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;`
		`}`
			`if(password === "1234") {`
			`  $(".result").show();`
			`}`

[ ]	Square Brackets	`let favColor = colors[1];`	`let colors = ["red", "blue", "yellow"];`

( )	Parentheses	`$("h1").hide();`	`for(let song of playlist){`
			`  $("ol").append(` ${song} `);`
			`}`

;	Semicolon	`let word = "hello";`	`colors.push("purple");`

:	Colon	`.two {`
		`  font-size: 20px;`
		`}`
			`#two {`
			`  width: 300px;`
			`}`

.	Dot	`$(".yourclass").text("hi");`	`.yourClass {`
			`  color: red;`
			`}`

`	Backtick	`$(".message").append(`Hi`);`	`$("h2").text(`Hi, ${name}. Welcome`);`

Comments

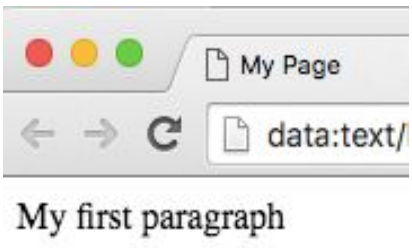
Comments allow you to include information for other coders and are ignored by the computer.

`<!-- 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>`	

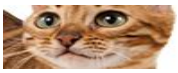
****Self-closing:** Does not have a closing tag.

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.	 Weekday Monday 	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 ↓ Google it!

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 classes*	`class=" "`	`<h1 class="thisClass">text</h1>`	**text**
Input w/ placeholder**	`<input placeholder=" ">`	`<input placeholder="type here">`	type here

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

Class Attributes

classes are HTML attributes that you can add to HTML elements.

assign class in HTML	`class=" "`	`<div class="myClass">`

select class in CSS	`.`	`.myClass {`
		`  text-align: right;`
		`}`

Assign multiple classes	`class="  "`	`<li class="important busy">`

- 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.
- Classes cannot start with a number (ex. class="1st" will not work)

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) .

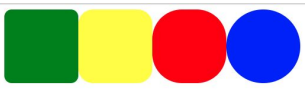
CSS Properties and Values

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`.

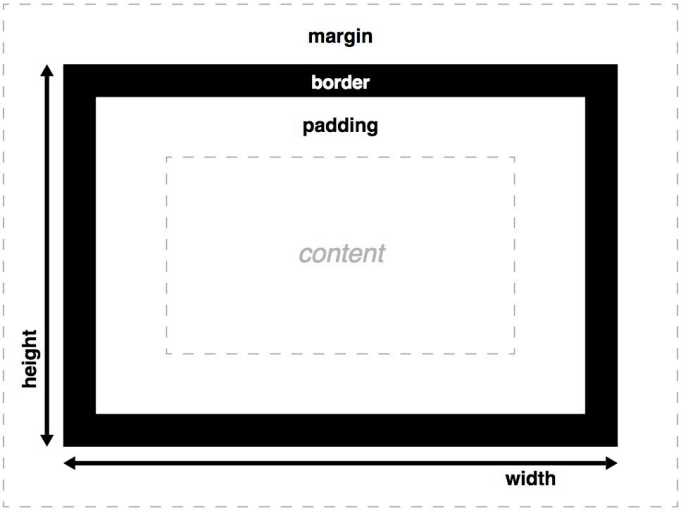
color	`background-color: #000000;` `color: yellow;`	hello	Changes the **background color** to the hex code #000000, which is black. Changes the **font color** to a specific shade of `yellow`.
background	`background-color: pink;` `background: url("ex.png");`	hello	Changes the **background color** to `pink`. 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. 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.). *This is not a property.*
padding	`padding: 20px;`	Spacing between the content and border.

border	`border: 2px 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: 150px;`	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;` `border-bottom: 20px dotted green;`	10px solid black border to the right only 20px dotted green border to the bottom only	

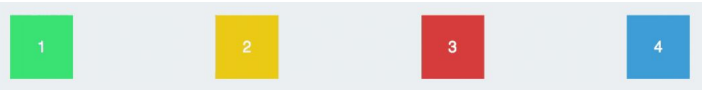
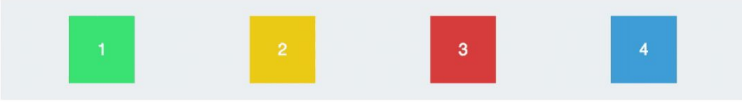
CSS Flexbox

