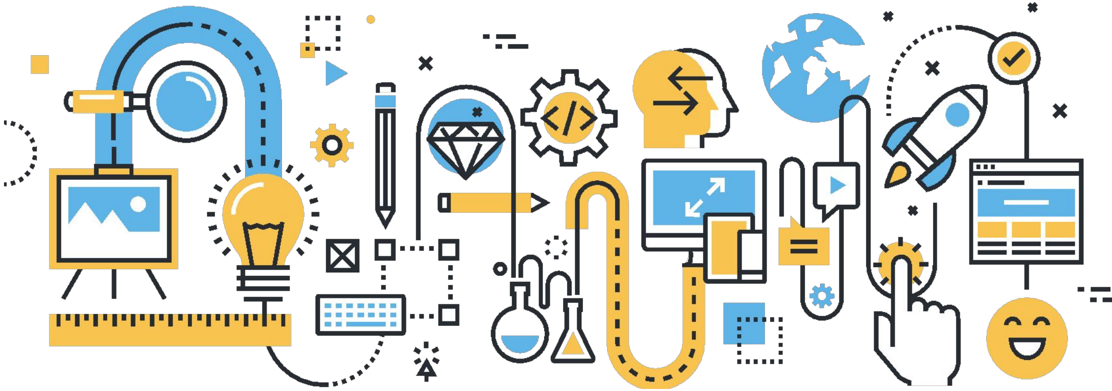
WEEK 8

ID

jQuery

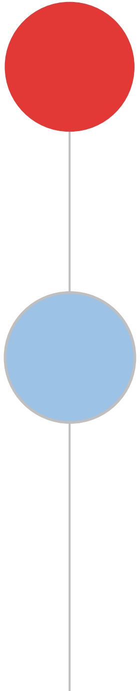
Make things faster

2020



INTERACTIVE DEVELOPMENT

jQuery



Traversing with Jquery

What is jQuery

Selecting with jQuery

Working with jQuery

Working with elements

jQuery functions

Scripts

STORY: SPEED

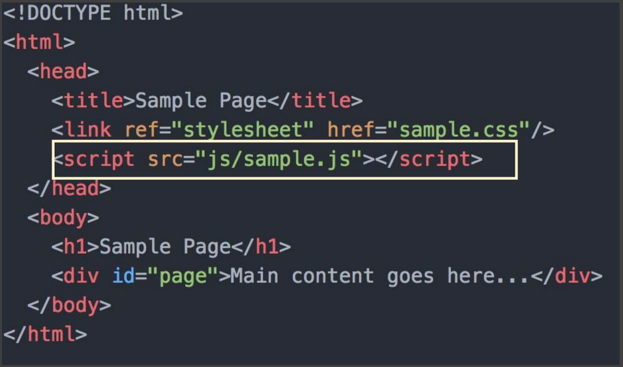
In the early days of the web, developers were told to place the *<script>* tags in the <head> of the page as you do with stylesheets. However this make pages seemed slower to load.

Your web page may use files from several different locations (files from CDN, fonts from third party places).

When a browser starts to download a JavaScript file, it stops all downloads and pauses to lay the page until the script has finished loading.

Therefore, if you place the script at the end of the page before the closing </body> tag, it will not affect the rendering of the page.

WHERE TO PLACE YOUR SCRIPTS

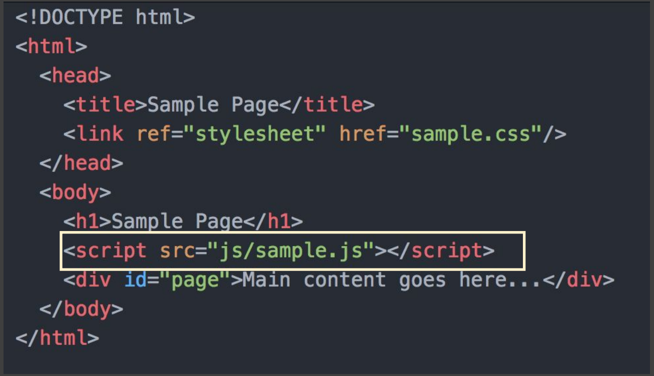


```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Page</title>
    <link ref="stylesheet" href="sample.css"/>
    <script src="js/sample.js"></script>
  </head>
  <body>
    <h1>Sample Page</h1>
    <div id="page">Main content goes here...</div>
  </body>
</html>
```

IN THE HEAD

- Best Avoided
- Pages seem slower to load
- DOM content is not loaded when the script is executed so you have to wait for an event like load or DOMContentLoaded to trigger your functions

If you muse use a <script> element with the head of the page, use it just before the closing </head> tag

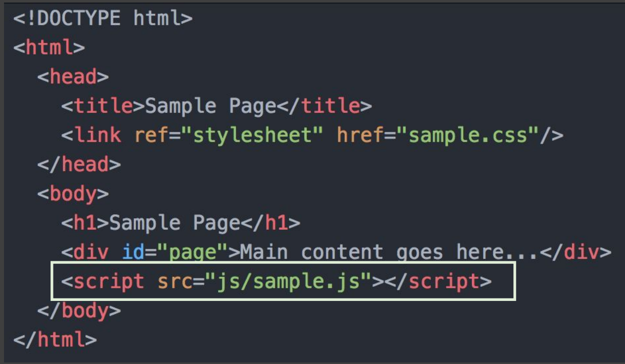


```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Page</title>
    <link ref="stylesheet" href="sample.css"/>
  </head>
  <body>
    <h1>Sample Page</h1>
    <script src="js/sample.js"></script>
    <div id="page">Main content goes here...</div>
  </body>
</html>
```

IN THE PAGE

- Best Avoided
- Pages seem slower to load

Certain JavaScript calls might not work as you could be referencing elements above the script.

