AMERICA ON TECH

2019 - 2020 HTML, CSS & JS Reference Tables

Anything orange is code that always stays the same. Wherever you see black text with a gray background you can choose what to type.

Syntax Reference

What is syntax?
In coding, **syntax** is the set of rules that describe the combination and sequence of symbols (including letters and numbers) that form a correctly structured program for a specific language.

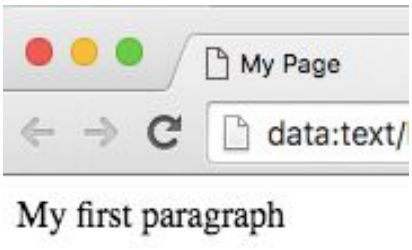
Symbol	Name	Example 1	Example 2
/	Forward Slash	`<body></body>`	`<img src="https://brit.co/1.jpg">`
-	Hyphen	`font-size: 20px;`	`$("#two").css("background-color","10px");`
" "	Quotes	`<img src = "awesome.jpg">`	`$("#div1").hide();`
< >	Angle Brackets	`<head> </head>`	`<!DOCTYPE html>`
{ }	Curly Brackets	`p { color: blue; }`	`function wrongAnswer() { $("#result").show(); }`
[ ]	Square Brackets	`var favColor = colors[1];`	`var colors = ["red", "blue", "yellow"];`
( )	Parentheses	`$("h1").hide();`	`wrongAnswer();`
;	Semicolon	`var word = "hello";`	`wrongAnswer();`
:	Colon	`#two { font-size: 20px; }`	`#two { width: 300px; }`
.	Dot	`$(".yourclass").text("hi");`	`.yourClass { color: red; }`
#	Hashtag	`$("#yourID").text("hi");`	`#yourID {color:red;}`

Comments	
Comments allow you to include information for other coders and are ignored by the computer.	
`<!-- These are comments in the code. -->`	Add a comment in HTML
`// One line of comments.`	Add one line comment in JavaScript
`/* Type a long section in the comments */`	Add a section of comments in JavaScript and CSS

HTML

Basic Structure of an HTML document (or webpage)

```
<!DOCTYPE html>
<html>
  <head>
    <title>My Page</title>
  </head>
  <body>
    <p>My first paragraph</p>
  </body>
</html>
```



HTML Element

an individual component of a webpage

Opening Tag ↓ `<p>` Content ↓ `This is a paragraph` Closing Tag ↓ `</p>`

HTML Elements		Code Example	Output
paragraph	`<p></p>`	`<p>This is a paragraph.</p>`	This is a paragraph.

heading	`<h1></h1>`
	`<h3></h3>`
	`<h6></h6>`
		`<h1>Heading level 1</h1>`
		`...`
		`<h6>Heading level 6</h6>`
			Heading level 1
			Heading level 6

ordered list (with numbers)	`<ol>`
	`  <li></li>`
	`</ol>`
		`<ol>`
		`  <li>George Washington</li>`
		`  <li>John Adams</li>`
		`</ol>`
			1. George Washington
			2. John Adams

unordered list (with bullets)	`<ul>`
	`  <li></li>`
	`</ul>`
		`<ul>`
		`  <li>George Washington</li>`
		`  <li>John Adams</li>`
		`</ul>`
			• George Washington
			• John Adams

button	`<button></button>`	`<button>Click Me</button>`	Click me!
div	`<div></div>`	`<div>This is a div</div>`	This is a div
input**	`<input>`	`<input>`	

****Self-closing:** Does not have a closing tag.

Nesting and Indentation in HTML

In coding, **nesting** is when you put one tag completely inside another tag's content.

Indentation helps you organize your code and makes it more readable. Remember to indent (press the tab key) when you're nesting an element inside another.

```
<div>
  <h1>Weekday</h1>
  <p>Monday</h1>
</div>
```

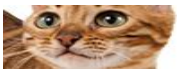
On the left, the `<h1>` and `<p>` tags are nested within the `<div>` tags.

HTML Attribute Syntax

An **attribute** adds extra information to an HTML element. In HTML syntax, attributes are part of an HTML opening tag.

Opening tag ↓ attribute ↓ Closing tag ↓

```
<a  href="www.google.com">Google it!</a>
```

HTML elements w/ attributes		Code Example	Output
Image **	`<img src=" ">`	`<img src="https://imgur/cats.png">`	
Link (anchor tag)	`<a href=" "></a>`	`<a href="https://www.google.com">This is a link to Google</a>`	This is a link to Google
Adding ids*	`id=" "`	`<p id="oneID">text</p>`	text
Adding classes*	`class=" "`	`<h1 class="aClass">text</h1>`	**text**
Input w/ placeholder**	`<input placeholder=" ">`	`<input placeholder="type here">`	type here

*You can add an `id` and/or `class` to any HTML element (`<img>`, `<a>`, `<li>`, `<ul>`, etc.)
****Self-closing:** Does not have a closing tag.

id vs. class

ids and **classes** are HTML attributes that you can add to HTML elements.

assign class in HTML	`class=" "`	`<div class="myClass">`	• The symbol that you use to select a `class` is a `.` (dot). • You can use the same `class` on multiple HTML elements. • You can use more than one `class` on the same HTML element • Classes are case-sensitive.
select class in CSS	`.`	`.myClass { text-align: right; }`	
assign id in HTML	`id=" "`	`<div id="myID">`	• The symbol that you use to select an `id` is a `#` (hashtag). • Each HTML element can only have one `id`. • Each page can only have one HTML element with that `id`. • ids are case-sensitive.
select id in CSS	`#`	`#myID { color: blue; }`	

CSS

CSS Syntax

```
1
img {
  2
  height: 30px;  3
  border: 1px solid red;
}
```

1. **Selector:** Identifies the parts of your page that will be affected by this CSS rule. You can select using the tag name, id, or class.
2. **Property:** The thing you want to change for the element(s) you've selected. Each property should be followed by a **:** (colon) .
3. **Value:** What you want to set this property to. Each value should be followed by a **;** (semicolon) .

