FEWD Week 6 • Class 11: DOM Traversal & Manipulation

Quick Review

- What is the basic syntax for an if else statement?
- What is the syntax used for a click event listener?
- What method can be used to remove all of the inline styles on one or more elements?
- How do you get the current font-family from an element?

What We'll Cover

- How selectors work in jQuery
- The meaning and use of $(this)
- Traversing the DOM with jQuery methods
- Adding and removing elements to/from the DOM

Primer

jQuery and Selectors

Every time you use `$( )`, you're asking the Javascript engine to read the DOM and find every element that matches the selector you supplied. **This takes a lot of work, especially in a large DOM!**

Filling up the Bucket

...matches selector? drop it in...

Emptying the Bucket

...reached the end of a code statement? empty it...

One Way to Write More Efficient Code...

Don't empty the bucket!

Imagine This

```
...
<ul>
  <li id="id-43297">
    <span class="fullname">Jane Ayers</span>
    <ul class="details">
      <li class="start-date">January, 12 2000</li>
      <li class="status">Full Time</li>
      <li class="department">Engineering</li>
    </ul>
  </li>
  /* 3000+ Employees */
  <li id="id-56425">
    ...
  </li>
</ul>
```

How NOT to Write Code

```
$('#id-43297').addClass('inactive');
$('#id-43297 .status').text('Resigned');
$('#id-43297 .details').append('<li class="cobra">No</li>');
```

💩 At each semicolon, we're emptying the bucket!
With 10 employees this took: 2.1999999880790710 milliseconds

Write This Instead

```
$('#id-43297')
  .addClass('inactive')
  .find('.details').append('<li class="cobra">No</li>')
  .find('.status').text('Resigned');
```

🤯 With 10 employees this took: **1.5999998431652784 milliseconds**

```
$(this)
```

Objectives:

- Understand what $(this) is and how we can use it

What is $(this)?

In Javascript, `this` inside a function refers to the thing the function is called on. In jQuery, we wrap it in the dollar sign parantheses: `$(this)` so that we can use the jQuery methods on it.

🚩**HEADS UP:** Notice that **this** doesn't get any quotes!

For example...

Here, `$(this)` refers back to the specific div that we clicked on!

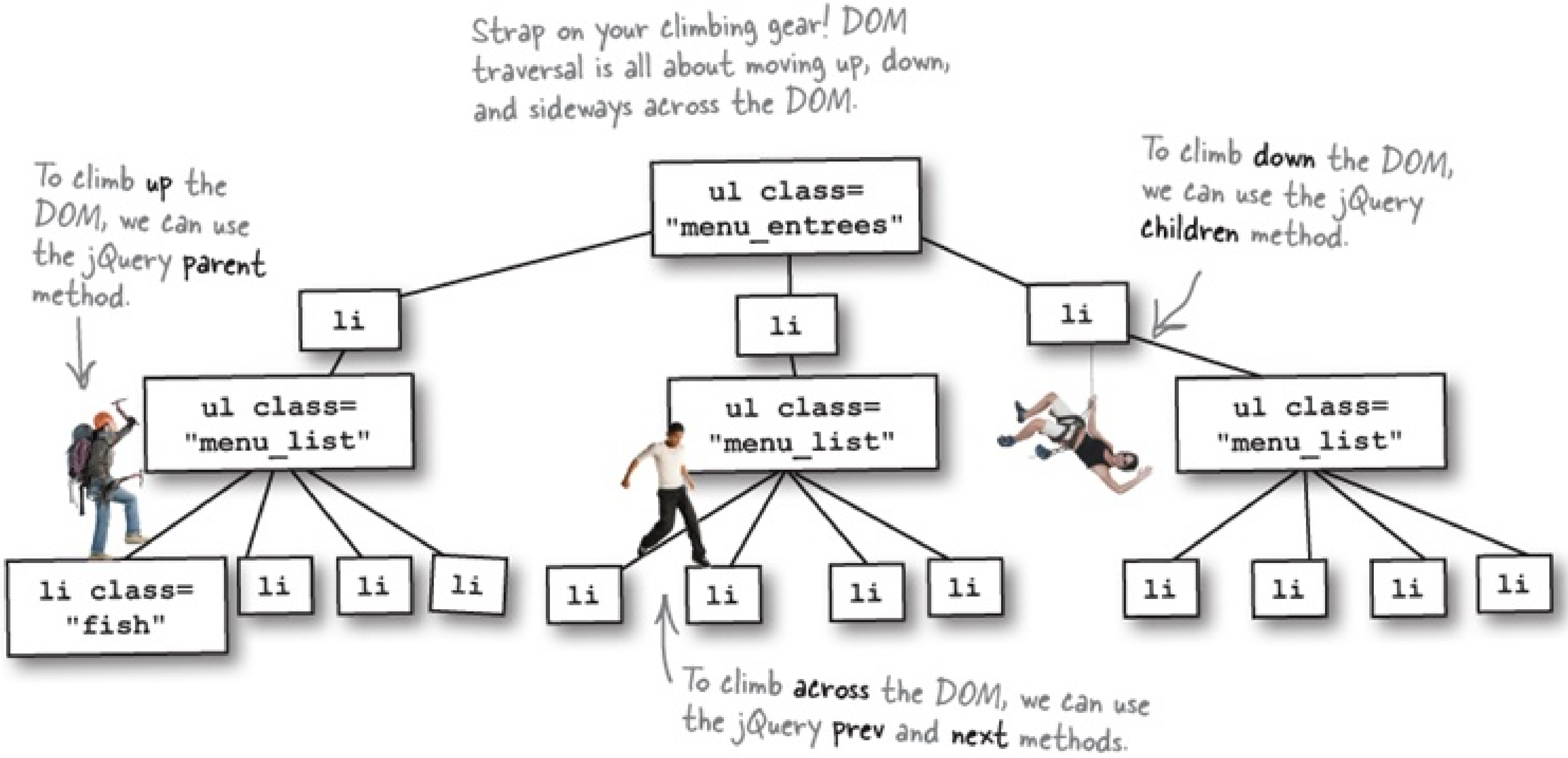
```
$('div').click(function(){

  $(this).css({'background-color': 'red'});

});
```