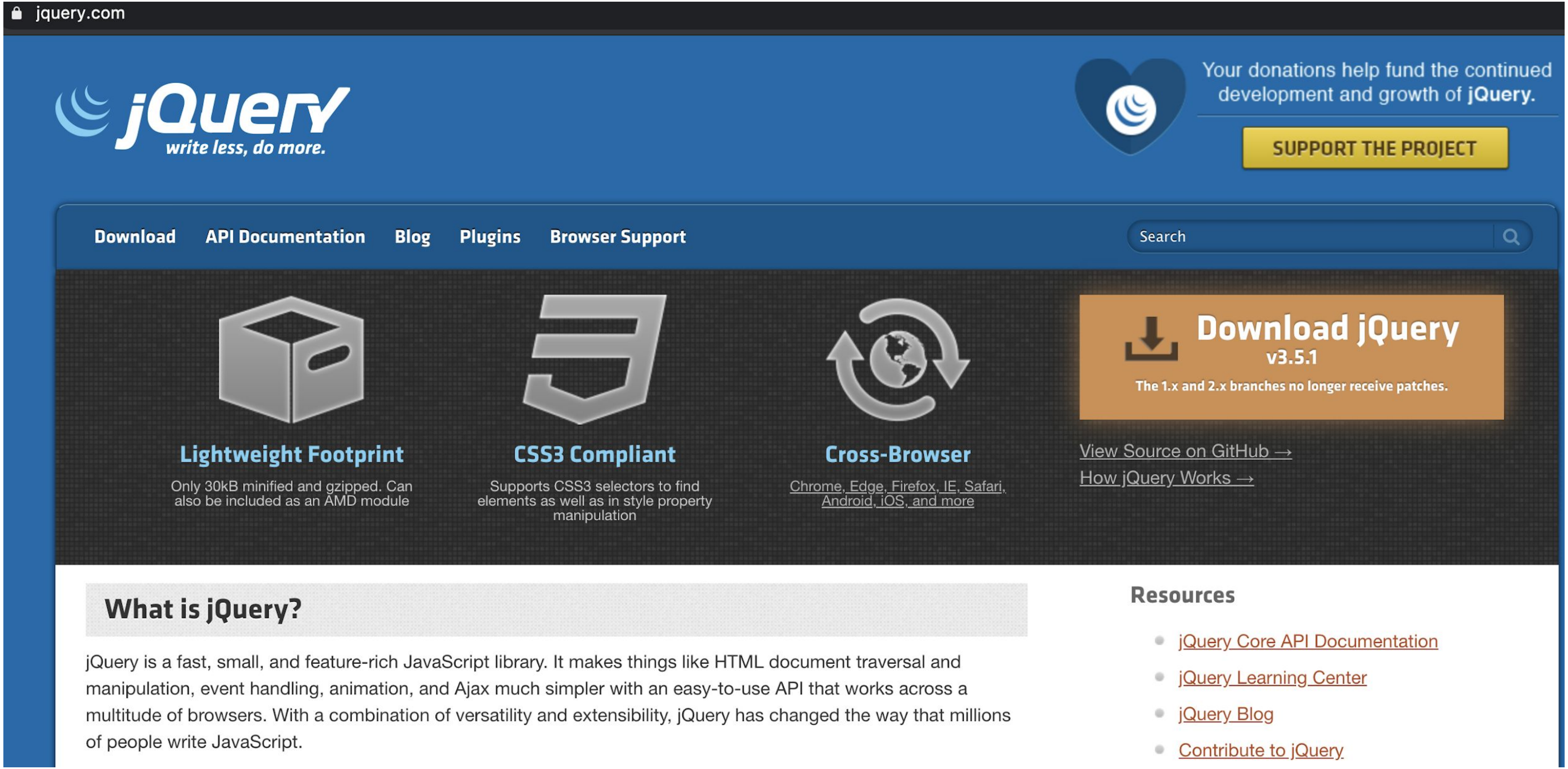


```
<!DOCTYPE html>
<html>
  <head>
    <title>Sample Page</title>
    <link ref="stylesheet" href="sample.css"/>
  </head>
  <body>
    <h1>Sample Page</h1>
    <div id="page">Main content goes here...</div>
    <script src="js/sample.js"></script>
  </body>
</html>
```

BEFORE THE CLOSING </body> tag

- Script is not blocking other things from downloading
- The DOM has already loaded by the time the script is executed

jQuery



Your donations help fund the continued development and growth of **jQuery**.

Download jQuery
v3.5.1

The 1.x and 2.x branches no longer receive patches.

Lightweight Footprint

Only 30kB minified and gzipped. Can also be included as an AMD module

CSS3 Compliant

Supports CSS3 selectors to find elements as well as in style property manipulation

Cross-Browser

Chrome, Edge, Firefox, IE, Safari, Android, iOS, and more

What is jQuery?

jQuery is a fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers. With a combination of versatility and extensibility, jQuery has changed the way that millions of people write JavaScript.

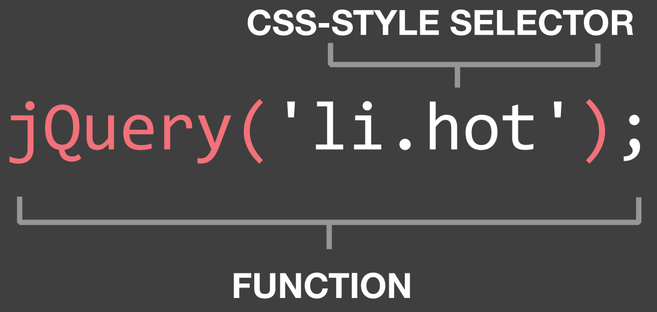
Resources

- jQuery Core API Documentation
- jQuery Learning Center
- jQuery Blog
- Contribute to jQuery

https://jquery.com/

1

Select elements
using CSS-style
selectors



CSS-STYLE SELECTOR

```
jQuery('li.hot');
```

FUNCTION

SHORTHAND FOR JQUERY FUNCTION

```
$('li.hot');
```

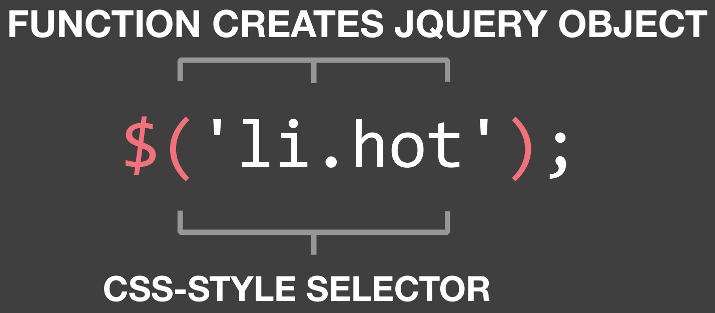
NORMAL JS

```
document.querySelectorAll('li.hot');
```

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

When you select an element or set of elements, it creates a **jQuery object.**

That object contains *references to* the elements.



