute in a tag (i.e. <p style="color:#ff0000">, then you will understand that it is CSS applied at the element level. You may also find inline style useful in cases where you do not have access to the html code or stylesheet on a page - for example if you are posting information from a form based application. For example, I use it in Blackboard to add styles since I don't have access to the style section or external stylesheet within eCampus.

An embedded style is a (internal) style sheet that is included in an HTML document. The styles in the embedded style sheet will apply to everything on that Web page. The style rules are coded between the <style> and </style> tags and are located in the <head> section of your document:

<style type="text/css">

style rules will be listed here

...

</style>

Copy

An external style sheet is a text document that contains a list of style rules which can be applied to any number of web pages. The text document does not contain any HTML content—only style definitions. The document should be named with an extension of .css and included in the same folder with your web pages. To use the style sheet on your Web pages, you must link it to each page that you want it to format by placing the <link> tag in the <head> section of each page:

<link rel="stylesheet" href="mystyles.css" />

Copy

where "mystyles.css" is the name of your external style sheet document.

Another option is to use @import to attach an external stylesheet to a page. The @import rule must be the first statement in the <style> element. You can import more than one external stylesheet, but they must all be before any style selectors and will be applied in the order listed.

<style>

@import url("filename");

body {

font-face: arial;

}

. . . more styles here . . .

</style>

Copy

**Internal vs. External Stylesheets**

Internal and external stylesheets each have advantages and disadvantages when used with mobile devices.

**Internal Stylesheets**

* Not reusable
* Styles must be coded on each and every page where they are used
* Stored in same document with html code
* Only one server response needed for the page
* Render quickly
* Increase page size

**External Stylesheets**

* Reusable across multiple pages
* Create styles once and apply to multiple pages.
* Ensures design consistency
* Easier to maintain and update
* Stylesheet only has to be loaded one time if the user browses several pages in your site
* Stored separately
* Increases download time for the initial page

So which one is best? If you have a lot of styles that are applied across the majority of your mobile web pages, then it is probably best to put these in an external stylesheet and attach it to each page. Although the initial download may take a little longer, performance on subsequent pages will be quicker. Plus your design will be consistent and any changes you make to the design will be quicker and easier. However, if you have styles that only apply to one page, you should probably include those in an internal stylesheet. This will reduce the size of the external stylesheet for all other pages and render quickly for the page where they are used.

**CSS Rules**

A CSS rule looks like this...

p {

color: #660000;

additional properties. . .

}

Copy

where p is the selected element that the style will be applied to and color: #660000 is the style property (color) and value (#660000) that will be applied. A CSS rule can contain any number of properties and values, separated by a semicolon (;). In this case, any text on a Web page that was inside a *<*p> element would be maroon. You can apply styles to multiple elements by separating them with a comma.

h1, h2, h3 {

color: #660000;

}

Copy

**Class attribute**

Classes allow you to set up multiple style specifications for the same HTML tag. To create a class in your style sheet, you simply type a period(.) followed by the name of the class:

<style>

.special {

color: #660000;

}

</style>

Copy

This creates a class called special which will turn the text maroon. To use the class, add a class attribute to the HTML tag:

<p class="special">

Copy

Now, instead of all the paragraphs on the Web page being maroon, only those with a class of "special" will be maroon. The class attribute can be added to any HTML tag. Occasionally, you may find that you want to add style where no html tag is currently found. There are 2 special html tags that can be used for this purpose. The <span> tag is an inline tag that can be used to add a style class to your code where no existing HTML tag is available and the <div> tag is a block tag that can be used to add a style class to an entire section of your HTML page (i.e. multiple paragraphs, headings, etc.)

You can create a class just for one html tag by placing the name of the tag in front of the class name:

<style>

p.special {

color: #660000;

. . .

}

</style>

Copy

This would create the special class just for the paragraph tag and not any other tags.

You can also apply multiple classes to the same tag by listing them inside the quotes:

<p class="special creative">

Copy

This would apply both the special and the creative classes to the paragraph.

