

# Week 11: jQuery

## Agenda

- Questions?
- Is jQuery Dying?
- Should I Use jQuery?
- The Toolbox
- Vanilla vs jQuery
- jQuery Advantages
- Additional Resources
- Next Week: Project

## What questions do you have?

## Is jQuery Dying?

Kind of.

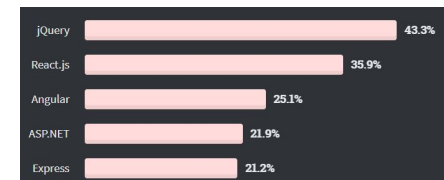
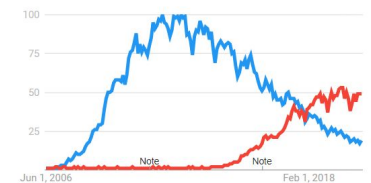
These days there is **very little that jQuery does that vanilla Javascript can't do** almost as easily.

But jQuery at one point was nearly synonymous with Javascript and **that popularity takes a long time to fade**.

There will likely be projects and companies still **using jQuery for a long time**.

But, **Bootstrap 5 no longer includes jQuery** which is probably the beginning of the end for jQuery.

Some new developers are not learning jQuery, it is starting to be considered a **legacy skill**.



# Should I Use jQuery?

If you are using React you should **NOT** use jQuery

## DOM Manipulation

Use It	Don't Use It
Simplifies DOM manipulation while being unopinionated	Vanilla is only slightly more complex and frameworks make low-level DOM manipulation unnecessary

## Animation

Use It	Don't Use It
Solid tools for moderately complex animation	CSS animation is faster and there are other libraries

## AJAX

Use It	Don't Use It
Streamlined AJAX with polyfilled browser support	You can use fetch() or a more robust AJAX library like Axios

# The Toolbox

## Finding

```
const $elem = $("query")
const $parent = $elem.parent()
```

## Reading & Updating

```
const oldValue = $elem.val()
$elem.val("new value")
```

## Creating

```
const $newLink = $("
```

## Moving

```
$parent.append($newNode)
$sibling.before($newNode)
```

## Deleting

```
$toRemove.remove()
$toEmpty.empty()
```

## Listening

```
$button.click(myFunction)
$button.on("click", myFunction)
```

# Vanilla vs jQuery

## Vanilla Javascript

```
const node = document.querySelector("query")
node.remove()
node.appendChild(anotherNode)
node.classList.add("new-class")
node.value = "new value"
node.textContent = "new text"
fetch("http://www.u.com").then(data => {
  console.log(data)
})
```

## jQuery

```
const $node = $("query")
$node.remove()
$node.append($anotherNode)
$node.addClass("new-class")
$node.val("new value")
$node.text("new text")
$.get("https://www.u.com", data => {
  console.log(data)
})
```

# jQuery Advantages

Cross-Browser Support	Nice, but cross-browser support is less and less of a problem these days Also that means it can be a little bloated with polyfills you don't need
Chainability	jQuery is often shorter and it's designed to be chainable: <code>\$(".class").parent().text("blah").addClass("something")</code> <code>document.querySelector(".class").parentNode.textContent = "blah"</code>
Popularity	If you load it in through a CDN, chances are the user won't even need to load it because another website will have already loaded it Using popular technologies makes it easier for you and others to maintain that code
Extra Tools	More options for how to find and manipulate the DOM. Things like: <code>.appendTo()</code> <code>.wrap()</code> <code>.empty()</code> <code>.closest()</code> <code>.filter()</code>

# Additional Resources

Free Code Camp jQuery vs Vanilla Javascript

<https://www.youtube.com/watch?v=X65s7WQLWJY>

Stack Overflow 2021 Developer Survey

<https://insights.stackoverflow.com/survey/2021>

State of Frontend

<https://tsh.io/frontend-development-trends-2020/>

jQuery API

<https://api.jquery.com/>

Can I Use

<https://caniuse.com/>