CS 22A JavaScript for Programmers

Today (Day04)

- Announcements
- Comparison Operators
- DOM & jQuery
- Lab: Basic jQuery

CodeCademy Courses

- CodeCademy has many new courses that we are **not** going to use this quarter
- Make sure you are working on the correct course by using the links inside the Canvas assignments!

CodeCademy Badges

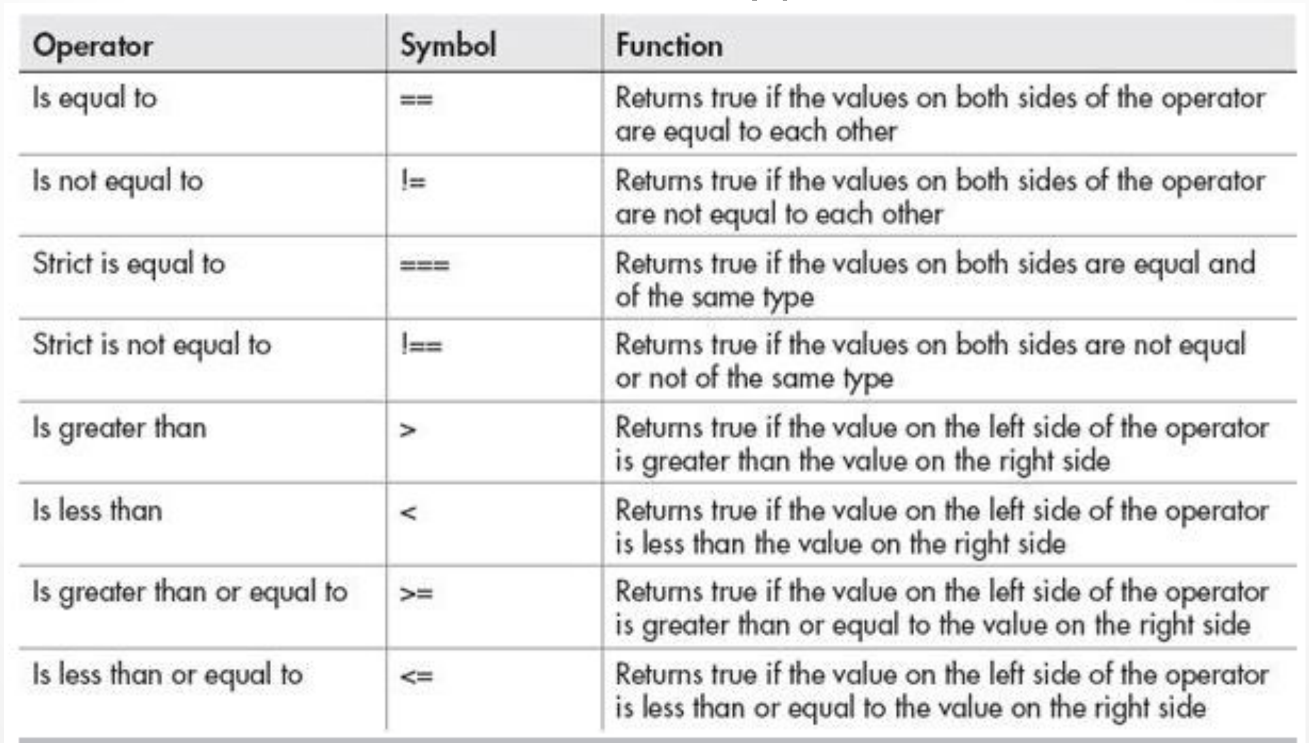
- The HTML & CSS CodeCademy badges from the first week are **optional**
- If you earned a CodeCademy badge but it is not showing yet, show me with screenshots or video
  - It can take a couple of days, so I generally grade badges over the weekend
- It's OK to complain about CodeCademy, **especially if the instructions are misleading or incorrect**. I might change deadlines for CodeCademy assignments like that. You can also complain anonymously!
  - Anonymous contact form is linked from the Syllabus

Guidelines

- Use **camelCase** for variables & functions/methods
- Open **curly braces** on the same line
- **Always** use blocks for if-statements, for-statements, loops, etc.,: even if it contains only 1 line
- Define your functions **before** you call them
- **Always** use semicolons to end a statement
- **Always** use **var** when declaring a variable

Comparison Operators

- Why use === and !== ? No type coercion!

