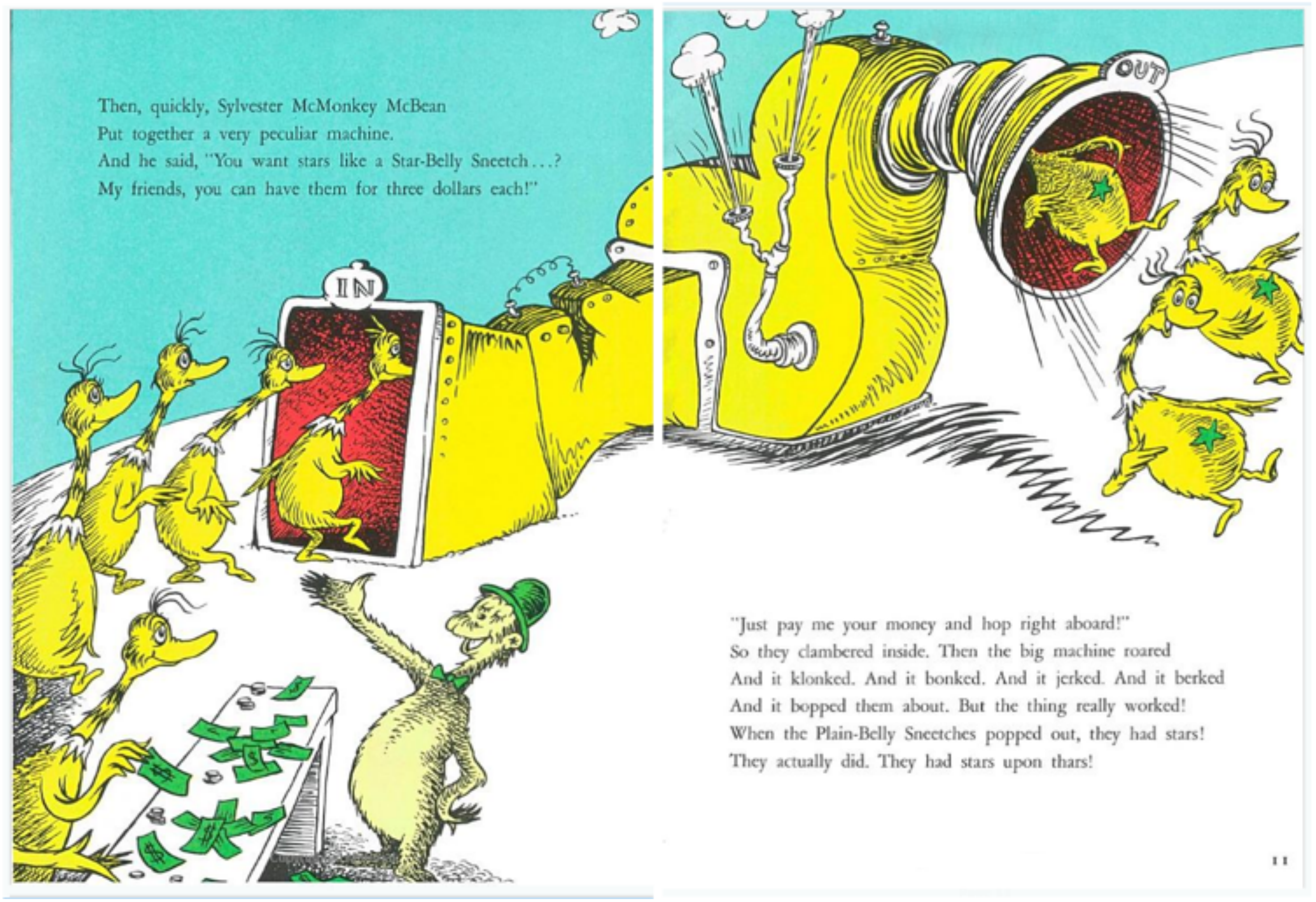
Lesson 9

jQuery

The jQuery function (we'll get back to this)



Then, quickly, Sylvester McMonkey McBean
Put together a very peculiar machine.
And he said, "You want stars like a Star-Belly Sneetch . . .?
My friends, you can have them for three dollars each!"

"Just pay me your money and hop right aboard!"
So they clambered inside. Then the big machine roared
And it klonked. And it bonked. And it jerked. And it berked
And it bopped them about. But the thing really worked!
When the Plain-Belly Sneetches popped out, they had stars!
They actually did. They had stars upon thars!