FUNCTION CREATES JQUERY OBJECT

```
$('li.hot');
```

CSS-STYLE SELECTOR

The jQuery() function has one parameter: a CSS-style selector

This selector finds all of the <li> elements with a class of **hot.**

1

Select elements
using CSS-style
selectors

2

Do something
using methods of
the jQuery object

HIDE ALL MATCHING LIST ITEMS

```
$('li.hot').hide();
```

METHOD OF THE JQUERY OBJECT

This is roughly equivalent to calling **.css("display", "none")**

https://api.jquery.com/hide/

With jQuery, you can

✔ Select or find elements

✔ Update their content / size / visibility

✔ Simplify event handling

Basics of jQuery Statement

Selector	action	parameters
jQuery("#page")	.css	('background-color','green');
$("p")	.css	('background-color','green');

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

Selecting by HTML Tags

$("div") – selects all <div> elements
$("li") – selects all <li> elements
$("p") – selects all <p> elements

We have to select by CSS to narrow down

NORMAL JS

```
document.getElementsByTagName('div');
document.querySelectorAll('div');
```

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

Selecting by CSS/ ID

$("#page") = select any with id of 'page'
$(".hot") = select any with class of 'hot'
$("li.cool") = select **only** li with class of 'cool'
$("li.cool a") = select **only** <a> that are inside <li> with class of 'cool'

NORMAL JS

```
document.getElementById('page');
document.querySelectorAll('li.cool a');
```

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

Multiple Selectors

Use a comma separate different search criteria

$("li, p") - select all <li> and <p>

$("li.hot, p.content") --

select all <li> with a class of hot, and all <p> with class of content.

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

CSS: Getters

Use .css() to get the value of a css property.

```
// returns its color
$("#one").css('color');

// height property
$("#one").css('height');
```

CSS: Setters

You can change the css of a selected element:

```
//set multiple css values at once
$('div').css({
    'background-color' : 'green',
    'color' : 'white'
});


$('.cool').css('font-size', 20);
```

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

SETTING HTML CONTENT

This example will replace the content of each list item with the word **Updated** in <b> tags. This includes HTML markup

```
$('li').html('<b>Updated</b>');
```

NORMAL JS

```
document.getElementsByTagName('li').innerHTML
```

SETTING TEXT CONTENT

This example will replace the text content of each list item with the word **Updated**.

```
$('li').text('Updated');
```

NORMAL JS

```
document.getElementsByTagName('li').textContent
```

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

Changing Content

Use the *html()* action

```
$('#two').html('<a>Goodbye World!</a>');
```

NEW HTML VALUE

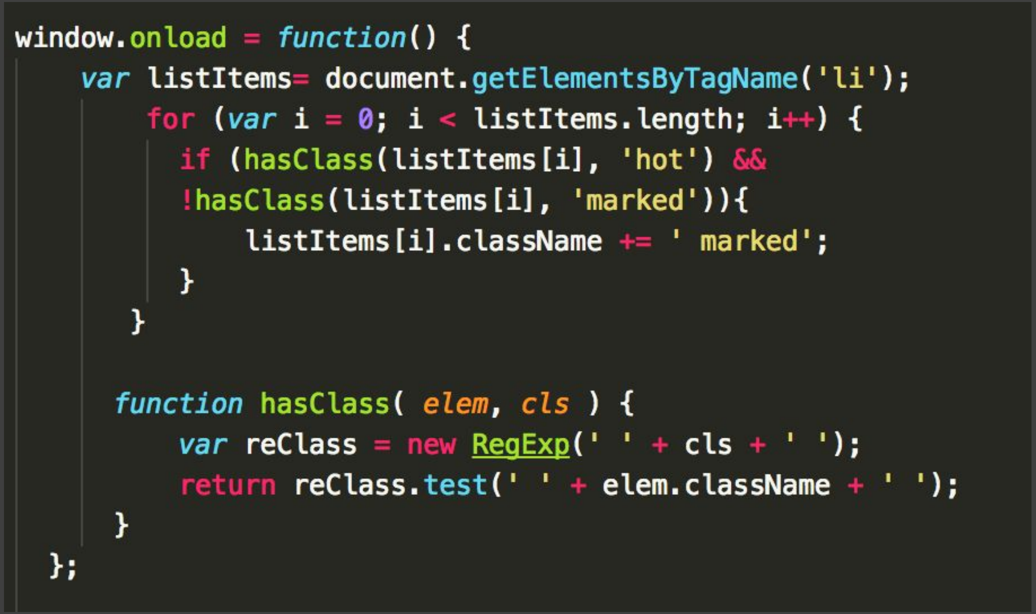
Get & Set

Attributes are those "x=y" inside tags
eg: <a **href**="…">, href is an attribute

```
$('a').attr('href') -- retrieve the value of href
$('a').attr('href', 'newvalue') -- set the value of
the href attribute
```

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

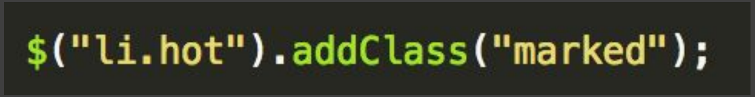
VANILLA



```
window.onload = function() {
    var listItems= document.getElementsByTagName('li');
        for (var i = 0; i < listItems.length; i++) {
          if (hasClass(listItems[i], 'hot') &&
          !hasClass(listItems[i], 'marked')){
              listItems[i].className += ' marked';
          }
       }

      function hasClass( elem, cls ) {
          var reClass = new RegExp(' ' + cls + ' ');
          return reClass.test(' ' + elem.className + ' ');
      }
  };
```

