Traversing

Another super helpful jQuery concept is the ability to **traverse** elements in your page. What we mean by **traversing** is the ability to refer to elements by the other elements around them. There are three relationships to know when it comes to traversing: **.children(), .parent(),** and **.siblings()**. To illustrate this relationship, we will reference the following code:

```
<div id='myDiv'>
    <p>Here is a paragraph.  It is a beautiful paragraph.</p>
    <h3>Michael Choi</h3>
    <h3>John Supsupin</h3>
    <h3>Trey Villafane</h3>
</div>
```

If you weren't sold on properly indenting your code before, you will be now. If your code is clean and well-indented, then the parent/sibling/children relationship we are about to discuss will be quite apparent.

.children()

The jQuery method **.children()** is a way of targeting any items **contained within** the selected item. If within our jQuery code we typed:

```
$('#myDiv').children().css('background-color', 'green');
```

When we loaded this page, all of the elements contained within *myDiv* would have a green background. Notice that all the **.children()** selector does is return the items that satisfy the relation and allow us to write jQuery code for them. All of the traversing methods allow us to specify which relatives we would like to target by taking as an input a selector. For example:

```
$('#myDiv').children('h3').css('background-color', 'green');
```

Now, when we load our page, only the *h3* elements will have a green background. How nifty is that!

.parent()

Now that you have seen an example of traversing, you are probably off and running! The **.parent()** function works quite the same way as the **.children()** selector. Nevertheless, let's run some sample code:

```
$('h3, p').click(function(){
    $(this).parent().append('<h2>HOWDY CODING DOJO!</h2>');
})
```

As expected, this will add an *h2* tag inside the *div* because the parent element of all of the *h3* tags, as well as the lone *p* tag, is the *div.*

.siblings()

By now, you are an expert in traversing. So when you see this example...

```
  $('h3, p').click(function(){
      $(this).siblings().hide();
  })
```

you know exactly what is coming! Click an *h3* or a *p* and voila! Everything else is hidden.

Chaining

So now that you have some exposure to traversing, let's kick it up a notch! Check out this HTML:

```
<div>
   <button>Click me for crazy things!</button>
   <ul>
       <li><h3>Tiger Woods</h3></li>
       <li><h3>Phil Mickelson</h3></li>
       <li><h3>Ben Hogan</h3></li>
       <li><h3>Jack Nicklaus</h3></li>
       <li><h3>Gary Player</h3></li>
       <li><h3>Rory Mcilroy</h3></li>
   </ul>
</div>
```

Let's spice it up with some jQuery traversing:

```
$('button').click(function(){
    $(this).siblings().children().children().text('Michael Choi was here!')
})
```

Sweet, huh? Chaining these jQuery methods is a breeze!