CODE ALONG

Card Flip

Traversing the DOM

jQuery Traversal Methods


Strap on your climbing gear! DOM traversal is all about moving up, down, and sideways across the DOM.
To climb up the DOM, we can use the jQuery parent method.
To climb down the DOM, we can use the jQuery children method.
To climb across the DOM, we can use the jQuery prev and next methods.
ul class= "menu_entrees"
li
li
li
ul class= "menu_list"
ul class= "menu_list"
ul class= "menu_list"
li class= "fish"
li
li
li
li
li
li
li
li
li
li
li

Add Some to Another Bucket

Up and Down the Tree

- .find()
- .children()
- .parent()
- .parents()

Find & Children

```
$('div').find('ul');
```

The .find() method gets the **descendants** of each element in the current set of matched elements, filtered by a selector.

```
$('div').children('ul');
```

The .children() method gets the **children** of each element in the current set of matched elements, *optionally* filtered by a selector.

Parent & Parents

```
$('div').parent('section');
```

The .parent() method gets the **parent** of each element in the current set of matched elements, *optionally* filtered by a selector.

```
$('div').parents('section');
```

The .parents() method gets the **ancestors** of each element in the current set of matched elements, *optionally* filtered by a selector.

Left and Right on the Tree

- .next()
- .prev()
- .prevAll()
- .nextAll()
- .prevUntil
- .nextUntil()

Next & Prev

```
$('div').next('div');
```

The .next() method gets the immediately **following** sibling of each element in the set of matched elements. If a selector is provided, it retrieves the next sibling only if it matches that selector.

```
$('div').prev('p');
```

The .prev() method gets the immediately **preceding** sibling of each element in the set of matched elements. If a selector is provided, it retrieves the next sibling only if it matches that selector.

NextAll & NextUntil

```
$('section').nextAll('div');
```

The .nextAll() method gets all **following** siblings of each element in the set of matched elements, optionally filtered by a selector.

```
$('section').nextUntil('footer');
```

The .nextUntil() method getsall following siblings of each element up to but **not including** the element matched by the selector, DOM node, or jQuery object passed.

End

The .end() method ends the most recent filtering operation in the current chain and return the set of matched elements to its previous state. **It takes the stuff in the second bucket and dumps it back into the first!**

CODE ALONG

Moving Around the DOM

Adding and Removing DOM Elements

Append & Prepend

```
$('section')
 .append('<div>My Content</div>');
```

The .append() method inserts content to the **end** of each element in the set of matched elements.

```
$('section')
 .prepend('<div>My Content</div>')
```

The .prepend() method inserts content to the **beginning** of each element in the set of matched elements.

AppendTo & PrependTo

The difference is what is the thing that we have in our bucket after the append or prepend operation!

```
$('section')
 .prepend('<div>My Content</div>');
```

section elements are in the bucket

```
$('<div>My Content</div>')
 .prependTo('section');
```

new div elements are in the bucket

CODE ALONG

Movie List

Midterm Checkup

- Schedule office hours check in with me.
- Make sure your assignments are submitted. I will have comments back to everyone by next week.
- Final project proposals were due, if you didn't get to complete them, make sure you do so immediately.
- Draft HTML is due week 7, so start now!

Go Build Awesome Things!