**id attribute**

The id attribute is similar to the class attribute in that it allows you to apply styles to specific elements. However, instead of assigning a class to an element, you give the element an id attribute. Generally, you should use id when the styles are unique to a single element. Classes are more generic and can be used for multiple elements. In the style definition, you can specify styles just for an id by using the #id designation:

<style>

#id1 {

border-color: red;

border-width: medium;

background-color: aqua;

padding: 20px;

}

</style>

Copy

**Link pseudoclasses**

CSS has some built in pseudoclasses that can be used to override the default styles for links:

<style>

a:link {color:#FF0000;} /\* unvisited link \*/

a:visited {color:#00FF00;} /\* visited link \*/

a:hover {color:#FF00FF;} /\* mouse over link \*/

a:active {color:#0000FF;} /\* selected link \*/

</style>

Copy

If you use all 4 anchor pseudoclasses, they must appear in the order listed here or they will not work properly.

**Action pseudo-classes**

* :focus - applies when a form element is selected and ready for input
* :hover - applies when the mouse hovers over an element
* :active - applies when the element is in the process of being clicked or tapped

**Pseudo-element selectors**

* :first-line - applies to the first line of the element. Limited style properties available
* :first-letter - applies to the first letter of the element. Limited style properties available
* :before - inserts content before the element
* :after - inserts content after the element

**Other selector options**

The following code is used in the examples for these selector types.

<div id="gallery">

<img src="image1.png" id="img1">

<p>

<img src="image2.png" id="img2">

<img src="image3.png" id="img3">

<img src="image4.png" id="img4">

</p>

</div>

Copy

| **Selector Definition Table** | | | |
| --- | --- | --- | --- |
| **Name** | **Target** | **Selector** | **Selection** |
| Descendent selector | Only selects elements that are a child of the first selector | #gallery img  Copy | #img1  Copy  ,  #img2  Copy  ,  #img3  Copy  and  #img4  Copy |
| Child selector | Only selects a direct child of the selector | #gallery > img  Copy | #img1  Copy  (others are in level below) |
| Adjacent sibling selector | Only selects an element that comes immediately after another element with the same parent. | #img2 + img  Copy | #img3  Copy |
| General sibling selector | Selects an element that shares a parent with the selector and comes after the selector | #img2 ~ img  Copy | #img3  Copy  and  #img4  Copy |

There are many more specific ways to apply CSS to your code. A good reference for CSS selectors can be found on the W3Schools site at <http://www.w3schools.com/cssref/css_selectors.asp>, if you need more help with this process.

**Comments in stylesheets**

If you want to document your styles in your stylesheets, you can add comments by placing the comment between /\* and \*/ delimiters.

<style>

/\* this is a comment \*/

body {

width: 80%;

margin: 0 auto;

}

</style>

Copy

**The Cascade**

In some cases you may have more than one style rule that applies to the same element on your page. There are several principles that can help you understand how they are applied to the page.

**Style Inheritance**

Styles defined for a parent tag or element will apply to the descendant elements or tags unless they conflict with styles defined for that element. For example, if you have a style on the body tag that changes the text color to maroon, then all text on the entire page will be maroon -- unless you have a style on a lower level tag that conflicts. So, if on that same page, you have style defined for the h1 tag that has the color property set to navy blue, then the text inside of the h1 tag would be navy instead of maroon.

**Specificity**

If you set the class on a paragraph tag to a different color, then it would override the body attribute. This is known as specificity - the more specific the rule, the stronger. So, an element nested inside another element (such as the h1 inside the body, or an em inside a paragraph) is more specific so those rules will apply. A class is more specific than an element and an id is more specific than a class.

**Style Precedence**

You may include embedded style and external style sheets in the same Web page. When conflicting styles are found for a web element, the browser will apply the styles in the order that they occur.

You can use both an internal and external stylesheet in the same page. The styles will be applied in the order that they appear. So if the link to the external stylesheet is above the <style> section, the external will apply first and the internal will be applied second. If the link to the external stylesheet is below the <style> section, then the internal styles are applied first and the external second. This is only significant when there are conflicting styles as demonstrated in the examples below. For example, if I have h1 {color: blue; font-style: italic;} in my external stylesheet and h1 {color: red; font-weight: bold;} in my internal stylesheet, then the following would apply.