With jQuery



```
$("li.hot").addClass("marked");
```

We save a number of steps and convenience

https://repl.it/@malcolmyam/simple-jquery-demo#script.js

Loading JS when page is ready

Use $(document).ready();

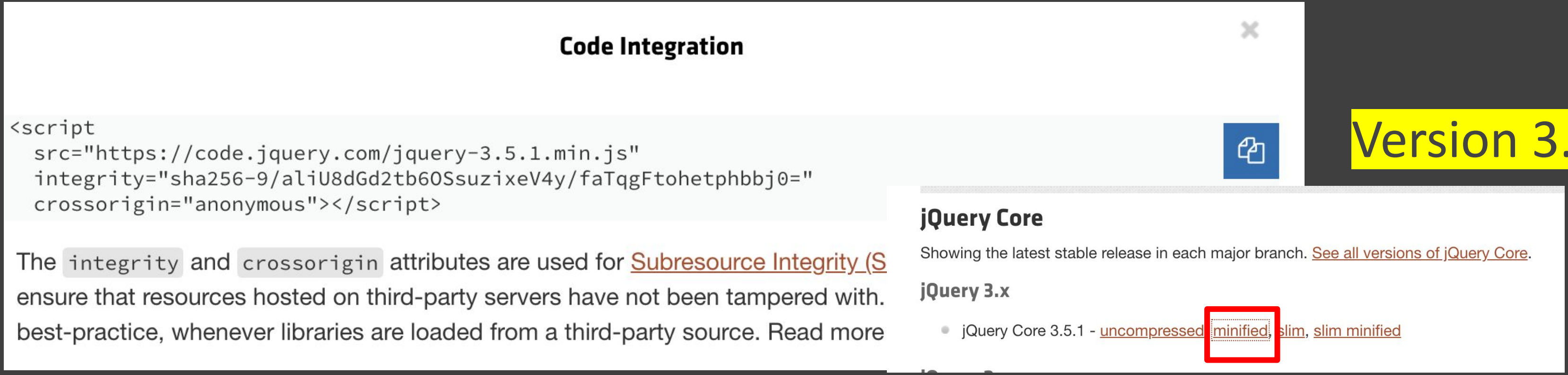
```
$(document).ready(function() {
   console.log("Hello World!");
});

$(function() {  //shorthand version
   console.log( "ready!" );
});
```

Why we are doing this?
HTML elements must be loaded and “safe” before we can safely manipulate them $(document).ready() allows us to safely detect that the page is all ready for us to use, then we can execute our JS code

Further Reading:
https://learn.jquery.com/using-jquery-core/document-ready/

How to Use?



Code Integration

```
<script
  src="https://code.jquery.com/jquery-3.5.1.min.js"
  integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
  crossorigin="anonymous"></script>
```

The `integrity` and `crossorigin` attributes are used for Subresource Integrity (S
ensure that resources hosted on third-party servers have not been tampered with.
best-practice, whenever libraries are loaded from a third-party source. Read more

jQuery Core

Showing the latest stable release in each major branch. See all versions of jQuery Core.

jQuery 3.x

- jQuery Core 3.5.1 - uncompressed, minified, slim, slim minified

Version 3.5.1

Insert the jQuery script code into your HTML file. We will use the minified version as it's a compressed version of the jQuery code. Which means it's lighter in size. You can download the file and save it into your local machine too. Here, we are referencing to the jQuery code online. This will only run when we have Internet Connectivity.

In this instance, we don't install jQuery. We simply add the jQuery script to our HTML code. Ensure that jQuery is inserted first before your own JS code.