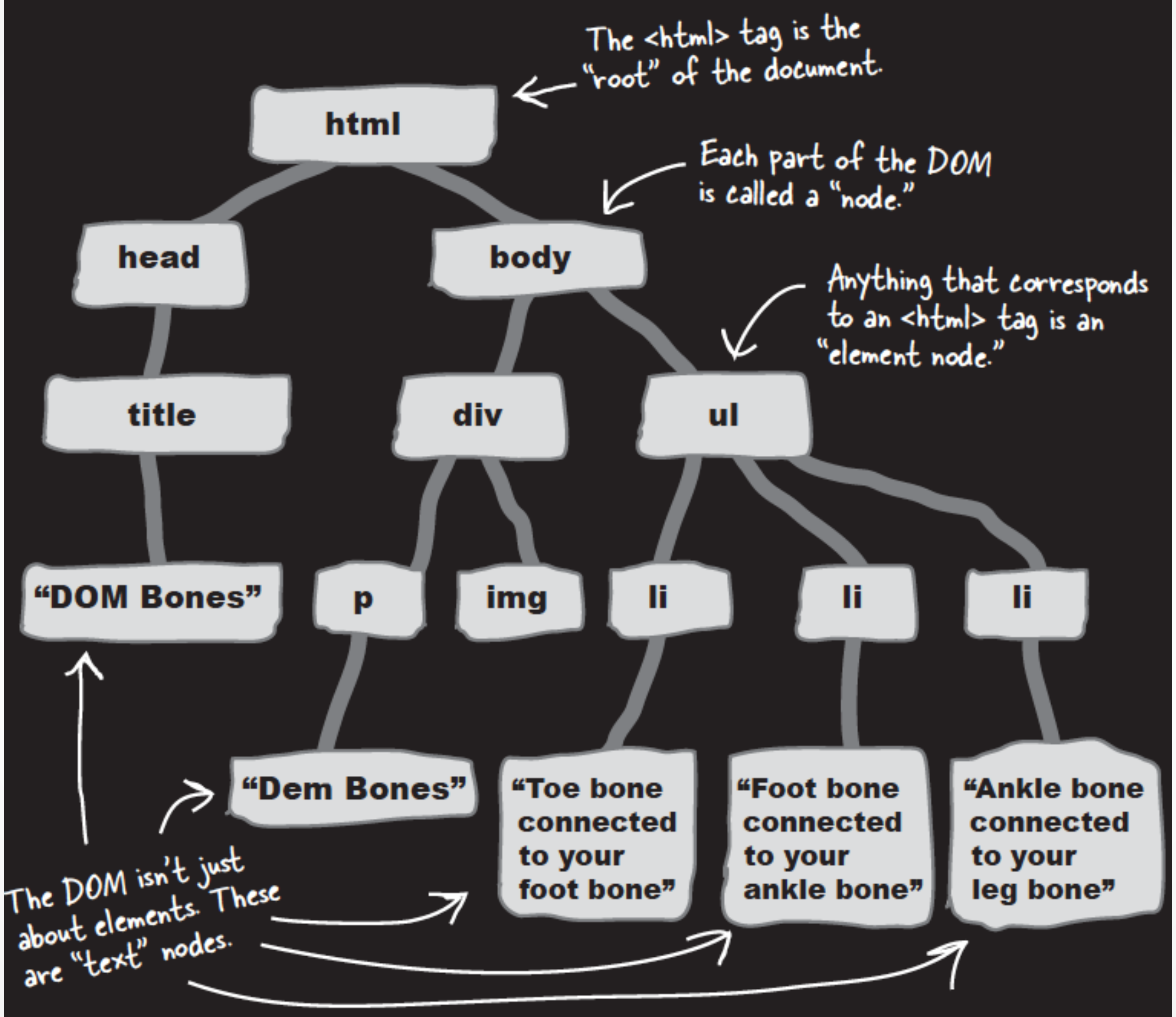


Operator	Symbol	Function
Is equal to	==	Returns true if the values on both sides of the operator are equal to each other
Is not equal to	!=	Returns true if the values on both sides of the operator are not equal to each other
Strict is equal to	===	Returns true if the values on both sides are equal and of the same type
Strict is not equal to	!==	Returns true if the values on both sides are not equal or not of the same type
Is greater than	>	Returns true if the value on the left side of the operator is greater than the value on the right side
Is less than	<	Returns true if the value on the left side of the operator is less than the value on the right side
Is greater than or equal to	>=	Returns true if the value on the left side of the operator is greater than or equal to the value on the right side
Is less than or equal to	<=	Returns true if the value on the left side of the operator is less than or equal to the value on the right side

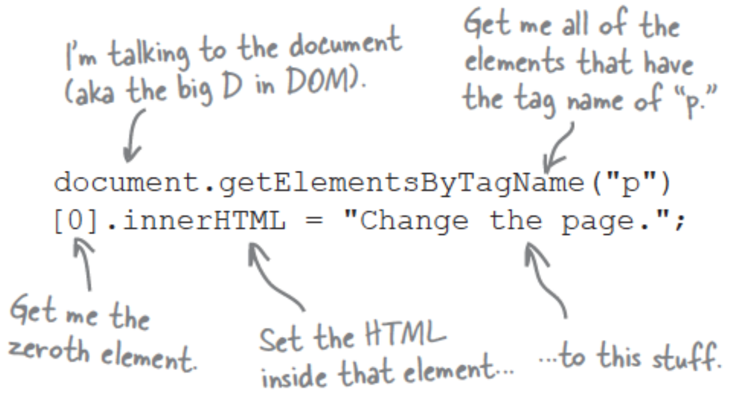
Document Object Model

- Why use the jQuery library?
  - It makes accessing and changing the HTML elements in the `document` object (the Web page's contents) easy!

The DOM (Document Object Model)


The <html> tag is the "root" of the document.
html
Each part of the DOM is called a "node."
head
body
Anything that corresponds to an <html> tag is an "element node."
title
div
ul
"DOM Bones"
p
img
li
li
li
