



# **CodeFirst:Girls Beginners Coding course - Front end Web development**

## **Week 6 - JavaScript & jQuery**

#### What have we learned so far

- How the Web works
- How to display content using HTML
- How to style content using CSS
- How to collaborate using GitHub & Git
- How to make coding easier using Twitter Bootstrap

We have reached the final stretch of the of our Introduction into Web Development course.

#### *What have we learned so far*

- We had a brief introduction how the Web works;
- We learned how to display content online in a structured way using HTML;
- We learned how to style and design that content using CSS;
- We learned how to work collaboratively on a website using GitHub and Git;
- And we learned about tools developers use to make their coding lives easier in the form of Twitter Bootstrap.

So what's next? We are going to dive into programming for the web by learning about JavaScript and jQuery.

## WHAT WE COVERED IN WEEK 5 - TWITTER BOOTSTRAP

What did we learn last week?

1. What's hard in CSS and why do we need to fix it?
2. Twitter Bootstrap - what it is, what problems it fixes, how to use it
3. Modifying Bootstrap
4. Even more awesome stuff!(changing social buttons, etc)

**Task:** Find a partner and together take a quick look through the notes from last week's session. If you're unclear on any of the concepts, work through them with your partner and an instructor

### Bootstrap recap

**Divide the class into pairs, and get them to discuss what we learned in the last lesson. Then ask one person from each pair to tell the whole class what they discussed. If the main topics are not all brought up, bring them up with questions**

### What is programming?

- A way for humans to tell computer to do logical things
- So that humans can interact with each other using machines
- Programming languages are organised in paradigms = ways of thinking and communicating with a computer

## But wait! What even is Programming?

1. In short programming is a way for humans to tell computers to do logical things in a systematic fashion, so that they can interact with each other using computers.
2. Programming languages are organised into paradigms, that is ways of thinking and communicating with a computer.

There are many different programming languages, but to decide which one to use, you first need to decide what you are going to use it for. You need to pick the programming language that is best suited for the use.

### Robot exercise

**Introducing, the new Instructorbot 3000! This is a very fancy robot and can, by itself, listen to commands, breathe, walk, turn and pick things up. The students task is to write a program which will enable the Instructorbot to pick up the mug from the desk at the front of the class.**

**Place a mug/similar item on the table a the front of the class, then stand in the**

corner, facing the wall. Then carry out whatever command the students shout out.

The purpose of this exercise is to teach the difference between writing and program, and remote controlling a robot. Be as obtuse with the directions as possible. For example, if someone says 'turn around', turn around and don't stop (as they never gave a conditional for when you should stop)

The same with a command to 'walk forward'. Walk forward until you start (gently) crashing into tables.

You may, after a while end up with a successful array of commands that goes something like:

- turn 180 degrees
- walk forward three steps
- pick up mug.

Explain that this is remote controlling. To make it a real program, it needs to be executable from anywhere in the room. This is when we add if statements.

The correct(ish solution) as as follows:

- turn *until* you see mug
- walk forward *until* you can reach mug
- pick up mug

### JavaScript is powerful

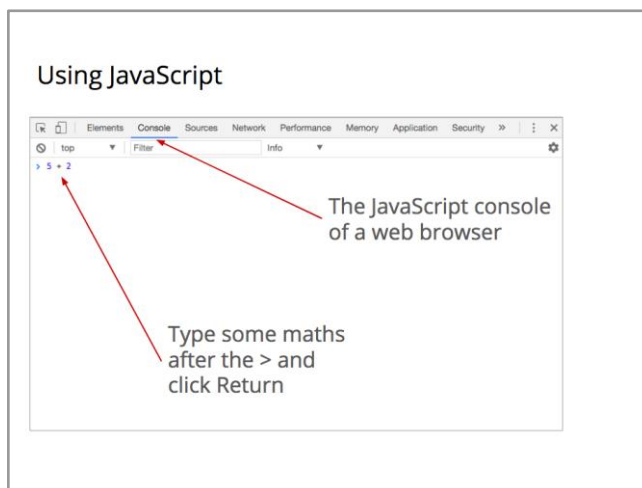
- It can accommodate a range of thinking styles
- Things you define can be changed easily
- It is designed to be used for the Web
- It is a full-stack language - it can be used both client and server sides

## JavaScript is powerful

As I already said JavaScript is a programming language, and it's powerful because:

- **Multi-paradigm:** it can accommodate a range of thinking styles
- **Prototype-based:** things you have defined can be changed easily
- **It's designed to be used for the Web:** unlike other languages which were designed for specific computing
- **It's a full-stack language:** that means it can be used both on the client side, the browsers people use on their devices, and the server side, where the files for a website are hosted. Other languages like Ruby or Python are server-side only.

