

4. jQuery Selectors

jQuery selectors are one of the most important parts of the jQuery library.

4.1. jQuery Selectors

jQuery selectors allow you to select and manipulate HTML element(s).

jQuery selectors are used to "find" (or select) HTML elements based on their id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.

All selectors in jQuery start with the dollar sign and parentheses:

```
$ ( )
```

4.2. The element Selector

The jQuery element selector selects elements based on the element name.

You can select all <p> elements on a page like this:

```
$ ("p")
```

Example

When a user clicks on a button, all <p> elements will be hidden:

Example

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("p").hide();  
    });  
});
```

4.3. The #id Selector

The jQuery #id selector uses the id attribute of an HTML tag to find the specific element.

An id should be unique within a page, so you should use the #id selector when you want to find a

single, unique element.

To find an element with a specific id, write a hash character, followed by the id of the element:

```
$("#test")
```

Example

When a user clicks on a button, the element with id="test" will be hidden:

Example

```
$(document).ready(function() {  
    $("button").click(function() {  
        $("#test").hide();  
    });  
});
```

4.4. The .class Selector

The jQuery class selector finds elements with a specific class.

To find elements with a specific class, write a period character, followed by the name of the class:

```
$(".test")
```

Example

When a user clicks on a button, the elements with class="test" will be hidden:

Example

```
$(document).ready(function() {  
    $("button").click(function() {  
        $(".test").hide();  
    });  
});
```

4.5. More Examples of jQuery Selectors

Syntax	Description
<code>\$("*")</code>	Selects all elements
<code>\$(this)</code>	Selects the current HTML element
<code>\$("p.intro")</code>	Selects all <code><p></code> elements with <code>class="intro"</code>
<code>\$("p:first")</code>	Selects the first <code><p></code> element
<code>\$("ul li:first")</code>	Selects the first <code></code> element of the first <code></code>
<code>\$("ul li:first-child")</code>	Selects the first <code></code> element of every <code></code>
<code>\$("[href]")</code>	Selects all elements with an href attribute
<code>\$("a[target='_blank']")</code>	Selects all <code><a></code> elements with a target attribute value equal to <code>"_blank"</code>
<code>\$("a[target!='_blank']")</code>	Selects all <code><a></code> elements with a target attribute value NOT equal to <code>"_blank"</code>
<code>\$(":button")</code>	Selects all <code><button></code> elements and <code><input></code> elements of <code>type="button"</code>
<code>\$("tr:even")</code>	Selects all even <code><tr></code> elements
<code>\$("tr:odd")</code>	Selects all odd <code><tr></code> elements

For a complete reference of all the jQuery selectors, please go to our [jQuery Selectors Reference](#).

4.6. Functions In a Separate File

If your website contains a lot of pages, and you want your jQuery functions to be easy to maintain, you can put your jQuery functions in a separate .js file.

When we demonstrate jQuery in this tutorial, the functions are added directly into the `<head>` section. However, sometimes it is preferable to place them in a separate file, like this (use the `src` attribute to refer to the .js file):

Example

```
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/j
query/1.10.1/jquery.min.js">
</script>
<script
src="my_jquery_functions.js"></script>
</head>
```