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"
  integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="
  crossorigin="anonymous"></script>
```

https://code.jquery.com/

ACTIVITY

https://repl.it/@malcolmyam/wk08-template10#index.html

#TODO CA1

Simply fork or download template_10.zip and use VSC to test out your script

Follow the instructions on the tutorial deck and complete question 1-2

You only need to create a .docx file of your answers to the questions. Label each part of your answer properly with the question number.

Go to the tutorial Google Docs

SUBMISSION
Create folder **wk08-template-10** and place your codes for this exercise in a **.docx** file

Compile all the activities for the week and submit as a ZIP file
wk08-studentid-studentname.zip file
Name each activity properly
Eg.
/wk08-studentid-studentname
../wk08-template-10
../wk08-simple-click
../wk08-circles
../wk08-randomuser
../wk08-weather
../wk08-about-me

DOM TRAVERSAL

Parent.child.sibling

It describes all the *HTML tags* as a **tree structure**

TL;DR: The Document Object Model (***DOM***) is a programming interface for HTML, XML and SVG documents. It provides a structured representation of the document (a tree) and it defines a way that the structure can be accessed from programs so that they can change the document structure, style and content.

Traversing

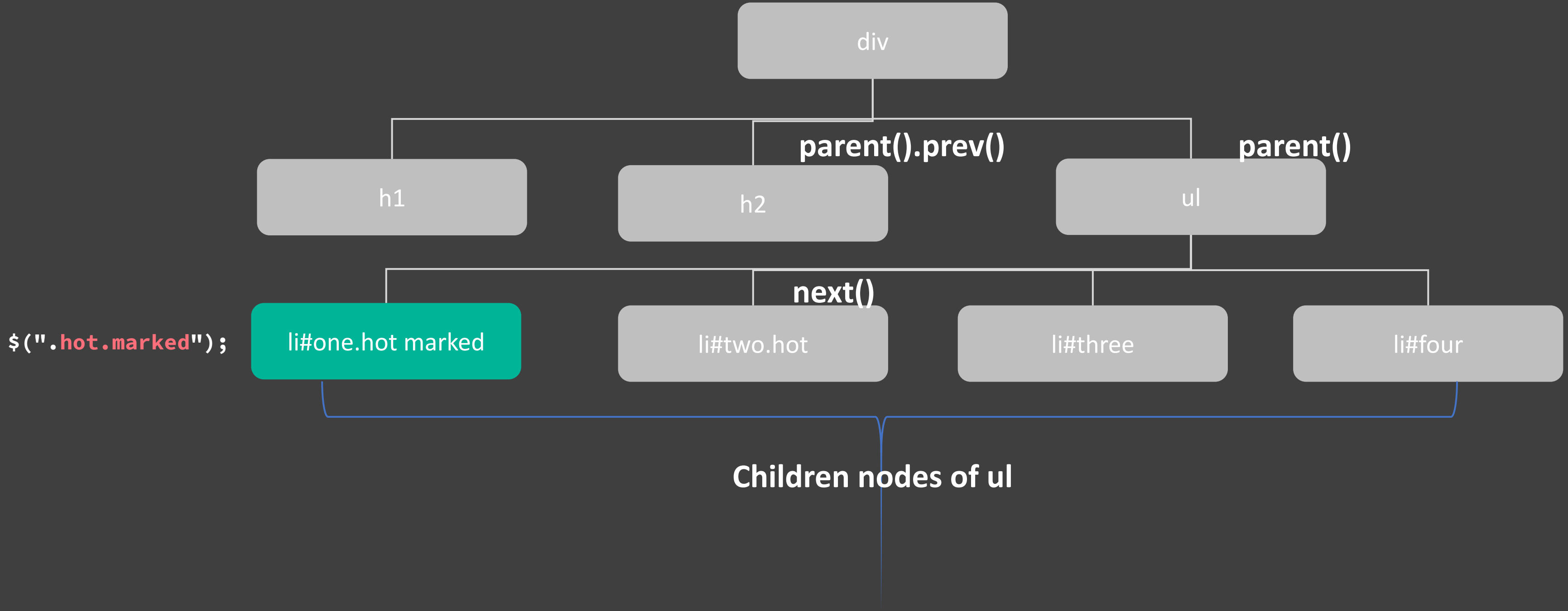
After you have select an element

- use `prev()` to move to one sibling element before itself
- use `next()` to move to one sibling element after itself
- use `parent()` to go to its parent
- use `children()` for all children
  - use `children().first()` for the first child
  - use `children().eq(n)` for the index element, or "*n* + 1" child element ,
  - where n is a number

Reading reference: https://api.jquery.com/eq/
https://api.jquery.com/eq-selector/

#DOM
#jQUERY

```
$(".hot.marked").next();
$(".hot.marked").parent();
$(".hot.marked").parent().prev();
```



DOM TRAVERSAL METHODS

```
.find()
.parent()
.parents()
.next()
.nextAll()
.closest()
.children()
.siblings()
.prev()
.prevAll
```

https://api.jquery.com/category/traversing/

Remove

Use **remove()**to remove the element from the DOM

Note: does not remove the actual HTML element in the HTML document itself. Only temporary

https://api.jquery.com/remove/

Clone

The **clone()** method makes a copy of selected elements, including child nodes, text and attributes.

html

```
<div id="target">
  I'm a small box
  <div class="smallBox">I’m very small</div>
</div>
```

jQuery

```
$('#target').clone().insertAfter('.sm
allBox');
```

```
<div id="target">
  I'm a small box
  <div class="smallBox">I'm a small small inner box</div>
</div>
 <div class="smallBox">
   I'm a small box
  <div class="smallBox">I'm a small small inner box</div>
</div>
```

https://repl.it/@malcolmyam/wk08-simple-jquery-clone#script.js

IMPORTANT

If your selector returns **more than 1** element, your **modification** will ***affect all*** of them.

Key Attributes

class
id
data

HTML5 allows custom attributes to be created
data-<custom name>

Used for attaching raw data
Better experience

```
<ul id="vegetable-seeds">
  <li data-spacing="10cm" data-sowing="March to June">Carrots</li>
  <li data-spacing="30cm" data-sowing="February to March">Celery</li>
  <li data-spacing="3cm" data-sowing="March to September">Radishes</li>
</ul>
```

Power of Data Attributes

You can add "variables" to a html5 element

```
// defines a data called tooltip
<a href="google.com" data-tooltip="Go to google">Google.com</a>

// define a data called highlight-color
<span special="highlight" data-highlight-color="green">Roll over me</span>
```

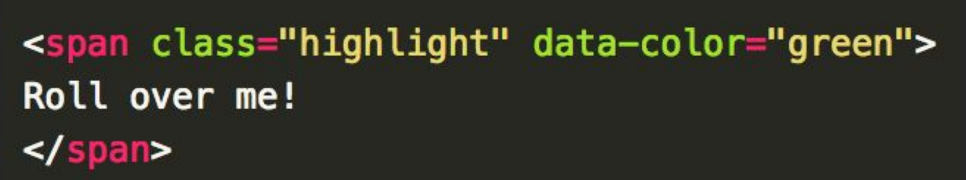
The DATA attribute

The **data attribute** let you stores information in a HTML element.
The name of the data is whatever that comes after the dash

- data-NRIC : name is "NRIC"
- data-phone-number : name is "phone-number"

https://repl.it/@malcolmyam/wk0-simple-data-attribute#script.js

Get & Set: Data



```
<span class="highlight" data-color="green">
Roll over me!
</span>
```

Getting

```
// returns "green"
$(".highlight").data("color");
```

Setting

```
$(".highlight").data("color",
 "red");
// returns red
$(".highlight").data("color");
```

https://repl.it/@malcolmyam/wk08-simple-data-attribute#index.html

Get & Set: Attribute

```
<span class="highlight" data-color="green">Roll over me!</span>
```

Getters

```
// returns "highlight"
$(".highlight").attr("class");

$(".highlight").attr("data-color");
```

Setters

```
$(".highlight").attr("class",
  "special");

$(".highlight").attr("data-color",
  "red");
```

#jQuery
#classes

hasClass()

You can use ***hasClass()*** to check if a HTML element has the specified class

https://repl.it/@malcolmyam/wk08-jquery-toggle#script.js

https://api.jquery.com/hasclass/

UPDATE // INSERT
GET // SET

UPDATE ELEMENTS

