Using jQuery Selectors

In this section, you select all the elements of a single type. You also find out how to filter elements based on id values, CSS classes, and their order on the page.

Selecting all elements

You can select every element on a page by using the * character. For example, if I wanted to add an id attribute to every element in the preceding code, I could use this line of code:

$('*').attr({id: 'myNewID'});

You shouldn't use the * character with some jQuery functions. Consider this line of code:

$('*')text('Not a good idea.');

When this line of code executes, it first replaces the outermost element's text. Because the outermost parent element of any HTML page is the <html> element, you end up with this HTML code on your page:

<html>Not a good idea.</html> In general, it's practical to use the * selector when you want to assign a CSS style or an attribute to every element on a page or can narrow the results further in some other way.

Selecting an id

In Chapter 2, you see src, alt, height, and width attributes used for <img> elements. There are other attributes, such as the id attribute, that you can assign to all elements. The id attribute contains a unique identifier that you can use with a selector to pinpoint a specific element.

For example, here is code that assigns ids to two <p> elements:

<p id='someTxt'>Some text<p>

<p id='moreTxt'>More text<p>

The first <p> element has the someTxt id. The second has the moreTxt id. If I want to select the second <p> element, I can use the id name moreTxt in a selector. Several rules govern id attributes.

✓ **An id must be unique.** You can use an id only once per HTML page.

✓ **An id can contain only letters, numbers, hyphens, underscores, colons, and periods.**

✓ **An id must begin with a letter.**

✓ **An id is case-sensitive.** The id you use in the HTML tag must match the one you use in your selector.

✓ **An id is used in jQuery with a pound sign (#) in front of the id name.** In my HTML code, for example, I use id = 'myidname'. But when I use the id with a selector in the jQuery code, I place a pound sign in front of the id name, that is, '#myidname'.

To display an attribute from an element using an id selector, follow these steps:

1. Create a Web page containing the following code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>My Test Page</title>
<script type="text/javascript" src="js/jquery-1.4.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
alert ($('#moreTxt').text());
});
</script>
</head>
<body>
<p id='someTxt'>Some text<p>
<p id='moreTxt'>More text<p>
</body>
</html>
```

Selecting classes

Using ids to select specific elements gives you a lot of control over the elements on your page. But because ids are unique, you have to select each element by id. If you want to select four elements on your page, for example, you have to use all four ids in your code. For example, to set the src attribute for four <img> elements on a page, you would use the following code:
$('#anid').attr('src') = 'images/newImage.gif';
$('#anotherid').attr('src') = 'images/newImage.gif';

$('#myid').attr('src') = 'images/newImage.gif';
$('#hereisanid').attr('src') = 'images/newImage.gif';

To select elements by id, the code in the HTML contains id attributes for each element without a pound sign. Each of these lines sets the src attribute of a particular <img> element. If you use a special attribute known as a class, you can select all elements with that attribute with a single line of code, such as:
$('.someClass').attr('src') = 'images/newImage.gif';
Several rules govern class attributes.

✓ **A class may be used by more than one element.** You can use an id only once per HTML page.

✓ **An element may contain more than one class.** If you want to give an element multiple class attributes, use a space between class names. For example, here are three class attributes assigned to a single <p> element:
<p attribute="firstclass anotherclass dogclass catclass">

✓ **A class attribute is used in jQuery with a period (.) in front of the class name.** In HTML code, for example, I use class = "myclass". But when I use the class with a selector in the jQuery code, I place a period in front of the class name, that is, '.myclass'.
To change the text in <p> and elements with the same class, do the following:

1. Create a Web page containing the following code:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>My Test Page</title>
<script type="text/javascript" src="js/jquery-1.4.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
        $('.changemytext').text('This is new text.');
});
</script>
</head>
<body>
<strong class="changemytext">some name</strong>
<p class="changemytext">Some text<p>
<strong>another name</strong>
<p>More text<p>
<strong>another name</strong>
<p>Even more text<p>
<strong class="changemytext">your name</strong>
<p class="changemytext">Last bit of text<p>
</body>
</html>
```

Selecting by order

When you have several of the same type of elements on a page, you may want to select by order. For example, you might want to select the third <p> element on the page. jQuery has a set of functions that lets you select based on an element's position on the page. When using lists of things in the jQuery library, the first item in the list is always considered number 0, not number 1. Code examples in this section that choose an element based on its position in a list count from the beginning of the list starting with 0. Consider this code with several different elements, <p>, , and <img>. There are eleven total elements, four <p> tags, four tags, and two <img> tags:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
```

```
<head>
<title>My Test Page</title>
<script type="text/javascript" src="js/jquery-1.4.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
// Your code goes here.
});
</script>
</head>
<body>
<strong>Element, first strong tag</strong>
<p>Element, first p tag<p>
<strong>Element, second strong tag</strong>
<p>Element, second p tag<p>
<strong>Element, third strong tag</strong>
<p>Element, third p tag<p>
<img src="images/cover1.jpg" alt="Element"/>
<strong>Element, fourth strong tag</strong>
<img src="images/cover1.jpg" alt="Element" />
<p>Element, fourth p tag<p>
<img src="images/cover2.jpg" alt="Element" />
</body>
</html>
```

Try the code in the following examples by replacing the //Your code goes here line in the preceding code.

✓ **:first:** Selects the first matching element. This code returns the value of the src attribute of the first <img>, which is images/cover1.jpg.