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

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

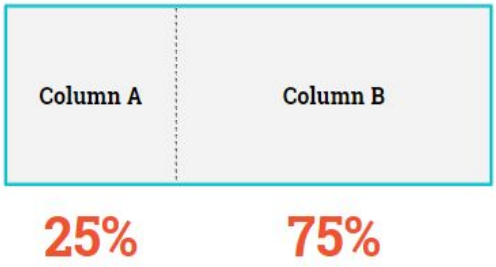
Arranged in a row

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.

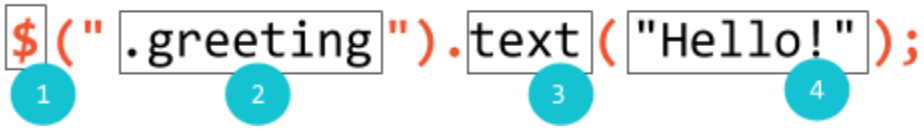



```
.section {
  display: flex;
}
.left {
  width: 25%;
}
.right {
  width: 75%;
}
```

jQuery

jQuery Syntax

jQuery is a JavaScript library with different actions that make it easier to make your page interactive.



1. The $ symbol lets the computer know that you are using jQuery, the JavaScript library.
2. The selector is exactly like a CSS selector. It selects or identifies the element on the page. You can use the name of an **HTML element** (, `<h1>`, `<body>`) or **class** (`.results`, `.div1`).
3. The jQuery action() to be performed on the element. See more options below.
4. The argument tells more information about how to change the element. Sometimes, there is no argument, i.e. `.show()`, and sometimes, there are several arguments, i.e. `.css()`.

Click Handler

```
1 $(".yourClass").click(function(){
2   $("img").hide;
3 });
```

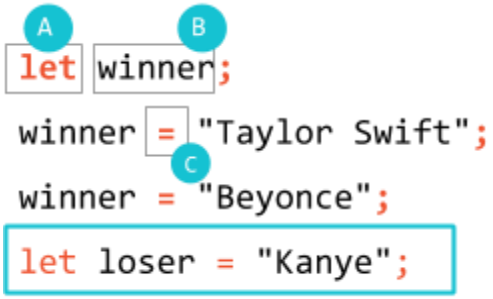
1. When the user clicks the HTML element with a class `yourClass`
2. The HTML element `img` hides.
3. Closes the click handler.

Action	Code Example	What it does
Show an element. **Hide an element.**	`$(".yourClass").show();` `$(".myClass").hide();`	Show all HTML elements with the class `yourClass`. Hide all HTML elements with the class `myClass`.
Replaces the content of an HTML element.	`$("body").html(` Hi! `);`	In the HTML, replace the content inside the `<body>` with `<p>Hi!</p>`.
Add/change the CSS, or style, of an element. (Change the property and/or value)	`$(".container").css("color", "red");`	Add/change the CSS property `color` to `red` for all HTML elements with a class of `container`.
Add/change the text in an element.	`$(".results").text("You won!");`	Add/change the text to `"You won!"` for the HTML element with the class `results`.
Add/change an HTML attribute. (See page 4 for info about attributes.)	`$("img").attr("src", "http://pics.com/blah.jpg");`	Add/change the HTML attribute `src`, or source, to that URL for all `<img>` tags.
Append (add) content to an element.	`$("div").append("Bye!");`	Append, or add, the text `"Bye!"` to the **end** of the all the `<div>` tags.
Retrieve a value from an <input>	`let firstName = $("input").val();`	Retrieve a value from the input tag and store it in a variable named `firstName`.

JavaScript

Variable Syntax

Variables are containers for storing data values.



Parts:

A. The keyword `let` declares a variable, or creates a new variable.
B. The variable name `winner`
C. The equal = sign assigns a value.

Line 1: Declares a variable and gives it the name, `winner`.
Line 2: Assigns a value to the variable `winner`.
Line 3: Re-assigns a different value to the variable `winner`. The value of `winner` is no longer `"Taylor Swift"`. It is now `"Beyonce"`.
Line 4: Declares a variable named `loser` and assigns it a value `"Kanye"` all in one line of code!