So as you can see, JavaScript is super flexible and therefore awesome.



## JavaScript & jQuery

### So... how do I start using JavaScript?

You may be wondering now how you can start using JavaScript. Because JavaScript is a standard web technology you can write it directly in your browser.

If ever you are in need of a calculator and all you have is a browser, you can use that.

1. Open up the Developer Tools in your browser, doesn't matter which website you are currently viewing.
2. Go to the Console tab and you'll see a little prompt with a blinking cursor.
3. Type a mathematical equation. For example  $5+2$ , and then click Enter/Return.

4. And there you go, your browser has added it up for you.

Numerical operations, comparators & strings

- **Numerical operations:** +, -, \*, /, %
- **Comparators:** ==, !=, <, >, <=, >=, booleans
- **Strings:** ordinary text

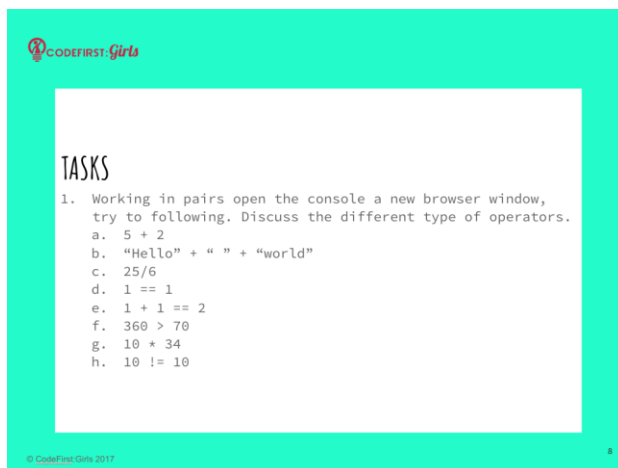
## Numerical operations, comparators and strings

That was a very simple calculation, but JavaScript can do a lot more than that.

- **Numerical operations:** Additions, subtraction, multiplication, division and percentages.
- **Comparators:** a little more involved. These compare two values to see if they are equal, not equal, smaller, bigger, equal or smaller, equal or bigger or if they are true or false.
- **Strings:** the easiest one of all. A string is a piece of ordinary text.



All of these can be used together to write JavaScript. Let's have a look at the main elements we will be using.



The slide features a red header with the 'CODEFIRST:GIRLS' logo. The main content is a white box with the title 'TASKS' in bold. Below the title is a numbered list of tasks. Task 1 instructs students to work in pairs, open a browser console, and try the following expressions, discussing the operators used. The expressions are: a. `5 + 2`, b. `"Hello" + " " + "world"`, c. `25/6`, d. `1 == 1`, e. `1 + 1 == 2`, f. `360 > 70`, g. `10 * 34`, and h. `10 != 10`. The footer of the slide includes the copyright notice '© CodeFirst Girls 2017' and a small number '8'.

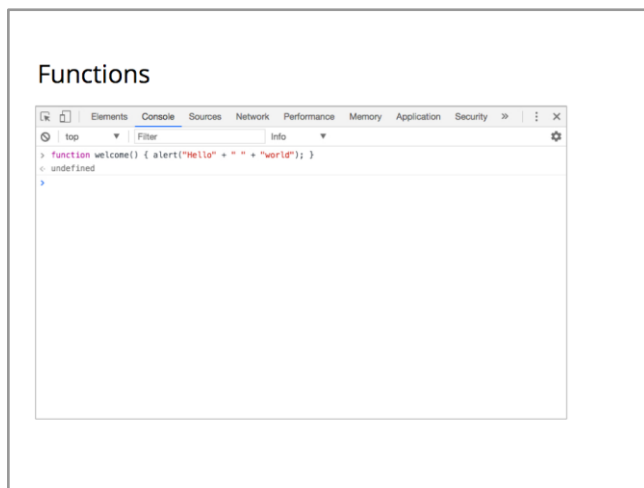
**TASKS**

1. Working in pairs open the console a new browser window, try to following. Discuss the different type of operators.
  - a. `5 + 2`
  - b. `"Hello" + " " + "world"`
  - c. `25/6`
  - d. `1 == 1`
  - e. `1 + 1 == 2`
  - f. `360 > 70`
  - g. `10 * 34`
  - h. `10 != 10`

© CodeFirst Girls 2017 8

These are all different types of **expressions**. An expression is an operation performed on a data type. Lets work through these in pairs in the console and then discuss them.