11

Learning Objectives

- Access and manipulate elements on a web page using jQuery
- Use event listeners to trigger events, using jQuery
- Use event delegation to manage dynamic content
- Understand where jQuery fits in the JavaScript ecosystem, past and present

Exit ticket question

Q: I feel like I'm understanding the grammar of JS but I don't have a good way to build my vocabulary. For example, other than **onclick**, I have no idea what other event triggers are out there. Are there good lists (that aren't overwhelmingly long)?

Exit ticket answer

- Short list of common events, later in these slides
- Long list of events can be found on MDN
- More generally, this is a really good question. We have to strike a balance between hitting major topics and going over all of the things you can do related to those topics. We don’t always get the balance right.
- It’s good to get used to searching in documentation for the answers to your questions. We will practice some of that today, together.

Homework

Taking a step back

Core JS

datatypes, control structures, standard library (Math.random function, etc.)

You get this for free when you run JS anywhere (browser, Node, etc.)

Taking a step back

DOM API

DOM objects like ‘document’, functions like ‘getElementById’

← You get this for free when you run JS in a web browser

Core JS

datatypes, control structures, standard library (Math.random function, etc.)

← You get this for free when you run JS anywhere (browser, Node, etc.)

Taking a step back

Third-party libraries
jQuery, React, Angular, etc.

← You have to get the user to download extra JS files — sometimes very large ones

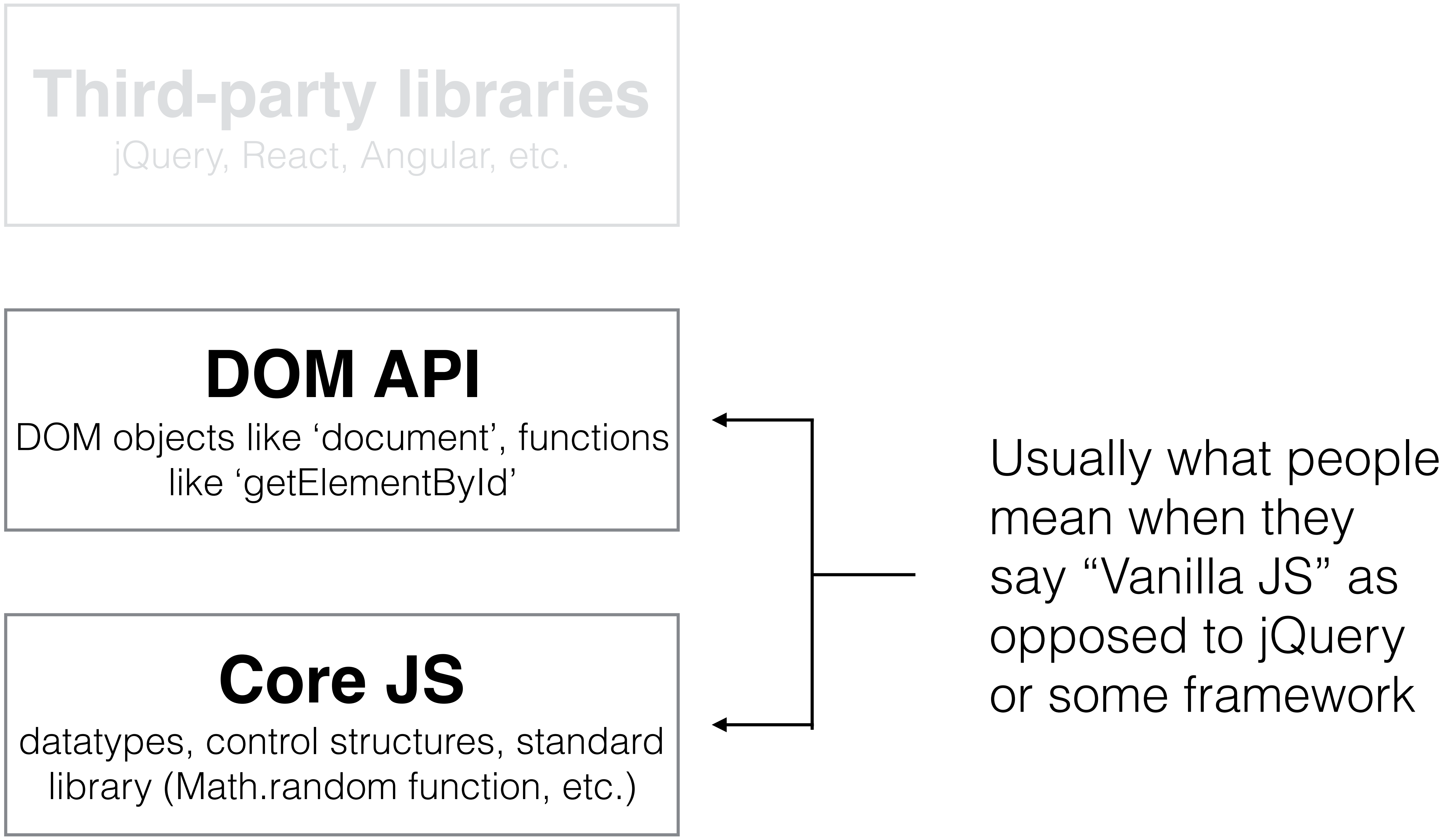
DOM API
DOM objects like ‘document’, functions like ‘getElementById’