"Dem Bones"
"Toe bone connected to your foot bone"
"Foot bone connected to your ankle bone"
"Ankle bone connected to your leg bone"
The DOM isn't just about elements. These are "text" nodes.

The raw JavaScript way


```
document.getElementsByTagName("p")
[0].innerHTML = "Change the page.";
```
I'm talking to the document (aka the big D in DOM).
Get me all of the elements that have the tag name of "p."
Get me the zeroth element.
Set the HTML inside that element...
...to this stuff.

The raw JavaScript way

I'm talking to the document (aka the big D in DOM).

Get me all of the elements that have the tag name of "p."

```
document.getElementsByTagName("p")
[0].innerHTML = "Change the page.";
```

Get me the zeroth element.

Set the HTML inside that element...

...to this stuff.

Or let's say we want to change the HTML inside of ***five*** paragraph elements on our page:

Loop through the number of elements I want to change.

```
for (i = 0; i <= 4; i++)
{
   document.getElementsByTagName("p")
[i].innerHTML="Change the page";
}
```

Get me the element we're looping over.

The JavaScript Query (jQuery) function

- This is the jQuery function:

```
jQuery( )
```

- This is the shortcut for the jQuery function:

```
$( )
```

- If you pass a CSS selector to the jQuery function, it will return the **set** of DOM elements that match that selector. You can then add behavior to those DOM elements by calling jQuery methods.