**Ask students to work in pairs for this task. Get them to open the console in chrome and work through each of expressions. When completed, go through this as a class, discussing the operators how they work.**

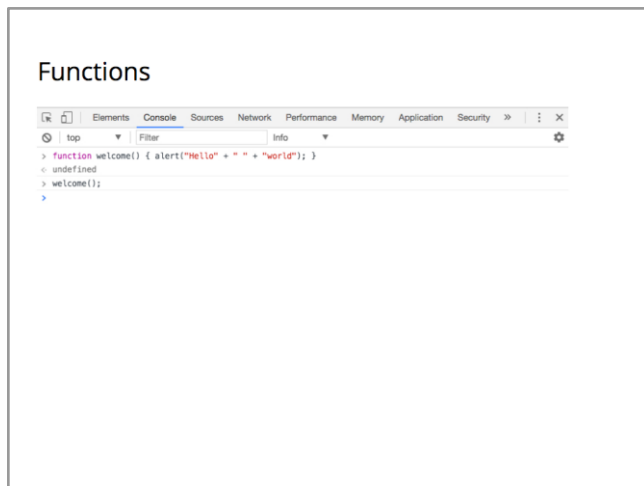


## Functions

A function is a group of expressions which come together to perform a particular task. This function is called by an action. An action is an event. Actions can be made by a user, a server, or some other event the program is listening to.

Type this in your Developers Tools Console. You may have to put it all on one line. After typing it in press Enter. Ignore the undefined error you get afterwards. We're not done yet.

```
function welcome() {  
    alert("Hello" + " " + "world");  
}
```



You created a function. If you want to call the function type this on the next line:

`welcome();`

This should create an alert box at the top of your browser window with **Hello world**.

## VARIABLES

```
> var name = "Harry"
< undefined
> name
< "Harry"
>
```


```
> var name = "Harry"
< undefined
> name
< "Harry"
> var name = "Ron"
< undefined
> name
< "Ron"
> name = " Ron Wesley"
< "Ron Wesley"
> |
```

12

## Variables

A variable is a what JavaScript uses to store, organise and manage raw data. Variables are used by expressions and functions.

Here we can see that the variable name was originally set as "harry", however, that was then overwritten with the name "ron". We can call that variable by simple typing 'name' and we can add to it accordingly by types name + something.



## TASKS

1. Working in pairs open the console write in this function, and then call it with different 'parameters' each time! What is the purpose of the variable in this function? How can you break the function?

```
function addTogether(oneThing, anotherThing) {  
  var newThing = oneThing + anotherThing  
  alert(oneThing + " + " + anotherThing + " = " + newThing)  
}
```

© CodeFirstGirls 2018 13

Again, lets work in pairs to create and play around with this function. Copy this into your console and play around with it.

**After 5 minutes, work through the task and go through the function and explain it, line by line. This is also a good opportunity to discuss the use of cases, for example - headless camel case in this instance.**

### Using JavaScript in a web page

Add it directly in your HTML pages in between `<script>...</script>` tags.

```
<!DOCTYPE html>
<html>
<head>
...
</head>
<body>
...
<script>
...
</script>
</body>
</html>
```

## How do you use JavaScript in a web page

So far we have been typing our JavaScript in a the console tab of our Developers Tools. That's fun to do, but that's not an efficient way of adding JavaScript power to web pages.

There are two better ways of adding it:

- Write the JavaScript directly into your HTML documents in between `<script>...</script>` tags.

### Using JavaScript in a web page

Link to a separate file like you would with CSS

```
<!DOCTYPE html>
<html>
<head>
...
</head>
<body>
...
<script src="js/scripts.js"></script>
</body>
</html>
```

RECOMMENDED

- **The recommended way** is writing your code in a separate file and linking to it like you do with a CSS file

You link to custom js files in the bottom of your HTML documents because usually the functions you write in there relate to HTML elements. When a browser loads a web page

it will do so from top to bottom. If it loads your script file before it has seen the HTML elements it refers to, it won't do anything.