← You get this for free when you run JS in a web browser

Core JS
datatypes, control structures, standard library (Math.random function, etc.)

← You get this for free when you run JS anywhere (browser, Node, etc.)

Taking a step back


Third-party libraries
jQuery, React, Angular, etc.
DOM API
DOM objects like ‘document’, functions like ‘getElementById’
Core JS
datatypes, control structures, standard library (Math.random function, etc.)
Usually what people mean when they say “Vanilla JS” as opposed to jQuery or some framework

jQuery: One library among many

- Released in 2006, was a godsend at the time
  - Smoothed over conflicts between different browsers' versions of JavaScript
  - provided convenience functions not available in the DOM API
- Browsers are much better now — other much smaller libraries are all you really need now, or even just vanilla JS

Downsides of jQuery

- It’s 10,000 lines of code — not awesome for bandwidth-constrained situations like mobile devices
- It’s a monolith that provides more than anyone really needs, and you have to download it all
- Cannot be used in Node.js at all
- Easy to become addicted to its way of doing DOM manipulation and its basic convenience functions, but it’s not always going to be available to you

Pluses to jQuery

- It still makes life pretty convenient for front-end programming (when you have access to it) so we’ll be leveraging it in order to accomplish more in this class

- It’s all over the place, and you will frequently encounter it in code bases. It’s not going anywhere

- Many, many “jQuery plugins” are available on the Internet

    - This just means scripts that you load through the jQuery object system

You Might Not Need jQuery

- Excellent website for keeping up to date on what modern browsers can do in Vanilla JS, that you used to have to use jQuery for:

  http://youmightnotneedjquery.com/

Final Word on jQuery vs. vanilla JS

- It’s good for a front-end developer to be able to do DOM manipulation both ways to some extent — with and without jQuery.

- For every exercise that uses jQuery, for those who are interested we will be posting two solutions: one with jQuery and one without, in case we didn’t go over the vanilla JS version in class.

- Don’t worry too much about knowing how every single DOM API method works, or every single jQuery method.

- Get comfortable looking up answers to your questions in the JS docs on MDN, and in the jQuery docs

So… about jQuery

- jQuery’s primary function is called **`jQuery`**.
- It does different things, depending on what’s passed to it.

jQuery object

- Main use of the jQuery function is to select a group of elements
- It’s a lot like **`querySelectorAll`**, except it returns something called a “jQuery object”, which is an array-like collection of elements
- This gets a jQuery object containing all the **`li`** elements inside the **`.specialList`** element:
    - **`jQuery(".specialList li")`**

    -
      ```
      <ul class="specialList”">
        <li>First Child</li>
        <li>Second Child</li>
      </ul>
      ```

jQuery object

- **`jQuery(".specialList li")`**
- You can do all kinds of things with the resulting “jQuery object” — add content, change styling, add event listeners
- Especially, you can perform operations *all at once on all the elements in the collection.*
- And it works like an array also — each element is a regular DOM API element. This gets the first element in the list:
    - **`jQuery(".specialList li”)[0]`**

Alias for jQuery object

- Since you type it so much, you can use a dollar sign as well, which is a legal variable name in JavaScript:

  ```
  jQuery(".specialList li")
  ```