The raw JavaScript way

I'm talking to the document (aka the big D in DOM).

Get me all of the elements that have the tag name of "p."

```
document.getElementsByTagName("p")
[0].innerHTML = "Change the page.";
```

Get me the zeroth element.

Set the HTML inside that element...

...to this stuff.

The jQuery way

Grab me a paragraph element.

Change the HTML of that element to what's in these parentheses.

```
$("p").html("Change the page.");
```

jQuery uses a "selector engine," which means you can get at stuff with selectors just like CSS does.

Or let's say we want to change the HTML inside of ***five*** paragraph elements on our page:

Loop through the number of elements I want to change.

```
for (i = 0; i <= 4; i++)
{
   document.getElementsByTagName("p")
[i].innerHTML="Change the page";
}
```

Get me the element we're looping over.

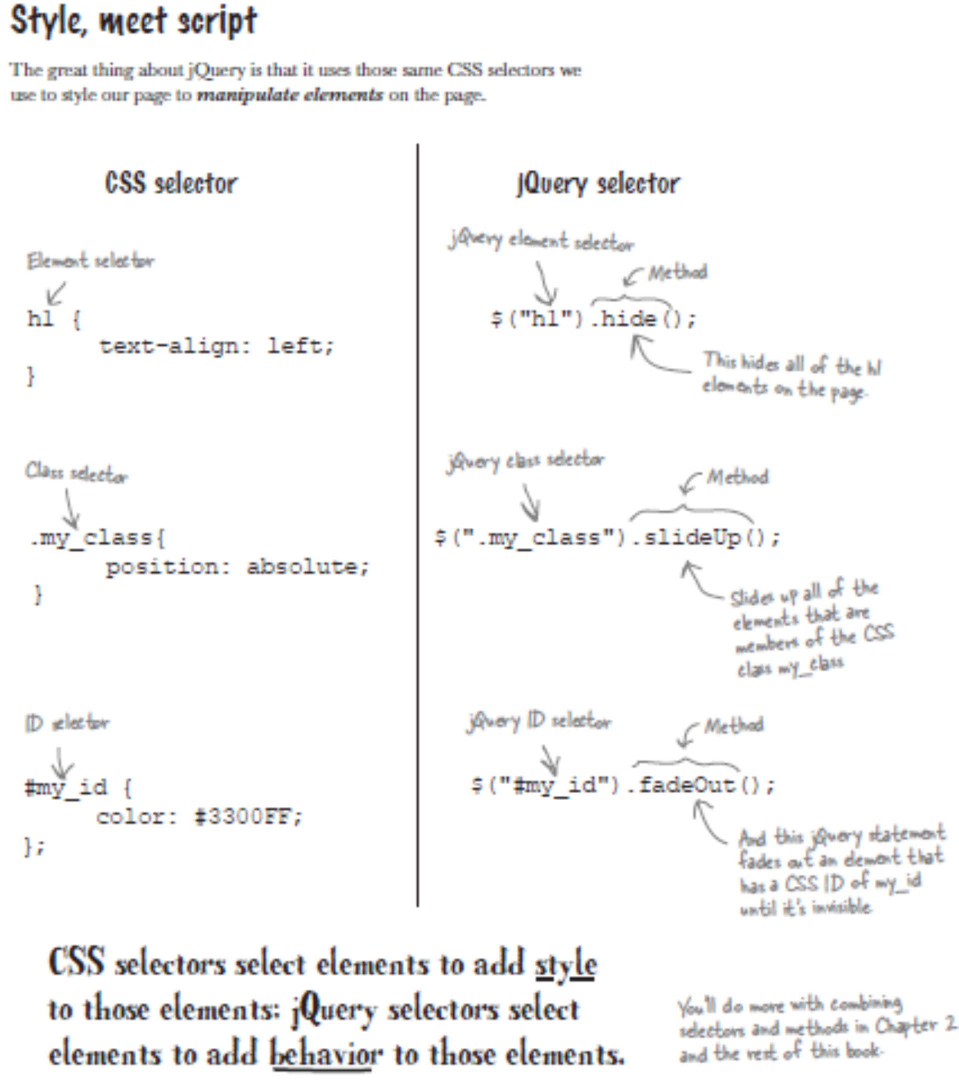
Because jQuery uses CSS selectors, we can say it the same way as above.

```
$("p").html("Change the page.");
```

```
<h1>Main Header</h1>

<p class="my_class">Some text.</p>
<p class="my_class">Some other
text.</p>

<p id="my_id">My unique element</p>
```