Value Types

Number	Duh... you know what a number is... No quotation marks, may start with a + or -, may include a decimal.	`let temperature = -1;` `let price = 5.99;`

String	Always inside single (' ') or double ("") quotes. Can be an empty string (" "). Can include letters, spaces, symbols, numbers... as long as it's in quotes.	`let greeting = "Joliz is here!";`
		`let space = '  ';`
		`let price = "5.99";`

Array	A list of multiple values separated by commas inside square brackets []	`let oddNumbers = [1, 3, 5, 7, 9];` `let airport = ["JFK", "LGA", "SFO"];`

Input and .val

JavaScript	Code Example	What it does
input field	`<input class="username">` `<button class="login">Go!</button>`	Creates an input field in HTML with a class, `username`. Creates a button that says Go! with a class `login`

input.val	`$(".login").click(function(){`
	`  let message = $(".username").val();`
	`});`
		When the user clicks the HTML element with an class `login` (which is the button), retrieve the value from the input field and store it in variable `message`.

String Interpolation

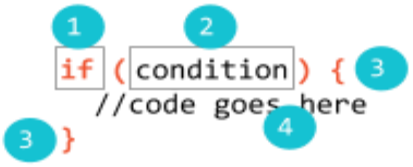
String interpolation allows you to combine strings and variables to build HTML in JavaScript.

Code Example	What it does
`let userName = "codeNationStudents";` `$("p").append(`Hello ${userName}.`);`	Appends the string "Hello codeNationStudents" to all the paragraph tags. The value stored in variable `userName` is combined with the string using the `` `${ }` `` notation.
`let newItem = $("input").val();` `$(".toDo").append(` ${newItem} `);`	Retrieve the value from the input field and store it in variable `newItem`. Append the value of `newItem` as a new list item to an element with class `toDo` using string interpolation to create a new HTML element.

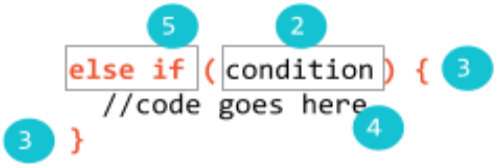
Conditional Syntax

Conditional statements are used to perform different actions based on conditions.

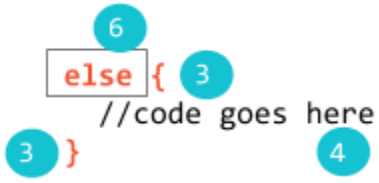
1) if statement



2) else-if statement



3) else statement



Conditional Statements can be created using a combination of the three statements on the left.

1. The keyword `if` indicates that this is an if statement
2. The condition goes between the ( ); the result should be true or false. If you need multiple conditions, you will need an else-if statement.
3. Curly brackets indicate the body of the condition statement.
4. Body - This is the code that executes if the condition is true. If the condition is false, then the code will NOT execute.
5. The keyword `else if` indicates an else-if statement.
6. The keyword `else` indicates an else statement. No condition with an else statement

An if statement is required to create a conditional statement, however an else-if statements and else statements are as needed. You can also use more than one else-if statement.

Basic Conditional Statement Example

```
1  let number = 3;
2  if (number < 5) {
3    $(".btn").hide();
4  } else {
5    $(".btn").show();
6  }
```

1 Declare variable named number and assign it a value of 3.
2 If the variable number is less than 5...
3 Hide the HTML element with the class `btn`.
4 Or else...
5 Show the HTML element with the class `btn`
6 End of conditional statement.

Conditional Statement with Multiple Conditions Example

```
1  let num = 11;
2  if (num < 5) {
```

1 Declare variable named num and assign it a value of 11.
2 If the variable value is less than 5...

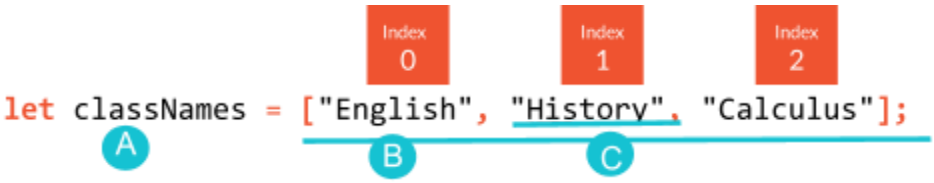
3
4
5
6
7
8
	`     console.log("Less than 5");`
	`} else if (num < 10)  {`
	`     console.log("Less than 10");`
	`} else {`
	`     console.log("Greater than 10");`
	`}`
		3
		4
		5
		6
		7
		8
			Print "Less than 5" to the console
			Else if the number is less than 10...
			Print "Less than 10" to the console
			Else
			Print "Greater than 10 to the console
			End of conditional statement.