  - is the same as

  ```
  $(".specialList li")
  ```

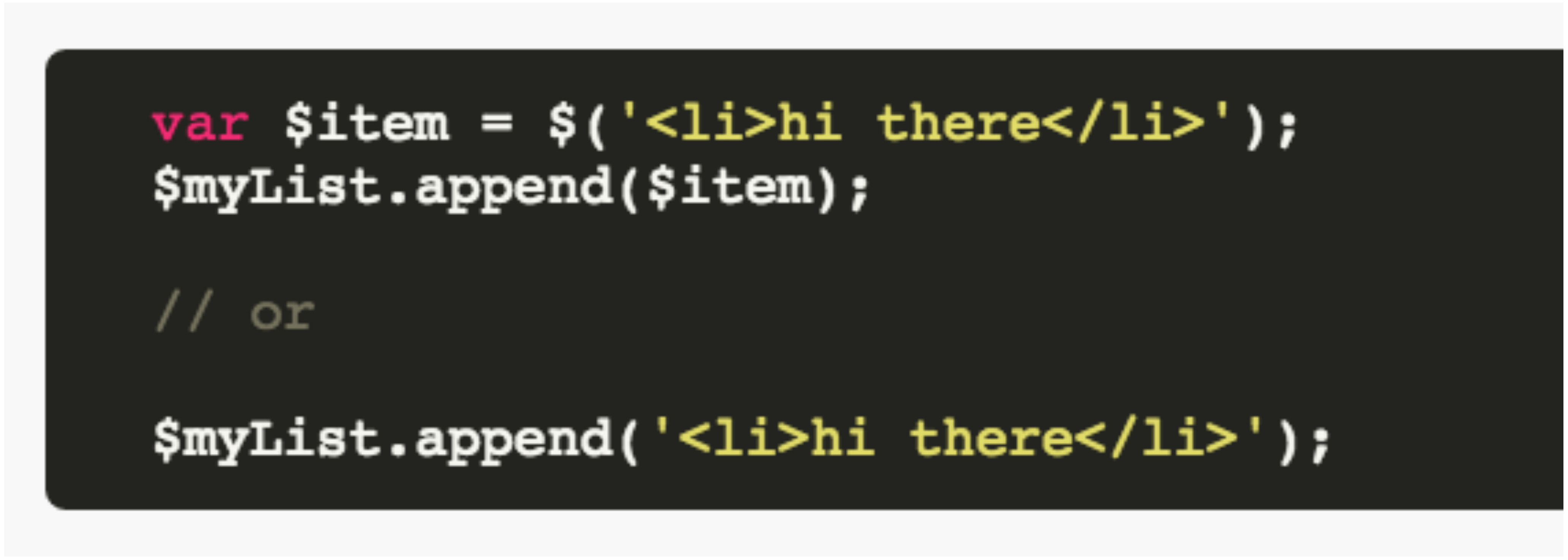
Target items:

```
// Target item by id
$('#item');

// Target item(s) by class
$('.box')

// Target item(s) by tag
$('h2')
```

Create new DOM elements:



```
var $item = $('<li>hi there</li>');
$myList.append($item);

// or

$myList.append('<li>hi there</li>');
```

Set CSS properties

```
var $item = $('#item');
$item.css('color', 'red');
```

- Although normally you'd probably do this by setting classes:

```
var $item = $('#item');
$item.removeClass('ordinary-text')
$item.addClass('banner-headline');
```

Moar Documentation!

- We're going to refactor our previous exercise into jQuery
- Instead of showing slides with all the methods before the codealong, we're going to look at the jQuery documentation.
- We'll try to figure out what functions we need by coming up with the right search queries

How to get the most out of the jQuery docs

- When reading the jQuery documentation, be sure to scroll through the whole document to ensure you're looking at the correct method signature.

- Most jQuery methods change their behavior depending on the number of arguments they have when called.

- When you need to look something up, most popular libraries will have a website dedicated to documentation. For example: http://api.jquery.com/

Codealong

- Open the following zipped folder and place it in your lesson 9 folder. Then open the folder in Sublime Text (“`subl .`”) and follow along.
- http://bit.ly/jsdev2-lesson9-codealong-convert-to-jquery

For future reference: some good jQuery methods to look up

`find()`	`append()`	`attr()`
`hide()`	`prepend()`	`val()`
`show()`	`on()`	`each()`
`html()`	`off()`	`parent()`
`text()`	`css()`	`siblings()`

What the $ does

function signature	what the argument is	what this does	what it's like
$('.main-box li')	CSS selector	creates new jQuery object containing existing DOM elements matched by selector	**document .querySelectorAll()**
$('<li>hi there</li>')	HTML string	creates new jQuery object containing new DOM elements	**document .createElement()**
$(el)	DOM element	creates new jQuery object containing existing DOM element	no parallel in vanilla JS
$([el1, el2, el3])	array of DOM elements	creates new jQuery object containing existing DOM elements	no parallel in vanilla JS

Practice on your own

- In your Lesson 9 folder, create a folder called "jquery-intro"
- Make a jquery-intro.html file and a jquery-intro.js file. Put one **ul** tag into the html file.
- Load jQuery in a script tag in your html file.
- Using jQuery, add three **li** elements inside the **ul**. Put some text inside the **li** elements.

jQuery lab: TODO list

- http://bit.ly/jsdev2-lesson9-lab-jquery-intro