Style, meet script

The great thing about jQuery is that it uses those same CSS selectors we use to style our page to ***manipulate elements*** on the page.

CSS selector

Element selector

```
hl {
      text-align: left;
}
```

Class selector

```
.my_class{
      position: absolute;
}
```

ID selector

```
#my_id {
      color: #3300FF;
};
```

jQuery selector

jQuery element selector

Method

```
$("hl").hide();
```

This hides all of the hl elements on the page.

jQuery class selector

Method

```
$(".my_class").slideUp();
```

Slides up all of the elements that are members of the CSS class my_class

jQuery ID selector

Method

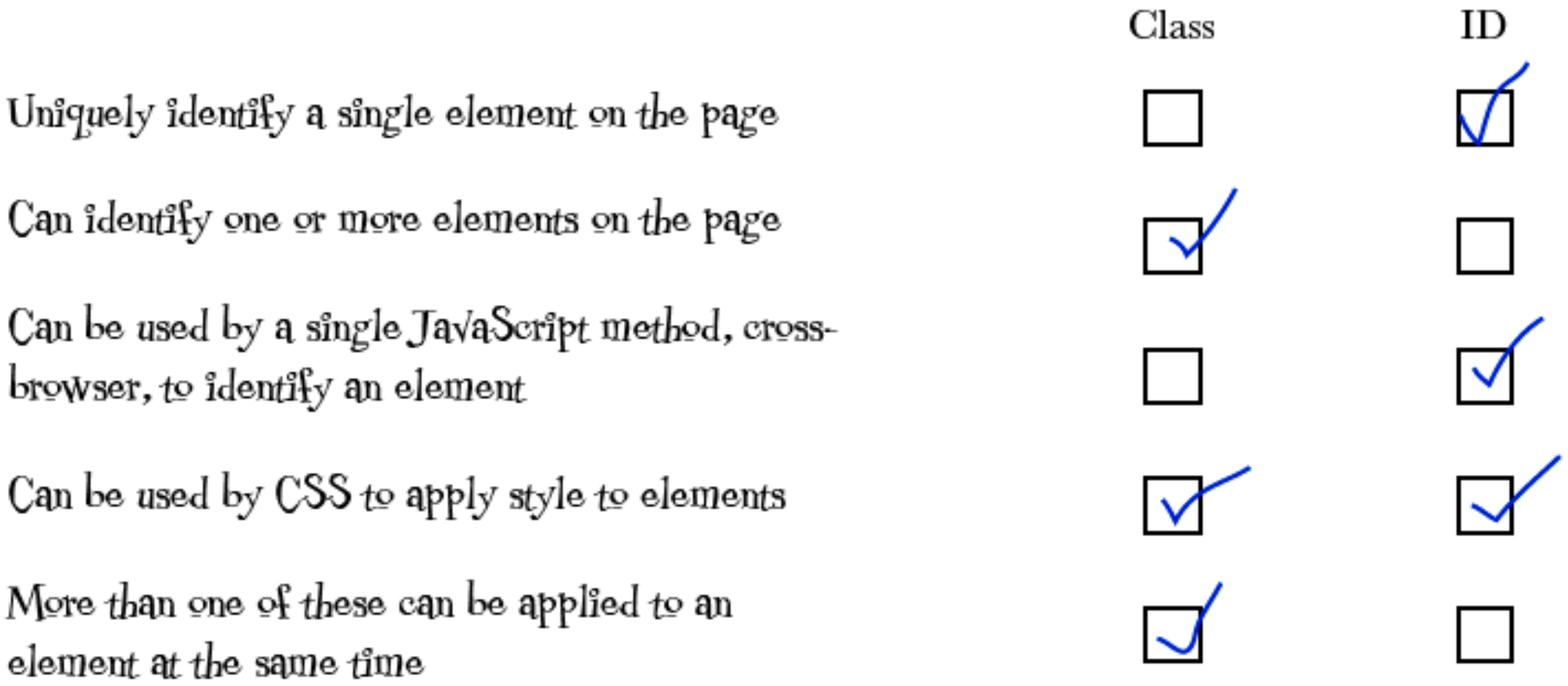
```
$("#my_id").fadeOut();
```

And this jQuery statement fades out an element that has a CSS ID of my_id until it's invisible

CSS selectors select elements to add <u>style</u> to those elements: jQuery selectors select elements to add <u>behavior</u> to those elements.

You'll do more with combining selectors and methods in Chapter 2 and the rest of this book.

CSS Class or ID?

