

The image features the jQuery logo, which consists of the word "jQuery" in a white, sans-serif font. The logo is positioned on the left side of a dark blue, angular shape that resembles a stylized arrow or a folded piece of paper. The background is a light blue gradient with geometric shapes, including a dark blue triangle and an orange horizontal bar at the bottom right.

jQuery

- **Selectors & their attributes** - returning info from the DOM
- **Events** - the new and improved way of adding event listeners
- **Traversing the DOM** - methods to navigate your dom tree and also to filter through those results.
- **Ajax is the new XHR** - a new way of writing your XHR calls. same core functionality, less confusing syntax

# 0.5

## What? Why?

jQuery is meant for DOM manipulation?  
Didn't we already learn that?

# What is jQuery

- A JavaScript library (some people wrote some JavaScript that makes it easier for other developers to write JS)
- Primary goals:
  - DOM manipulation
  - DOM Events
  - AJAX / Data retrieval
  - Effects & animation
  - Multi-browser support
- Yes, it is old school. But also:  
jQuery is used by 96.1% of all the websites whose JavaScript library we know. This is 73.4% of all websites.

Yeah, you're welcome.



# What is jQuery

Once we learn jQuery, we will basically be deprecating these from our life:

- `document.getElementById...`
- `.addEventListener`
- `element.innerHTML = ...`
- `.parentNode.parentNode.children[0]`
- `classList.contains`
- `XHR`

# 1

## Selectors & their attributes

Returning info from the DOM

## jQuery Basic Syntax

jQuery basic syntax **selects a dom element** and then performs **some action**

```
$( "selector" ).action()
```

jQuery **selects all <p>'s** and then **hides them all**

```
$( "p" ).hide()
```

## jQuery Selector Basics

jQuery uses CSS syntax to “find” and select HTML elements based on their name, id, classes, types, attributes, values of attributes, etc.

```
$("#main") // gets the element with ID of 'main'  
$(".fancy") // gets a list of elements with class of 'fancy'  
$("p") // gets a list of elements with 'p' tag  
$("div.fancy") // gets a list of all divs with class 'fancy'
```



# jQuery Selector Basics

Old Way:

```
const childrenOfMain =  
document.getElementById("main").getElementsByClassName("child")
```

New Way:

```
const childrenOfMain = $("#main .child")
```

## In Class Challenge: Find the funky bananas

```
<div>
  <h1 class='funky'> Monkey </h1>
  <div>
    <banana class='funky'> Bernerner </banana>
    <banana class='funky large'> B A N A N A </banana>
    <banana> Banana </banana>
  </div>
  <banana class='funky'> 100 </banana>
</div>
```

- Given the HTML above, what would the jQuery selector be to target all of the bananas with the class 'funky'.

## Getting Attribute

So, jQuery has a bunch of different methods that allow you to both GET and SET values of HTML elements

```
$( "#main" ).html(); // GETS the HTML content of the div main
```

```
$( "#main" ).html('<p>Sample</p>'); // SETS the HTML content of the div main
```

You can also use `.val()`, `.text()`, `.height()`, and various other methods to quickly get information from an HTML element or set it to new values.

# 2

## Events

The new and improved way of adding event listeners

## jQuery Basic Events

jQuery makes it drastically easier to listen for events on the DOM

```
$("#button").click((e) => {  
    console.log(e);  
});
```

```
const logEvent = (e) => {  
    console.log(e);  
}  
  
$("#button").click(logEvent);
```

## jQuery Common Events

Here are some of the methods that you will most commonly use to attach events with jQuery:

- Mouse events: `.click()`, `.hover()`, `.mouseleave()`
- Keyboard events: `.keyup()`, `.keypress()`
- Form events: `.change()`, `.select()`, `.submit()`
- Event handler attachment: `.on()`

Let's talk more about this `.on()` ...

## jQuery .click() vs .on()

.click() works perfectly well until it doesn't...