**External Stylesheet**

<style>

h1 {

color: blue;

font-style: italic;

}

</style>

Copy

| **External Stylesheet Linked above Internal** |
| --- |
| **Example Code** |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8" />  <title>Example 1  <link href="cssEx.css" type="text/css" rel="stylesheet" />  <style>  h1 {  color: red;  border-top: solid green;  }  </style>  </head>  <body>  <header>  <h1>Example 1</h1>  </header>  </body>  </html>  Copy |
| **Appearance on Page** |
| ***Example 1*** |

| **Internal Stylesheet Linked above External** |
| --- |
| **Example Code** |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="utf-8" />  <title>Example 2</title>  <style>  h1 {  color: red;  border-top: solid green;  }  </style>  <link href="cssEx.css" type="text/css" rel="stylesheet" />  </head>  <body>  <header>  <h1>Example 2</h1>  </header>  </body>  </html>  Copy |
| **Appearance on Page** |
| ***Example 2*** |

Because the font-style and border-top attributes do not conflict, they apply to the h1 text in both examples. However, since the color attribute is different, in the first example, the internal style applies because it occurs second. In the second example the external style applies because it comes after the internal.

**!important**

To prevent a subsequent CSS rule from overriding a rule (such as in a cascade), you can add !important after the style value.

<style>

blockquote {

font-style: italic **!important**;

}

</style>

Copy

You can also link multiple stylesheets to a single web page and designate them for use on a particular output such as screen, print, handheld, etc. by adding media="output" to the link.

<style>

@media output {

style rules...

}

</style>

Copy

To designate rules for a particular media, for example, if your site has a 2 column layout, you might want to have a different layout for mobile devices (media="handheld") to make it more readable on a smaller screen.

**HTML5 Web Page with CSS**

The example below uses some minimal CSS to create a wrieframe layout. A wireframe layout is a quick and easy layout used to visualize a page concept. Designers and developers use this to provide choices to clients and solicit their feedback. It's much easier to make site changes early than to make a lot of changes later.

<!DOCTYPE html>

<html lang="en">

<head>

<title>Wireframe</title>

</head>

<body>

<header>

<h1>My Header</h1>

</header>

<nav>

&nbsp;<a href="#">Home</a> | <a href="#">Products</a> | <a href="#">About Us</a>

</nav>

<main>

<section>

<h2>A Section</h2>

<p>A section usually contains the main content of your site. This section has a width limit.</p>

<p>On most browsers and displays you will see that the section and aside make two uneven length columns.

This can be difficult to control. You can however,

hide these unequal lengths by making the background colors the same.</p>

</section>

<aside>

<h3>The Aside</h3>

<p>This can contain any kind of content but often has ads or other navigation.</p>

</aside>

</main>

<footer>

<h3>My Footer &copy; 2020</h3>

</footer>

</body>

</html>

\* {

box-sizing: border-box;

}

header {

width: 100%;

height: 10%;

background-color: darkblue;

color: whitesmoke;

padding: 1em;

}

nav {

width: 100%;

height: 1.5em;

background-color: darkblue;

color: whitesmoke;

}

a {

color: white;

}

main {

width: 80%;

height: 24em;

margin-left: auto;

margin-right: auto;

background-color: brown;

}

section {

width: 80%;

height: 100%;

float: left;

background-color: bisque;

padding: .5em;

}

aside {

width: 20%;

height: 100%;

float: left;

background-color: cadetblue;

padding: .5em;

}

footer {

width: 100%;

height: 4em;

clear: both;

background-color: crimson;

color: bisque;

text-align: center;

padding-top: .5em;

}

### CSS3

CSS3 is the name of the latest standard for CSS. These are backwards compatible - i.e. CSS2 designs will display properly in CSS3, but they not fully supported by all browsers. The good news is they are becoming more main stream every day. For detailed list of CSS browser support see [http://www.w3schools.com/cssref/css3\_browsersupport.asp.](http://www.w3schools.com/cssref/css3_browsersupport.asp)

Browser specific versions of some attributes.