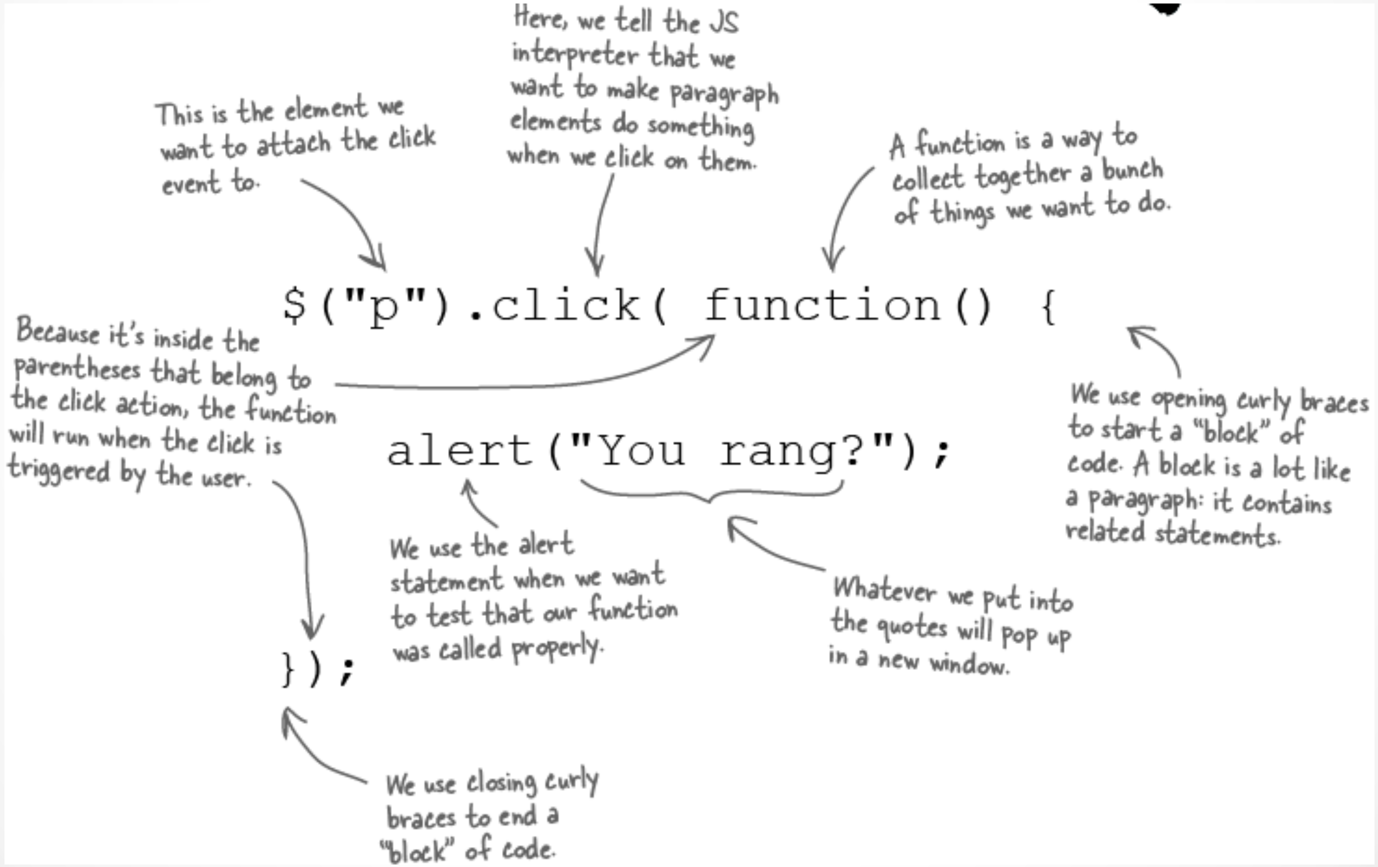


	Class	ID
Uniquely identify a single element on the page	☐	☑
Can identify one or more elements on the page	☑	☐
Can be used by a single JavaScript method, cross-browser, to identify an element	☐	☑
Can be used by CSS to apply style to elements	☑	☑
More than one of these can be applied to an element at the same time	☑	☐

Examples of Using jQuery

- http://learn.jquery.com/about-jquery/how-jquery-works

Passing a custom function to the click event



```
$("p").click( function() {

    alert("You rang?");

});
```

This is the element we want to attach the click event to.
Here, we tell the JS interpreter that we want to make paragraph elements do something when we click on them.
A function is a way to collect together a bunch of things we want to do.
Because it's inside the parentheses that belong to the click action, the function will run when the click is triggered by the user.
We use opening curly braces to start a "block" of code. A block is a lot like a paragraph: it contains related statements.
We use the alert statement when we want to test that our function was called properly.
Whatever we put into the quotes will pop up in a new window.
We use closing curly braces to end a "block" of code.

<div> or <span> tag?

- div is a block-level element
  - div “block” element visually isolates a section of a document on the page
- span is an inline element
  - span element contains a piece of information inline with the surrounding text

$(this)

