# DOM manipulation
# jQuery vs plain Javascript

Web Development Boot Camp

Lesson 5.2

# Remember:

**jQuery is written in Javascript.**

# jQuery is a Javascript library

## It was developed to make DOM manipulation easier

- Old Javascript had no way to query the DOM
- Javascript only had `document.getElementById()`
- A lot of tools for DOM manipulation (e.g., `attr, css, animate`)
- Affect multiple elements without a `for` loop (e.g., `$(“myclass”).on(“click”...)` )
- Code is more expressive, declarative
- Has been extended by the community (search for jQuery plugins)

# Drawbacks of jQuery

## There is no free lunch

- It is feature rich, which means a lot of code to download (bad for slow internet on mobile)
- jQuery’s simple syntax is built on a lot of Javascript (less efficient in some cases)
- Plain Javascript now has `querySelector` and `querySelectorAll`
- Has been overtaken by newer front end technologies (React, Angular, Vue)
- Doesn’t work on the server-side (i.e., node)

# jQuery in a Nutshell

We use the jQuery $( ) identifier to capture HTML elements:

| | |
|---|---|
| `$(“.classname”)` | `$(“div”)` |
| `$(“#idname”)` | `$(“p”)` |

Then, we tie the element to a jQuery method of our choice to capture events:

| | |
|---|---|
| `.on(“click”)` | `.ready()` |

Finally, we modify the selected element or add or remove elements from the DOM:

| | | |
|---|---|---|
| `.animate()` | `.append()` | `.remove()` |

# Compare jQuery to plain Javascript

## jQuery code is more expressive

| | Plain Javascript | jQuery |
|---|---|---|
| Get by id | `document.getElementById(“myId”)`<br>`document.querySelector(“#myId”)` | `$(“#myId”)` |
| Get element by CSS Selector | `document.querySelector(“.myClass”)`<br>Gets the first element that matches the selector | `$(“.myClass”)` |
| Get elements by CSS selector | `document.querySelectorAll(“.myClass”)` | `$(“.myClass”)` |

# Compare jQuery to plain Javascript

## jQuery code is more expressive

|  | Plain Javascript | jQuery |
|---|---|---|
| Change elements | `var items = document.querySelectorAll(“li.myClass”);`<br><br>`for(var i = 0; i < items.length; i++) {`<br>`   items[i].style.color = “#0000FF”;`<br>`}`<br><br>Or<br><br>`items.forEach(function(el){`<br>`  el.style.color = “#0000FF”`<br>`})` | `$(“li.myClass”).css(“color”, “#0000FF”);` |

# Compare jQuery to plain Javascript

## jQuery code is more expressive

| | Plain Javascript | jQuery |
|---|---|---|
| Add event handlers | // console.log a button’s data attribute<br><br>var buttons = document.querySelectorAll(“buttons”);<br><br>for(var i = 0; i < buttons.length; i++) {<br>   buttons[i].addEventListener(“click”,<br>     function(event){<br>       console.log(<br>         event.target.getAttribute(“data-command”));<br>       };<br>   );<br>} | $(“button”).on(“click”, function(){<br>  // this refers to event.target<br>  // but it is not a jQuery object<br>  console.log($(this).data(“command”));<br>}) |

# Compare jQuery to plain Javascript

## jQuery code is more expressive

| | Plain Javascript | jQuery |
|---|---|---|
| Add Element to DOM | `var newEl = document.createElement(“div”);`<br>`newEl.textContent = “Here is a new Element”;`<br>`document.body.appendChild(newEl);` | `var newEl = $(“<div>”);`<br>`newEl.text(“Here is a new Element”);`<br>`$(“body”).append(newEl);`<br>**OR**<br>`$(“body”).append($(“<div>”).text(“Hello”));` |

# Questions?