```
.html()             .text()
.replaceWith()      .remove()
```

```
$('li#one').remove();
```

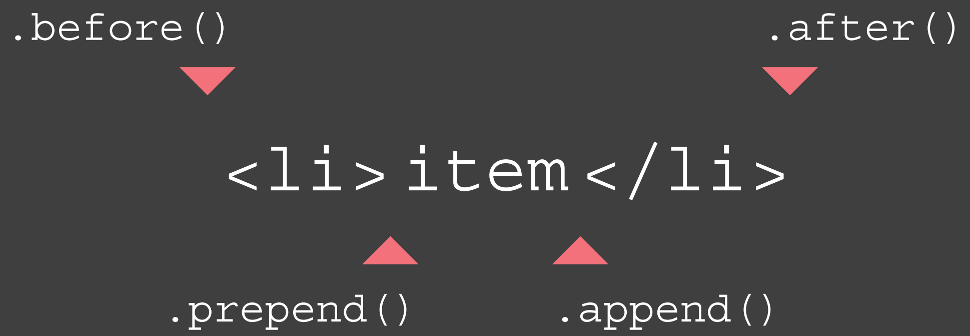
Creating new HTML elements

We can use jQuery to create new HTML element on the fly

We can then add it to the DOM

```
var n = $("<p>Hello World</p>");
$("body").append(n);
```

INSERT ELEMENTS



.before()

.after()

`<li>item</li>`

.prepend()

.append()

DEMO: https://repl.it/@malcolmyam/wk08-jquery-insertelements#script.js

Adding to the DOM

Alter the content of a document:

Use **`before()`** and **`after()`** to insert before/after the selected element
Use **`append()`** and **`prepend()`** to add at the end or at the start of the selected element

Append VS After

```
<p>Secret Level</p>

$('p').append('<p>Treasure
  Box</p>');

<p>Secret Level <p>Treasure
  Box</p></p>
```

```
<p>Secret Level</p>

$('p').after('<p>Treasure
  Box</p>');

<p>Secret Level</p>
<p>Treasure Box</p>
```

GET & SET ATTRIBUTES

```
.attr()
.addClass()
```

```
.removeAttr()
.removeClass()
```

GET & SET ATTRIBUTES

`.attr()`

This method can **get or set a specified attribute** and its value. To get the value of an attribute, you specify the name of the attribute in the parentheses.

```
$('li#one').attr('id');
```

To update the value of an attribute, you specify both the attribute name and its new value.

```
$('li#one').attr('id' , 'hot' );
```

GET & SET ATTRIBUTES

```
.removeAttr()
```

This method **removes** a specified attribute (and its value). You just specify the name of the attribute that you want to remove from the element in the parentheses.

```
$('li #one') .removeAttr (' id' );
```

GET & SET ATTRIBUTES

```
.addClass()
```

This method adds a new value to the existing value of the class attribute. It does not overwrite existing values.

GET & SET ATTRIBUTES

`.removeClass()`

This method removes a value from the class attribute, leaving any other class names within that attribute intact.

Creating new HTML elements

Alternate Method:

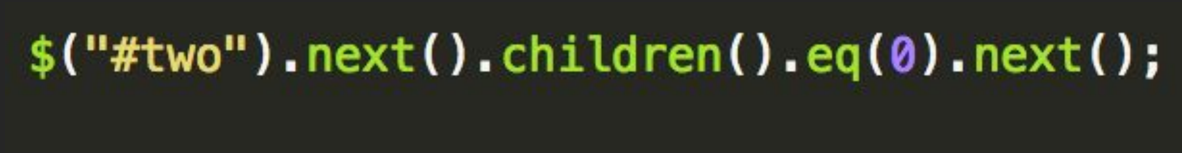
```
$("<p>Hello World</p>").appendTo("body")
```

Chaining: Powerful

It makes your code short and easy to manage.

It gives better performance.

The chain starts from left to right. So left most will be called first and so on.



```
$("#two").next().children().eq(0).next();
```

#DOM
#NODES

Live NodeList

When your script updates the page, the NodeList is updated at the same time. The methods beginning **getElementsBy ..**
Return live NodeLists.

They are typically faster to generate than static NodeLists

getElementById()
getElementsByClassName()
getElementsByTagName

Static NodeList

When your script updates the page, the NodeList is not updated to reflect the changes made by the script

querySelector… (which uses CSS selector syntax) return static NodeLists. They reflect the document when the query was made. If the script changes the content of the page, the NodeList is not updated to reflect those changes.

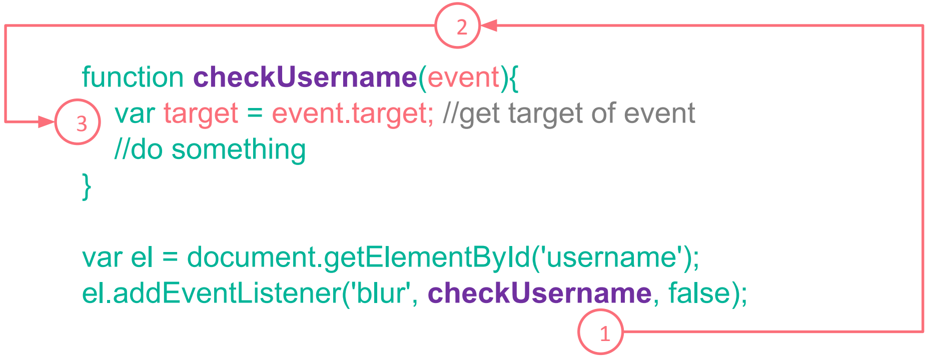
querySelector()
querySelectorAll()

Watch: Learn about querySelectors on LinkedIn Learning

Event Object

With jQuery

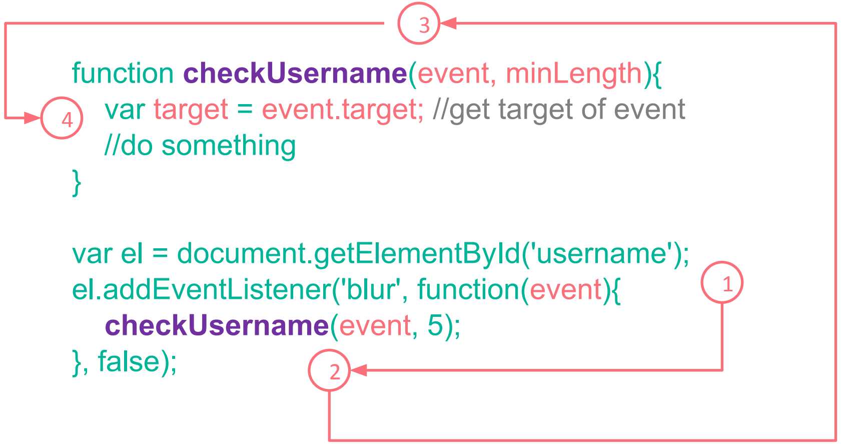
EVENT LISTENER WITH NO PARAMETERS (A)