#### What is jQuery?

- JavaScript is complicated to learn and a little verbose
- jQuery is a library containing ready-made JavaScript functions you can use out-of-the-box
- It will speed up development
- Unlike JavaScript it does not come with every browser
- You need to link to the jQuery library from your website pages

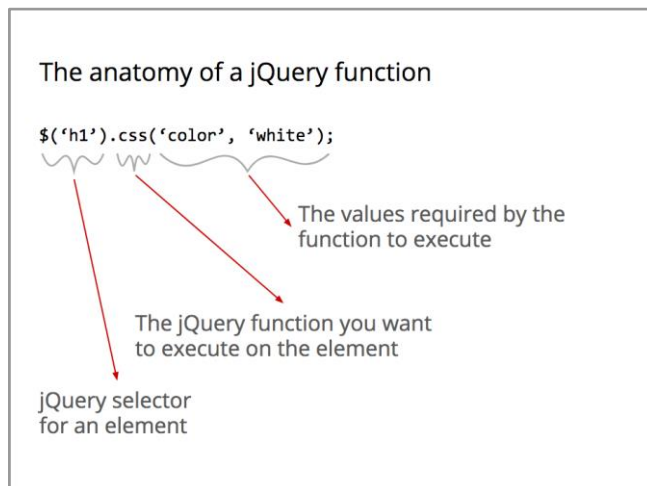
## And what is jQuery?

The little examples we tried are very straightforward, but JavaScript can get a lot more complicated. It's also quite verbose when writing basic functionality.

You remember how much easier and faster it became to style a website when we added Twitter Bootstrap? Well, jQuery is a JavaScript library full of ready made functions to build interactive web pages.

jQuery is so common now that for beginners, learning JavaScript had become learning jQuery. And that is what we are going to do as well.

But there is a little catch. Unlike JavaScript, jQuery does not come standard with every browser. Because it is a library you need to link to that library in your web pages in order to use the functionality. jQuery is not the only JavaScript library, but it is one of the more popular ones.

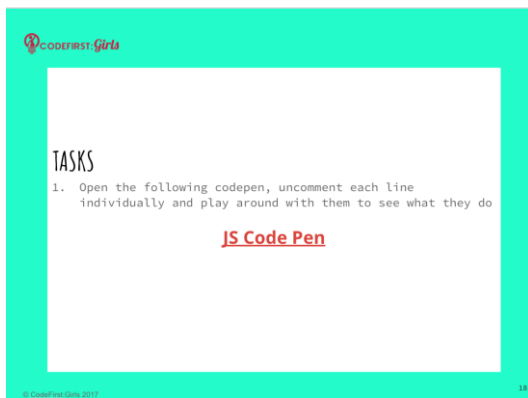


## The anatomy of a jQuery function

To manipulate elements on a web page with jQuery you need to select them. And for that jQuery uses this function: `$( ' ' )`. The `$` sign indicates that it is jQuery, and the bit between the quote marks is where you place the element you want to select.

The selectors for elements are the same as the ones we have been using in CSS.

The bit after the element selector will be a jQuery function you want to execute on the element. In this case it will manipulate the CSS. Inside the `css` brackets we can then write our CSS declaration block by splitting the property and value, adding single quotes around each and separating them with a comma.



## ACTION

Go to this code pen <https://codepen.io/rosie934/pen/yQaKKG?editors=1010#0> and talk the students through the examples. You will need to uncomment the JS lines as you go along.

1. You can see we have a list of fruit. First thing we can do is hide the title. The jQuery command first selects the `<h1>` element, and then it uses the function `hide()`. The result is that the `<h1>` is now hidden.
2. In the same way as we hiding the element, we can show it too. And as you might have guessed, the function to show something is `show()`.