* -webkit- browsers based on the WebKit rendering engine including Safari, Chrome, and iPhone browsers
* -o- Opera browsers - seldom used
* -moz- optimized for Mozilla rendering engine - mainly Firefox

**borders**

border-radius: top-left top-right bottom-right bottom-left;

/\* Example \*/

border-radius: 25px;

/\* places rounded corners on all four corners \*/

Copy

* rounds the corners of a border
* specified in pixels
* supported by all major browsers
* if 1 number, applies to all 4 corners
* if bottom-right omitted, same as top-left
* if bottom-left omitted, same as top-right

box-shadow:h-shadow v-shadow blur spread colorinset;

Copy

* adds shadow to boxes
* not recommended for mobile sites
* values are usually specified in pixels
* h-shadow is horizontal offset - required - may be negative
* v-shadow is vertical offset - required - may be negative
* blur is blur distance - optional
* spread is spread distance - optional
* color is color - optional
* inset changes the shadow from outset to inset - optional
* supported in all browsers

**backgrounds**

background-size: width height;

Copy

* specify the size of the background image
* may be in pixels or percentages
* percentages are within the parent element

background-origin: position;

Copy

* specifies what area is filled by the background
* possible values are content-box, padding-box, or border-box;
* supported in all major current browsers but may not work in older versions.

**text**

text-shadow:h-shadow v-shadow blur color;

Copy

* adds shadow to text
* not recommended for mobile sites
* values are usually specified in pixels
* h-shadow is horizontal offset - required - may be negative
* v-shadow is vertical offset - required - may be negative
* blur is blur distance - optional
* color is color - optional
* not supported in IE

word-wrap: break-word;

Copy

* allows long words to break and wrap onto the next line

@font-face rule

Copy

* automatically downloads a font file to users computer when needed
* font-file must be included on the website
* good source for font-files is <http://www.fontsquirrel.com/>

@font-face {

font-family: fontname;

src: url('filename.ttf') format("truetype"),

src: url('filename.ttf') format("woff"),

src: url('filename.svg') format("svg"),

url('filename2.eot') format("opentype");

}

Copy

* fontname is a name you give the font which will become the name for the font-family: CSS attribute for elements that should use this font
* filename.ttf is the url of the font download file in ttf format for Firefox, Chrome, Safari, and Opera
* filename.eot is the url of the font download file in eot format for Internet Explorer
* filename.svg is the url of the font download file in svg format for iPad and iPhone
* filename.woff is the url of the font download file in woff format for Firefox, IE and Chrome

**box sizing**

box-sizing: content-box|border-box|inherit;

Copy

|  |  |
| --- | --- |
| content-box | The specified width and height (and min/max properties) apply to the width and height respectively of the content box of the element. The padding and border of the element are laid out and drawn outside the specified width and height |
| border-box | The specified width and height (and min/max properties) on this element determine the border box of the element. That is, any padding or border specified on the element is laid out and drawn inside this specified width and height. The content width and height are calculated by subtracting the border and padding widths of the respective sides from the specified 'width' and 'height' properties |
| inherit | Specifies that the value of the box-sizing property should be inherited from the parent element |

* supported in Internet Explorer, Chrome, and Opera.
* Firefox requires the prefix -moz-.
* Safari requires the prefix -webkit-.

div.box {

box-sizing:border-box;

-moz-box-sizing:border-box; /\* Firefox \*/

-webkit-box-sizing:border-box; /\* Safari \*/

width:50%;

border:1em solid red;

float:left;

}

Copy

**attribute selection**

* applies a style to an element with a specific attribute
* Example: Apply background color to textboxes

input[type="text"] {

background-color: #CCCCFF;

}

Copy

**:not**

* inverse selection
* p:not(.special) would apply to all paragraphs without class="special"

### Multiple Background Images

CSS3 allows you to specify multiple background images for an element. Supported by most current browsers.

**format**

background:

url(filename) [position] [size] [repeat] [attachment],

url(filename) [position] [size] [repeat] [attachment],

. . .

url(filename) [position] [size] [repeat] [attachment];

Copy

* url is the only required attribute
* images are stacked in the order listed with the 1st one on top and last one on bottom
* background color should be included as a separate item
* good idea to include fallback for browsers that don't support multiple backgrounds

background: url(image1.jpg) 100% no-repeat center center;

background:

url(image1.jpg) repeat-x top left,

url(image2.jpg) repeat-x bottom left,

url(image3.jpg) 100% 100% no-repeat center center;

background-color: black;

Copy

A picture containing logo

Description automatically generated