```
function checkUsername(event){
   var target = event.target; //get target of event
   //do something
}

var el = document.getElementById('username');
el.addEventListener('blur', checkUsername, false);
```

EVENT LISTENER WITH PARAMETERS (B)



```
function checkUsername(event, minLength){
   var target = event.target; //get target of event
   //do something
}

var el = document.getElementById('username');
el.addEventListener('blur', function(event){
   checkUsername(event, 5);
}, false);
```

jQuery

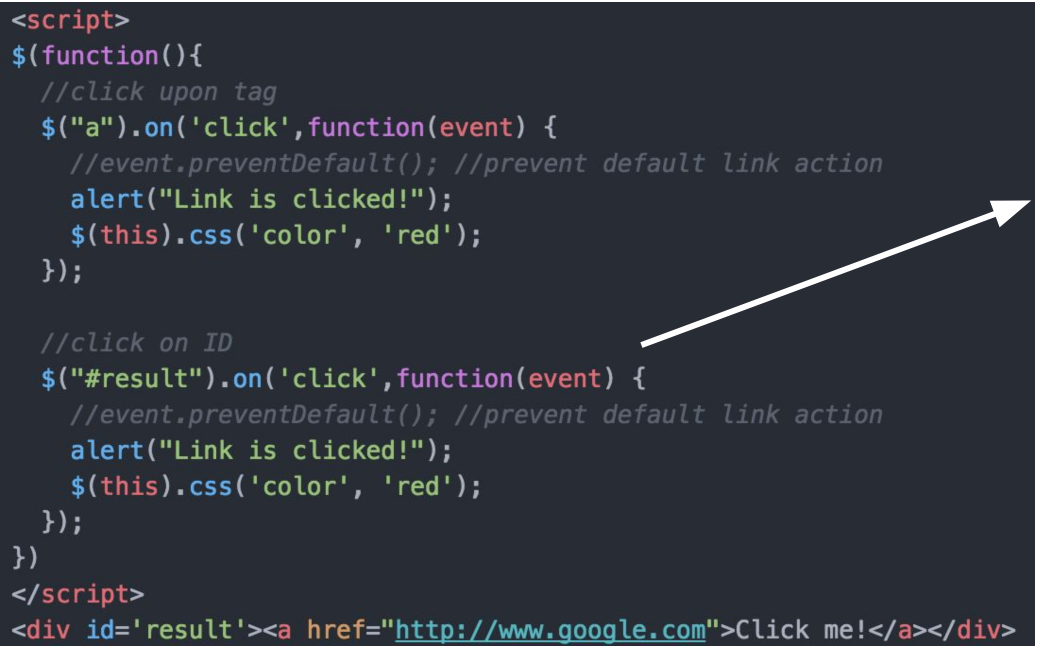
```
$("#username").on("blur", function(event){
  checkUsername(event)
});
```

jQuery

```
$("#username").on("blur", function(e) {
      checkUsername(e, 5)
});
```

https://repl.it/@malcolmyam/wk07-sample-username-check#index.html

Refer to activity/js/event-listener-with-event-object.js
https://repl.it/@malcolmyam/wk07-activity-materials#index.html



```
<script>
$(function(){
  //click upon tag
  $("a").on('click',function(event) {
    //event.preventDefault(); //prevent default link action
    alert("Link is clicked!");
    $(this).css('color', 'red');
  });

  //click on ID
  $("#result").on('click',function(event) {
    //event.preventDefault(); //prevent default link action
    alert("Link is clicked!");
    $(this).css('color', 'red');
  });
})
</script>
<div id='result'><a href="http://www.google.com">Click me!</a></div>
```

jQuery (Equivalent)

```
$("#result").click(function(e) {
   alert("Link is clicked!");
   $(this).css('color','red');
});
```

Sample source of event click using jQuery:
https://repl.it/@malcolmyam/wk08-simpleclick-on#script.js

https://api.jquery.com/category/events/mouse-events/
https://api.jquery.com/on/
https://api.jquery.com/click/
https://api.jquery.com/on/#additional-notes

SELECTING FORM ELEMENT

By assigning a id or a class (like normal HTML)

By selecting by name:

```
// select all input with the name "username"
$("input[name=username]")
```

RETRIEVE FORM VALUE

The value is what is stored in the input element
Use `.val()` to retrieve:

```
$("#username").val()
```

Use the same method, but pass in a parameter to set

```
$("#username").val("leeroy");
```

GET SELECTED CHECKBOXES

If we have a checkbox with id called checkbox

```
$("#checkbox:checked")
```

Returns **an array of all the checkboxes** that are clicked.