Compound Conditional Statement Example

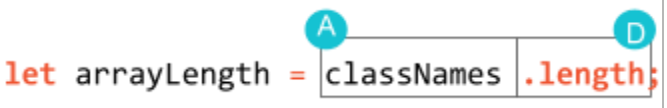
1
2
3
4
5
	`if (age > 16 && passedTest===true) {`
	`    console.log("you can drive.");`
	`} else {`
	`    console.log("no driving yet.");`
	`}`
		1
		2
		3
		4
		5
			If the value of age is greater than 16 AND(&&) passedTest is true
			Log "you can drive." to the console.
			Else
			Log "you can't drive." to the console
			End of conditional statement.

Array Syntax

An array is a way to store more than one value at a time. Think of it like a list.



A. Declare a variable called `classNames`.
B. An array is a list of values — they can be numbers, strings, or a combination of different value types. Square brackets start and end an array.
C. Each array element, or individual item (i.e. `"History"`) in the array, is separated by a comma.

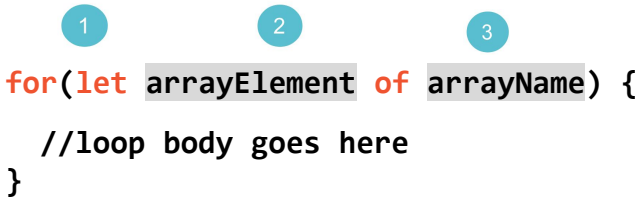


D. Arrays have properties that you can use, including length. Use the name of the array, in this case, `classNames` + `.length` to represent the length. The length of this array is 3, because there are 3 total elements in this array. The value of `arrayLength` is 3.

E. To use a specific array element, use the array index. It (see above) represents the location of an array element and always begins with 0. The array index uses the name of the array + [the index surrounded by square brackets]. The value of `favElement` is `"English"`.

for of Loop Syntax

Loops repeat an action. A for of loop repeats an action until all elements in an array have been selected.



1. The for of loop is used to iterate over an array. It can be any array with any number of values or array elements
2. Iterating over an array means looping over the elements of the array and selecting each element one at a time. This variable represents a selected array element . You can name this variable anything that you want.
3. Identify which array you are going to iterate over with the keyword of and the array name
4. The for of body goes between the curly brackets. This block of code executes every time an element is selected from the array. Usually the code is doing something to the array element that is currently selected.

For Of Loop Example

```
1 let courses = ["history", "math", "science"];
2 for(let course of courses){
3   $(".schedule").append(`<p> ${course} </p>`);
4 }
5
```

1 Create an array to iterate over.
2 Use a for of loop to iterate over the array.
3 The variable course represents the array element that is currently selected. The first time the loop runs `course` represents "history", the second time it is "math", and the third time it is "science".

Mathematical Operators**		
Symbol	**Definition**	**Code Example**
+	Addition****	`a + b;`
-	Subtraction	`a - b;`
*	Multiplication	`a * b;`
/	Division	`a / b;`

** Follow the order of operations rule **PEMDAS**: 1) Parentheses, 2) Exponents, 3) Multiply/Divide, 4) Add/Subtract
****Can *ALSO* be used to concatenate, or combine, strings -- not just add numbers.

Comparison Operators		
Symbol	**Definition**	**Code Example**
<	Less than	`if (number < 10)`
>	Greater than	`else if (grade > 70)`
<=	Less than or equal to	`if (points <= 100)`
>=	Greater than or equal to	`else if (age >= 16)`
===	Equal to	`if (username === "scripted1")`
!==	NOT equal to	`else if (password !== "p@$sw0rd")`

Logical Operators		
Symbol	**Definition**	**Code Example**
&&	And	`if (resume > true && interview < true)`
||	Or	`if (grade > 65 || passedRegents)`
!	Not	`if (!(number < 10))`