**It will not work for elements that are loaded into the DOM after jQuery has loaded. For example, anything that is printed dynamically with our printToDom function. For those cases, we need to use .on()**

```
$(".dynamic-button").on("click", (e) => {  
  console.log(e);  
});
```

## Attaching event listeners to many DOM elements

Old Way:

```
const children = document.getElementsByClassName("child");  
  
for (let i = 0; i < children.length; i++) {  
  children[i].addEventListener('click', myEvent);  
}
```

New Way:

```
$(".child").on('click', myEvent)
```



# 3

## Traversing the DOM

Methods to navigate your dom tree and also to filter through those results

## Common Methods

Going **up** the tree:

```
.parent() // returns the direct parent  
.closest('.fancy') // returns the closest parent with the class of fancy
```

Going **down** the tree:

```
.children() // returns the list of direct children  
.find('.fancy') // returns the closest descendant with the class of fancy
```

Going **sideways?** the tree:

```
.siblings() // returns the list of any siblings in the DOM
```

## Bonus Methods

You can use `'show'`, `'hide'`, and `'toggle'` to control the display of your html elements:

```
$('.crap .child').hide(); // hides all of the elements with class 'child' that  
are children of the parent with the class 'crap'
```

## Wrapping non-jQuery elements up to use jQuery methods

Sometimes you will have methods that return non-jQuery enabled DOM elements, for example: `event.target`. You will need to wrap them in the jQuery syntax in order to use jQuery methods on it.

```
$('.crazy button').on('click', (event) => {  
    $(event.target).hide()  
});
```

# In Class Coding: The Fish Store

FORK and CLONE this repository: <https://github.com/nss-evening-cohort-7/fish-store>



## BEFORE YOU CLONE, YOU NEED TO FORK

nss-evening-cohort-7 / fish-store

Unwatch 3 ★ Star 0 **Fork 0**

Code Issues 8 Pull requests 0 Projects 1 Wiki Insights Settings

A fish store for the jQuery.

[Add topics](#)

Edit

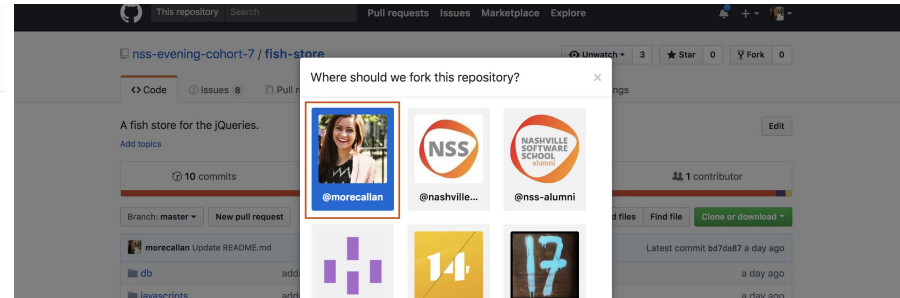
10 commits 2 branches 0 releases 1 contributor

Branch: master New pull request

Create new file Upload files Find file **Clone or download**

morecallan Update README.md

Latest commit bd7da87 a day ago



# 4

## AJAX is the new XHR

A new way of writing your XHR calls.

## The old way...

```
const loadItems = (loadFunction, errorFunction) => {  
  const itemLoader = new XMLHttpRequest();  
  itemLoader.addEventListener('load', loadFunction);  
  itemLoader.addEventListener('error', errorFunction);  
  itemLoader.open('GET', '../db/items.json');  
  itemLoader.send();  
};
```

## The new way...

```
const loadItems = (loadFunction, errorFunction) => {  
  $.get('/db/items.json')  
    .done(loadFunction)  
    .fail(errorFunction)  
};
```



## The old way...

```
const loadFunction = function() {  
  const data = JSON.parse(this.responseText);  
  // do something with data.items  
};
```

## The new way...

```
const loadFunction = (data) => {  
  // do something with data.items  
};
```