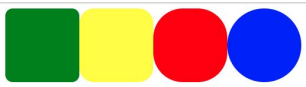
CSS Properties and Values

Change ...	Code Examples	Output	What it does

text	`font-family: "Comic Sans";`
	`font-size: 12px;`
	`text-align: center;`
	`color: blue;`
		hello	Changes the **font** to `Comic Sans`.
			Changes **font size** to `12` pixels.
			Aligns the text to the `center`.
			Changes the **font color** to `blue`.

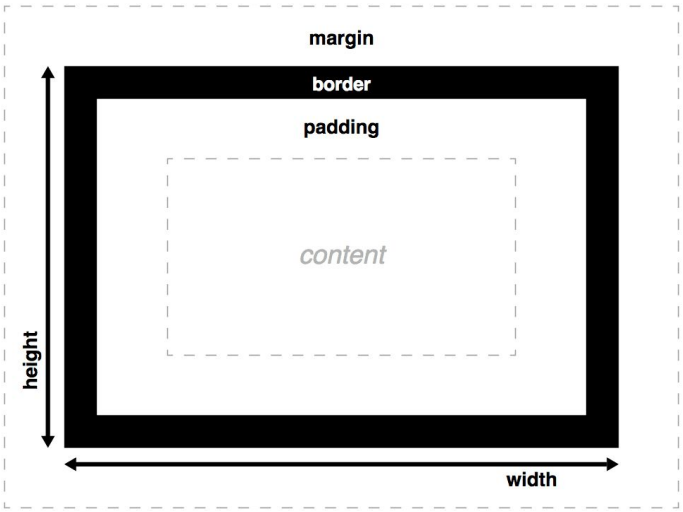
color	`background-color: #000000;` `color: yellow;`	hello	Changes the **background color** to the hex code #000000, which is black. Changes the **font color** to a specific shade of `yellow`.
background	`background-color: pink;` `background: url("ex.png");`	hello	Changes the **background color** to `pink`. Changes the **background to an image** w/ URL "`www.ex.png`"

size	`width: 50px;`
	`width: 50%;`
	`font-size: 20px;`
			Changes the **width** to `50` pixels.
			Changes the **width** to `50%` of the screen, whatever the size.
			Changes the **font-size** to `20` pixels.

border-radius	`border-radius: 500px;`		Makes the corners of a div slightly rounded
opacity	`opacity: 0.5;`		Make the whole div and all its content semi-transparent. Accepted values can be between 0 and 1.

CSS Layout

CSS Box Model



All HTML elements are shaped like boxes.

Each box has a content area (text, image, link, etc.) and optional surrounding padding, border, and margin areas.

Change...	Code Examples	What it does
content	`<p>hey</p>` `<img src="cat.jpg">`	Any HTML element (paragraph, image, link, etc.). *Not a property*.
padding	`padding: 20px;`	Spacing between the content and border.

border	`border: 20px solid red;`
	`border: 10px dotted yellow;`
	`border: 50px groove red;`
		Surrounds the padding. Think of it like an outline around a picture.
		Border takes 3 values that define how thick the border is, the style, and the color.

margin	`margin: 15px;`	Spacing between the border of this element and the start of another element.
If we define only one value, it will be applied to all 4 sides of the content.		
`padding: 10px;`		10px padding applied to all sides
We can define a different value for all 4 sides (top, right, bottom, left).		
`margin: 10px 20px 30px 40px;`		10px margin to **top** of content, 20px margin to **right** of content, 30px margin to **bottom** of content, 40px margin to **left** of content
You can define a value for a specific side of the property.		
`padding-left: 100px;` `margin-top: 25px;`		100px padding to the left only 25px margin to the top only
Similarly, you define a border for a specific side of the box.		
`border-right: 10px solid black;`		10px solid black border to the right only

`border-bottom: 20px dotted green;`	20px dotted green border to the bottom only

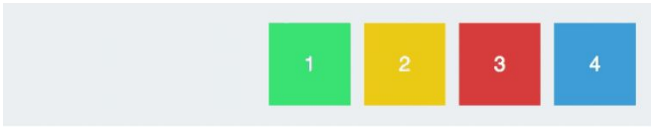
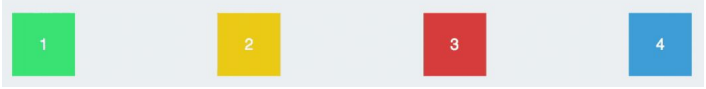
CSS Flexbox

When using flexbox, turn on **flexbox** for the parent element, using the property `display` and value `flex`.

```
.container {
  display: flex;
}
```

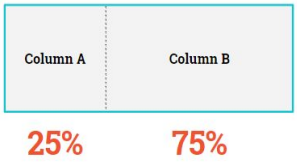
Arranged in a row

Use the **justify-content** property to align the child elements to a specific side.

Change...	Code Examples	What it does
flex-start	`.container { display: flex; justify-content: flex-start; }`	
center	`.container { display: flex; justify-content: center; }`	
flex-end	`.container { display: flex; justify-content: flex-end; }`	
space-between	`.container { display: flex; justify-content: space-between; }`	
space-around	`.container { display: flex; justify-content: space-around; }`	

Arranged by columns

Step 1: Turn on flexbox for the parent element (see above).
Step 2: Define the width for the child elements.




```
.section {
  display: flex;
}
.left {
  width: 25%;
}
.right {
  width: 75%;
}
```