You may now start to think that jQuery is easy. And you're not wrong. Remember jQuery is a library of functions. Behind these simple functions of `hide()` and `show()` there is a larger function using plain JavaScript to make these actions work. If we were learning



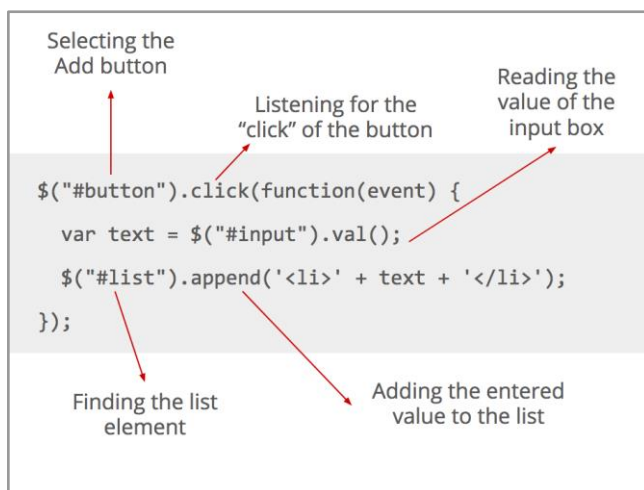
JavaScript instead of jQuery, we would need quite a few more lines of code to achieve the same thing.

3. Let's crack on with manipulating some CSS and change the colour of the `<h1>` element. This time the function is `css` and in between the brackets we are telling jQuery what colour we want the text to be.
4. You can change any CSS property with jQuery. For example font-size.
5. Or the text alignment.

In practice it is better to use plain CSS to change the style of your web pages. But this is to show you how jQuery works and it's fun.

6. Back to some more useful functions which CSS can't do. How about fading out the `<h1>` element? And as you might guess, the function for that is `fadeOut()`.
7. Let's fade it back in. Yep, that is done with `fadeIn()`.

The best thing to use jquery is conditional formatting, so when someone clicks on an elements, something else happens.



The first thing this function does is select the **Add** button in the HTML. The button has an **ID** of **button** and this makes it unique on the page. By using the id you know that your jQuery won't mistakenly select the wrong element.

In the next part the **click** function is called. Click is an event function, and what it does is that it listens to the click of that button. And when it's clicked it will execute the commands inside the function. Two things are happening inside this function:

A variable is created to collect what someone has typed inside the box. The variable declaration includes another jQuery selector element. This time it's selecting the value

of the **input** element. Again it's using the id of that element. The **val** function fetches the value of an element.

The second line of code is will be add the entered text to the list below the input box. This time the selector is pointing to the list element's id name. Followed by the **append()** function. Append will add something at the end of an element. If you were wondering, yes there is a **prepend()** function too.

Inside this **append()** function you need to write exactly what you will be appending. As the list is an HTML list that means you need to include the correct **<li>** tags, and in between them you place the value of the text variable. You'll recognize the operators we saw at the beginning of this section.

And that's it. The start of a To Do list app.

Using the jQuery CDN

**<http://code.jquery.com>**

- Click the most recent and minified version
- Copy the link from pop-up window
- Paste link inside your HTML `<head>...</head>` tags

```
<!DOCTYPE html>
<html>
<head>
...
<script src="http://code.jquery.com/jquery-3.2.1.min.js"></script>
</head>
...
```

Similar to Bootstrap, we could either download the whole framework or link to it through a CDN.

## Using the jQuery CDN

When we used Bootstrap it came with a warning of the potential dangers that come with using a CDN. But in this instance we are going to link to the jQuery CDN to add it to our projects.

**CODFIRSTGIRLS**

## TASKS

1. Fork the project for week 6 and download it into your coding folder
2. Add jquery using a cdn: make sure it is called above the other js files
3. Read through the code on the 'background.js', try and complete the function which will change the background when the button is clicked
4. Extension task: using variables, change the text in the span to be the name of the artist when the background changes

© CodeFirstGirls 2017

## Task

Found here: <https://github.com/CodeFirstGirls/beginners-week-six>

This last task is quite challenging, so it may help to work in pairs again.

Fork the week 6 project down to you desktop and follow the README to take you through the steps. Make sure you load jquery into the right place before you carry out the task.

Similarly, like the other challenges, the solution is available on the 'solution' branch on github (it's not cheating to peak!)

## **Homework**

Carry on with with challenge, add some JS to your first site and carry on with your group projects!

## **Useful resources**

And that's all folks.

What remains is for you to start putting all this knowledge into practice and code all the things. If you get stuck Google it. There is a good chance that the problem you are trying to solve has already been solved by someone else who had the same problem. A good source for solution is Stack Overflow. Other good resources are W3C Schools, and CSS Tricks. But there are many more. Above all, have fun.