

# 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 `.value` attribute:

```
<!-- Sample form input element -->
```

```
<input type="text" name="username" id="username">
```

```
document.getElementById( 'username' ).value
```

```
// 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 attribute `.innerText` to retrieve the title inside the `<h1>` tag, or change it

```
<h1 id="title">Enter your information</h1>
```

```
heading = document.getElementById( 'title' )
```

```
heading.innerText
```

```
>> "Enter your information"
```

```
// set name to Zach
```

```
var name = "Zach"
```

```
// Changes the text in the DOM
```

```
heading.innerText = "Enter " + name + "'s Information"
```

```
// inside of <h1> to say this instead
```

```
heading.innerText
```

```
>> "Enter Zach's Information"
```

# Change the content of a `<div>`

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

# Change the content of a <div>

Before:

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

JS:

```
document.getElementById('main-section').innerHTML = "<h3>Hello World Smaller</h3>"
```

After:

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



# Dealing with Multiple Elements

- So far we have been selecting elements with unique id's
- What if we wanted to select by a class or tag?
  - `document.getElementsByClassName('class-name-here')`
  - `document.getElementsByTagName('tag-name-here')`

*Notice:* get**Elements**ByClassName vs. get**Element**ById

# Class & Tag Selectors

- `getElementById`: will return **ONE object** or null.
- `getElementsByClassName` or `getElementsByTagName`: will return an **ARRAY of objects**.

```
document.getElementById( 'title' )
```

```
>> <h1>Hello World!</h1>
```

```
document.getElementsByTagName( 'p' )
```

```
>> [<p>welcome to my site</p>, <p>buy my stuff</p>]
```

# Selecting From an Array

If you are using one of the multiple selectors, you can use [i] to select from the returned array.

```
paragraphs = document.getElementsByTagName( 'p' )  
>> [<p>welcome to my site</p>, <p>buy my stuff</p>]
```

```
paragraphs[0]  
>> <p>welcome to my site</p>
```

```
paragraphs[1].innerText  
>> 'buy my stuff'
```

# Selecting From an Array

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

```
var secondHeader = document.getElementsByClassName( 'h1' )[1];  
secondHeader.innerText = "New Text"
```

```
// the inner text of "secondHeader" is now set to "New Text"  
secondHeader.innerText  
>> 'New Text'
```

# Query Selectors

Some methods allow you to select an element or multiple elements by a selector or multiple selectors such as class, id or any other attribute name.

- `document.querySelector` - select one element that matches
- `document.querySelectorAll` - select all elements that match selector

# Query Selectors Example

```
<p class="dog-paragraph" id="paragraph-1">I love dogs</p>  
<p class="dog-paragraph" id="paragraph-2">I do not love dogs</p>
```

```
// will select the first paragraph with the class "dog-paragraph"  
document.querySelector(".dog-paragraph")
```

```
// will select all paragraphs with the class "dog-paragraph"  
document.querySelectorAll(".dog-paragraph")
```

```
// will select the first paragraph with the id "paragraph-1"  
document.querySelector("#paragraph-1")
```

```
// will select the only paragraph with the id "paragraph-2" within an array  
// like structure  
document.querySelectorAll("#paragraph-2")
```

# 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

# Resources

Mozilla

[Mozilla WebAPI - Document Object Model](#)

Css Tricks

[CSS Tricks - DOM Explanation](#)