

CSS Selectors and Prototype

Most JavaScript libraries provide some form of the `$()` function. This function is a short hand version of `document.getElementById()`. However, I was only recently introduced to the `$$()` function in Prototype. `$$()` takes a list of CSS selectors and returns an array of objects that match. This ends up being a much easier and more maintainable way to access elements in the DOM than giving EVERYTHING an id. Although the scope of CSS selectors in general is way to large for this tutorial, we will walk thru a simple use of the `$$()` function in prototype and introduce you to CSS selectors.

The Elements of a List

Let's look at the following HTML. We have created a simple ordered list with three elements.

```
<html>
  <body>
    <h1>To Do List</h1>
    <ol>
      <li>Finish this blog post</li>
      <li>Go car shopping with wife</li>
      <li>Cry about 401k</li>
    </ol>
  </body>
</html>
```

Now notice that nothing has an id. However, let's say I want to loop over my todo list using Prototype. Here's all the code I would need:

```
function alertItems() {
  $$('li').each(function(element) {
    alert(element.innerHTML);
  });
}
```

Let's take a look at that function. What Prototype is doing is getting all of the li elements in the page and returning them as an array. We then loop over the array, alerting the contents of each element. Now let's look at a little more complex example.

Multiple Lists

Let's say we have more than one list on the page (*gasp!) like the HTML below.

```
...
<h1>To Do List</h1>
<ol id="todo">
  <li>Finish this blog post</li>
  <li>Go car shopping with wife</li>
  <li>Cry about 401k</li>
</ol>

<h1>Completed List</h1>
<ol id="finished">
  <li>Dishes</li>
```

```

    <li>Laundry</li>
    <li>Email responses</li>
</ol>
...

```

Now what would happen if we reran the same function we created previously? It would give us six alerts, one for each li in both lists. But let's say we only want to loop thru the items in the second list. That's easy:

```

function alertItems() {
    $('#finished li').each(function(element) {
        alert(element.innerHTML);
    });
}

```

All we need to do is add "#finished" to the selector in our function. Now Prototype looks for the element that has an id that equals finished and gets all of the list items within that element. Keep in mind that the element that is identified by the id=finished could be any element. It doesn't need to be the list tag itself. It could be a div that had a list in it for example.

Selecting by Attribute

Selectors don't only work with elements. You can use them to select by attributes of elements as well. Let's say we wanted to identify all of the items on our todo list that are overdue with the class overdue:

```

...
<h1>To Do List</h1>
<ol id="todo">
    <li class="overdue">Finish this blog post</li>
    <li class="overdue">Go car shopping with wife</li>
    <li>Cry about 401k</li>
</ol>

<h1>Completed List</h1>
<ol id="finished">
    <li>Dishes</li>
    <li>Laundry</li>
    <li>Email responses</li>
</ol>
...

```

To iterate over the items that are overdue we could do it two ways. If we want all items that are overdue (regardless of the list), our function could look like this:

```

function alertItems() {
    $('li[class="overdue"]').each(function(element) {
        alert(element.innerHTML);
    });
}

```

However, if we wanted to be specific and only include the items that are overdue in our todo list, we would add the #todo selector so our function looks like this:

```
function alertItems() {
    $$('#todo li[class="overdue"]').each(function(element) {
        alert(element.innerHTML);
    });
}
```

Conclusion

CSS selectors are a powerful way to navigate the DOM and are a great tool to have in your belt, whether you are using them for CSS or JavaScript. You can read more about Selectors on the w3c site here: <http://www.w3.org/TR/2001/CR-css3-selectors-20011113/#selectors> and you can read more about Prototype's \$\$() function in the Prototype API documentation here: <http://www.prototypejs.org/api/utility/dollar-dollar>

For reference, the finished HTML document I created for this example is below.

```
<html>
  <head>
    <script
src="http://ajax.googleapis.com/ajax/libs/prototype/1.6.0.3/prototype.js">
    </script>
  </head>
  <body>
    <h1>To Do List</h1>
    <ol id="todo">
      <li class="overdue">Finish this blog post</li>
      <li class="overdue">Go car shopping with wife</li>
      <li>Cry about 401k</li>
    </ol>

    <h1>Completed List</h1>
    <ol id="finished">
      <li>Dishes</li>
      <li>Laundry</li>
      <li>Email responses</li>
    </ol>
    <script>
      function alertItems() {
        $$('#todo li[class="overdue"]').each(function(element) {
          alert(element.innerHTML);
        });
      }
      Event.observe(window, 'load', alertItems);
    </script>
  </body>
</html>
```