**# Traversing the DOM**

**## Methods summary**

**### Traversing UP and DOWN - PARENT/CHILDREN**

$("#button2").parent();

$("#button2").children();

**### Traversing SIDEWAYS - NEXT/PREV**

$("div").next();

$("div").prev();

**### toggleClass() function - ADD/REMOVE**

$("div").toggleClass();

**### slideDown() function - SLIDE**

$("div").slideToggle();

**### first - returns the first element in .contentDiv**

$(".contentDiv").first().css("background-color", "red");

**### last**

$(".contentDiv").last().css("background-color", "red");

**### siblings - returns all siblings of a given element that share the same parent**

$("div").click() {

$(this).siblings().fadeTo("slow", .2);

}

**### filter - lets you specify some additional criteria**

$("p").filter(".intro").addClass("highlite");

**## TRAVERSING**

JQuery has a way of accessing elements based on the tree structure of the DOM.

This way of finding elements is called traversing.

**## Traversing UP and DOWN - PARENT/CHILDREN**

These methods only travers up or down one level!

**### parent() function - UP**

$("#button2").parent();

**### children() function - DOWN**

$("#button2").children();

If I want to return a specific child element:

$("#button2").children("p");

**## Traversing SIDEWAYS - NEXT/PREV**

**### next() function**

$("div").next();

The next() function traverses along the DOM and returns the next sibling element in the DOM after the selected element.

Siblings share the same parent, and they are side by side in the DOM tree.

1 $("div").next(); the next "div" in the DOM tree (sideways)

2 var firstDiv = $(".contentDiv").first() Selects all elements in the class ".contentDiv", then it requests the first element of "contentDiv".

3 firstDiv.next().css("background-color", "red"); Finally, it requests the NEXT element in the DOM tree after the first of the selected and sets a css property "background-color".

**### prev() function**

$("div").prev();

The prev() function traverses backwards along the DOM tree and returns the next sibling element in the DOM after the selected element.

The prev() function is the opposite of the next() function.

1 $("div").prev(); the next "div" in the DOM tree (sideways)

2 var firstDiv = $(".contentDiv").first() Selects all elements in the class ".contentDiv", then it requests the first element of "contentDiv".

3 firstDiv.next().css("background-color", "red"); Finally, it requests the NEXT element in the DOM tree after the first of the selected and sets a css property "background-color".

**### toggleClass() function - ADD/REMOVE**

$("div").toggleClass();

JQuery has a very useful class for toggling an element's class on and off, add and remove.

$("div").toggleClass("bigBorder"); This would add/remove the class .bigBorder when we click on the "div"

"this" .card is targeted by click! Set the .card and .highlite properties in the css file

$(".card").click(function() {

$(this).toggleClass("highlight");

});

**#### slideDown() function - SLIDE**

$("div").slideToggle();

$("img").click(function () {

$(this).next().children("p").slideDown();

});

1) Here we are targeting the image with the click function

2) We use (this) so its only "this image".

3) The next() function returns the sibling element to the image.

4) The children("p") function returns the specific "p" child to the parent "image"

5) The slideDown() function slides the "p" down when image is clicked!

**## All .js script for the challenges**

//wait until page is ready

$(document).ready(function() {

//sets <a> element within paragraph to yellow

$("p").click(function(){

$(this).children("a").css("background-color", "yellow"); /\* returns all the <a> child elements that are

within this paragraph\*/

});

// Toggle the visibility of the paragraph when a button is clicked

$("button").click(function(){

$(this).prev().slideToggle('slow');

});

// Open the paragraph once the image is clicked

$("img").click(function() {

$(this).next().children("p").slideDown();

});

$(".card").click(function() {

$(this).toggleClass("highlight");

});

// All cards that are not currently selected will be hidden when `select\_btn` is clicked

$("#select\_btn").click(function() {

$(".card:not(.highlight)").hide();

});

// Select all cards

$("#all\_btn").click(function(){

$(".card").show();

});

})