```
$('img:first').attr('src');
```

✓ **:last:** Selects the last matching element. This code returns the value of the src attribute of the last <img>, which is images/cover2.jpg.

```
$('img:last').attr('src');
```

✓ **:even:** Matches even elements, starting with 0. This code changes the text of the first and third elements

```
$('strong:even').text('Changed this text.');
```

✓ **:odd:** Matches odd elements, starting with 1. This code changes the text of the second and fourth elements.

```
$('strong:odd').text('Changed this text.');
```

Lists of things used with jQuery begin numbering at 0, not 1. To select the first element, use an index of 0. The second element has an index of 1, the third an index of 2, and so on.

✓ **:eq(index):** Matches a specific element by counting from the first element to the index value. Suppose that you want to choose the third element on the page. Because the count starts with 0, to select the third element, do the following:

```
$('strong:eq(2)').text('Changed this text.');
```

✓ **:gt(index):** Selects all elements with an index value greater than the index. Selected elements are elements below the selected element on the page.

✓ **:lt(index):** Selects all elements with an index value less than the index. Selected elements are above the selected element on the page.

Selecting from Forms

jQuery has a special set of filters just for selecting elements in HTML forms. The examples in this section work with the following code to select form elements:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html>
<head>
<title>My Test Page</title>
<script type="text/javascript" src="js/jquery-1.4.min.js"></script>
<script type="text/javascript">
$(document).ready(function(){
// Your code goes here.
});
</script>
```

```
</head>
<body>
<form action="" method="post">
Your name<input type="text" />
<input type="checkbox" />
<input type = "radio" />
<select><option>First Choice</option></select>
<input type="submit" />
</form>
</body>
</html>
```

The following are some of the form element selectors and examples of their use on elements in the preceding code:

✓ **:input:** Selects all form elements including <input />, <select>, <textarea>, and <button>. This code shows the number of input elements in my form in an alert box:

```
alert( $(':input').length);
```

When a selector selects more than one element, the result is a list of values known as an *array.* When I select all the inputs on my page, I get back an array of all the elements. The length keyword tells me how many elements are in my array.

✓ **:text:** Selects all elements with the type attribute set to text. The following code returns the value 1 in an alert box:

```
alert( $(':text').length);
```

✓ **:radio:** Selects all elements with the type attribute set to radio. The following code returns the value 1 in an alert box:

```
alert( $(':radio').length);
```

✓ **:checkbox:** Selects all elements with type attribute set to radio. The following code sets the checked attribute to true for all check boxes

```
$(':checkbox').attr({checked:'true'});
```

✓ **:checked:** Selects all check boxes and radio buttons that are checked.

Selecting Attributes

Elements can be selected by using their attributes and attribute values. Here are some attribute filters:

✓ **[attribute]:** Selects all elements with a specific attribute. The following code displays the number of <img> elements with a height attribute:

```
alert( $('img[height]').length);
```

You can leave off the name of the element to select all the elements with a particular attribute. For example, the following code returns all elements with a height attribute, whether or not they are <img> elements:

```
alert( $('[height]').length);
```

✓ **[attribute=value]:** Selects all elements with a particular attribute set to a specific value. The following code displays the number of elements with a class attribute set to myclass:

```
alert( $('[class=myclass]').length);
```

✓ **[attribute!=value]:** Selects all elements with a particular attribute not set to a specific value. The following code displays the number of elements with a class attribute that isn't myclass. Elements with no class attribute are ignored:

```
alert( $('[class!=myclass]').length);
```

Selecting Visibility

Being able to hide and show elements are some of the fun things you can easily do with jQuery. In Chapter 5, you find out how to hide elements. The following selectors will come in handy then:

✓ **:hidden:** Selects all hidden elements

✓ **:visible:** Selects all visible elements

Selecting Parents and Children

Often the elements you need to select are nested inside other elements. The following code shows two elements, each with the same content inside:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

```
<html>
<head>
<title>My Test Page</title>
<script type=”text/javascript” src=”js/jquery-1.4.min.js”></script>
<script type=”text/javascript”>
$(document).ready(function(){
// Your code goes here.
});
</script>
</head>
<body>
<div id=”myfirstdiv”>
<strong class=”changemytext”>some name</strong>
<p class=”changemytext”>Some text<p>
<strong>another name</strong>
<p>More text<p>
</div>
<div id=”myseconddiv”>
<strong class=”changemytext”>some name</strong>
<p class=”changemytext”>Some text<p>
<strong>another name</strong>
<p>More text<p>
</div>
</body>
</html>
```

The outer element is considered the parent, and inner elements are the children. To select elements based on their parents or children, try these selectors:

✓ **:first-child:** Selects the first child element. The following code selects the first child of the first and changes the text of the selected element:

```
$(‘div:first-child’).text(‘Change me.’);
```

✓ **:last-child:** Selects the last child element. The following code selects the last child of the second and changes the text of the selected element:

```
$(‘div:last-child’).text(‘Change me.’);
```

✓ **parent** > **child:** Selects the child element of the parent element. This code changes the text of every element that is a child of a element.

```
$(‘div > strong’).text(‘Change me.’);
```