```
var chkStyle = [];
$.each($("input[name='chkStyle']:checked"), function(){
  chkStyle.push($(this).val());
});
```

//Looping through each element

CHECKBOX SELECTED?

If we have

```
<input type="checkbox" id="male">Male</input>
<input type="checkbox" id="female">Female</input>
```

Then we can check if the checkbox with id "male" is checked or not with

```
if ($("#male").is(":checked")) {
 console.log("Male is checked!");
}
```

Example - KeyUp

```
$("#username").keyup( function(e) {
 console.log($(this).val());
})
```

NORMAL JS

```
document.getElementById('#username').addEventListener('keyup',function
(e){

)
```

Detect change in select dropdown

Use the `.change()` event

```
$("#select").change(function(e){
 console.log("You have changed to " + $(this).value());
})
```

If you want to get the text of the option

```
$("#select").change(function(e){
 console.log("You have changed to " +
  $(this).children(":selected").text());
})
```

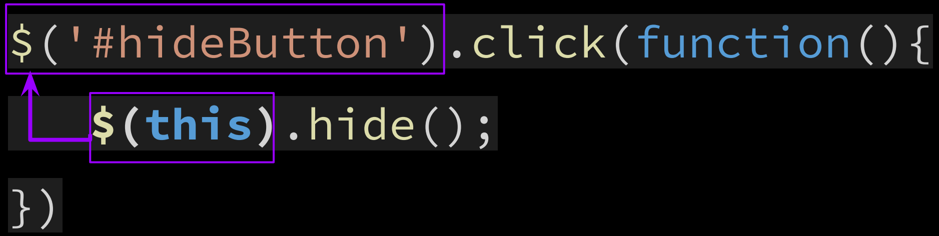
What is $(this)

You can access the html element that is clicked by using $(this)

Without using *$(this)*

```
$('#hideButton').click(function(){
    $('#hideButton').hide();
})
```

With using *$(this)*



```
$('#hideButton').click(function(){
    $(this).hide();
})
```

Further Reading: https://www.quora.com/What-is-the-difference-between-click-and-on-click-in-jQuery
https://coding.abel.nu/2013/04/this-and-this-in-jquery-callbacks/

Effects

EFFECTS

Used to enhance page with transitions and movements

Basic Effects

METHOD	DESCRIPTION
.show()	Displays selected elements
.hide()	Hides selected elements
.toggle()	Toggles between showing and hiding selected elements

Fading Effects

METHOD	DESCRIPTION
.fadeIn()	Fades in selected elements making them opaque
.fadeOut()	Fades out selected elements making them transparent
.fadeTo()	Changes opacity of selected element
.fadeToggle()	Hides or shows selected elements by changing their opacity (opposite of current state)

Further Reference Reading:
https://api.jquery.com/category/effects/
https://api.jquery.com/animate/

EFFECTS

Used to enhance page with transitions and movements

Sliding Effects

METHOD	DESCRIPTION
.slideUp()	Hides selected elements with a sliding motion
.slideDown()	Shows selected elements with a sliding motion
.slideToggle()	Hides or show selected elements with a sliding motion (opposite direction of current state)

Custom Effects

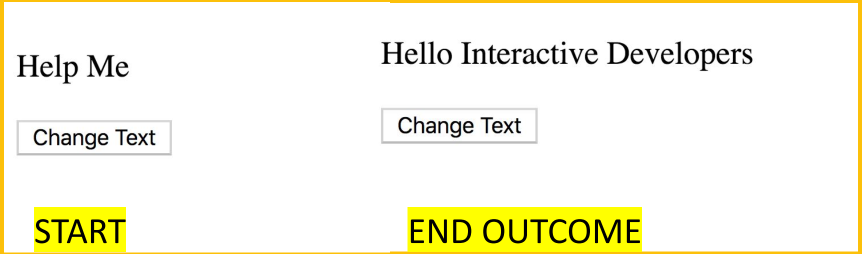
METHOD	DESCRIPTION
.delay()	Delays execution of subsequent items in queue
.stop()	Stops an animation if it is currently running
.animate()	Creates custom animation

Further Reference Reading:
https://api.jquery.com/category/effects/
https://api.jquery.com/animate/

ACTIVITY

“Simple JS Click” (15min)

#TODO CA2
Playing with inputs and jQuery selectors
Create a button that allows text to be changed from "Help Me"
To "Hello Interactive Developers"



Help Me

START

Hello Interactive Developers

END OUTCOME

https://repl.it/@malcolmyam/wk08-ca-simpleclick#script.js

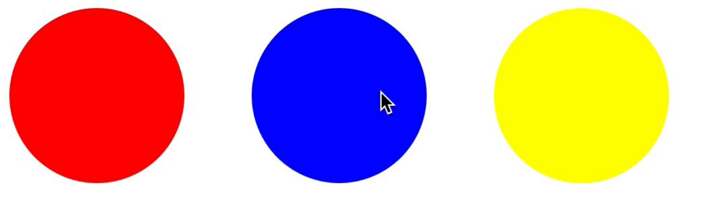
SUBMISSION
Create folder **wk08-simple-click** and place your codes for this exercise

Compile all the activities for the week and submit as a ZIP file
wk08-studentid-studentname.zip file
Name each activity properly
Eg.
/wk08-studentid-studentname
../wk08-template-10
../wk08-simple-click
../wk08-circles
../wk08-randomuser
../wk08-weather
../wk08-about-me

ACTIVITY

“Disappearing Act” (15min)

#TODO CA 3
Fork or download the repl.it
Using jQuery, upon **mouse click** of a circle, hide the circle



https://repl.it/@malcolmyam/wk08-ca-circles#script.js

SUBMISSION
Create folder **wk08-circles** and place your codes for this exercise

Compile all the activities for the week and submit as a ZIP file
wk08-studentid-studentname.zip file
Name each activity properly
Eg.
/wk08-studentid-studentname
../wk08-template-10
../wk08-simple-click
../wk08-circles
../wk08-randomuser
../wk08-weather
../wk08-about-me

jQuery

Summary

What have we covered?

jQuery Selectors

jQuery Events

jQuery

Key Takeaway?

Practise & Practise

jQuery is a library that helps us to simplify JS calls. Underlying jQuery is still (vanilla) JavaScript. jQuery just makes things easier. We still need to know what works behind the scenes in order to do better web development.