- `this` is the current element that called the current method
  - In many object-oriented programming languages, `this` (or `self`) is a keyword that can be used in instance methods to refer to the object on which the currently executing method has been invoked.
- `$(this)` allows us to use jQuery methods to affect the current element
- The meaning of `$(this)` will change throughout your code, depending on where it is being referenced.

Selectors in jQuery

`$(this)`

Selects the current element.

`$(“div”)`

Selects all the div elements on the page.

`$(“div p”)`

Selects all the p elements that are directly inside div elements.

`$(“.my_class”)`

Selects all the elements with the my_class class.

`$(“div.my_class”)`

Selects only the divs that have the my_class class.
(Different types of elements can share a class.)

`$(“#my_id”)`

Selects the element that has the ID of my_id.

jQuery methods

method

- A method is a function that is a property of some object.
- You use methods to do actions in JavaScript.
- Think of a method as a verb—it's all about web page action.
- A jQuery method is reusable code defined in the jQuery library, usually as a property of the jQuery function: `jQuery( )` or its alias `$( )`

`$( ).append()`

- Inserts the specified content into the DOM. It gets added to the end of whatever element calls it.

`$( ).remove()`

- Takes elements out of the DOM.

jQuery Reference

- https://api.jquery.com/

To Do

- If behind, you should catch-up **before** the drop deadline on Sunday.
- Complete Week 2 Module