JavaScript: DOM Manipulation

What is DOM manipulation?

- Remember that DOM stands for **Document Object Model**
- The DOM is a representation of the HTML structure on the page that JavaScripts can interact with
- DOM manipulation simply means changing an element's text or inner html, or perhaps replacing it entirely

The HTML `<form>` element

- The login, signup, and address forms you see online all share a common tag: `<form>`!
- Inside of `<form>` are several elements that make up forms: text input boxes, dropdowns, radio buttons, checkboxes, etc.
- Today, we'll just be using the text and password input elements, but in future classes you'll learn about all of them!

<form> example

```
<form action="/process" method="POST">
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <label for="password">Password</label>
  <input type="password" name="password">
  <input type="submit" value="Submit">
</form>
```

Don't worry about action and method for now, we'll go over this when we learn Sinatra in a few weeks- also don't worry about submitting your form just yet

Retrieve input from a form element

You can see what's inside of a form element fairly easily, using the jQuery `val()` function:

```
<!-- Sample form input element -->
<input type="text" name="username" id="username">

// Sample jQuery, should be inside
// $(document).ready() normally

$("#username").val();

// returns the value of the field
```

Get the title of the form

- Imagine a `<form>` with an `<h1>` tag above it that has the form title
- We can use the jQuery `text()` function to retrieve the title inside the `<h1>` tag, or change it

```
<h1>Enter your information</h1>

$("h1").text()
>>"Enter your information"
var name = "Zach"
$("h1").text("Enter " + name + "'s Information")
// Changes the text in the DOM
// inside of <h1> to say this instead
```

Change the content of a `<div>` entirely

- Let's say that our `<h1>` lives inside of a `<div>`
- Using the `html()` jQuery function, we can change the innerHTML of the `<div>` entirely (see next slide)

Before:

```
html
<div class="main-section">
  <h1>Hello World</h1>
</div>
```

JS:

```
js
$(".main-section").html("<h3>Hello World Smaller</h3>")
```

After:

```
html
<div class="main-section">
  <h3>Hello World Smaller</h3>
</div>
```

Dealing with multiple elements on the page

- Sometimes you'll have more than one of each element, which can make selection difficult
- To overcome this, use the `first()` and `last()` functions

```
<h1>First Header</h1><h1>Second Header</h1>
```

```
// The below selects the first header
$("h1").first()
```

Dealing with multiple elements on the page

- If you're not selecting the first or last element, you'll need to choose an item in the array of items returned
- Unfortunately, accessing the array of items returned gives back a plain HTML element
- We'll need to re-wrap this element in jQuery to use it (see next slide)

```html
<h1>First Header</h1><h1>Second Header</h1><h1>Third Header</h1>
```

```javascript
var secondHeader = $("h1")[1];
$(secondHeader).text("New Text");

// Or simplified...
$($("h1")[1])
```

Exercise

- Create a page with multiple elements, some nested inside of each other, or use a page you've already built
- Practice selecting items and changing their contents from the JavaScript Console
- Once you have the hang of it, write a script to change the content in at least 5 elements

Manipulating the DOM when events are fired

Binding actions to the `change` event

- Remember that JavaScript can fire events on elements, which can be intercepted using an event handler
- An event handler is an anonymous function that is called when the event is fired

```
$("#password").change(funnction(){
  if($("#password").val() == ""){
    alert('password cannot be blank');
  }
});
```

Binding actions to the `submit` event

You can also make JavaScript "fire" when the form is submitted

```
$("form").submit(function(){
  if($("#password").val() == ""){
     alert('password cannot be blank');
    return false;
  } else if($("#username").val() == ""){
     alert('username cannot be blank');
    return false;
  } else {
  alert('all clear');
  }
});
```

Exercise

- Create a form that, when the password field is changed, checks the value of the field against a variable set earlier in the script, `correctPassword`
- If the password is correct, tell the user by changing an element on the page and vice versa for if the password is incorrect
- Bonus: See if you can create an anonymous function that stops the form from submitting when the "Submit" button is clicked