

# jQuery Selectors

- jQuery Selectors are used to select HTML element(s) from an HTML document. Consider an HTML document is given and you need to select all the <div> from this document. This is where jQuery Selectors will help.
- jQuery Selectors can find HTML elements (ie. Select HTML elements) based on the following:
  - HTML element Name
  - Element ID
  - Element Class
  - Element attribute name
  - Element attribute value
  - Many more criteria
- The jQuery library harnesses the power of **Cascading Style Sheets** (CSS) selectors to let us quickly and easily access elements or groups of elements in the Document Object Model (DOM).
- jQuery Selectors works in very similar way on an HTML document like an **SQL Select Statement** works on a Database to select the records.

# jQuery Selector Syntax

```
$(document).ready(function()  
{  
    $(selector)  
});
```

A jQuery selector starts with a dollar sign **\$** and then we put a **selector** inside the braces **()**. Here **\$()** is called **factory function**, which makes use of following three building blocks while selecting elements in a given document:

- The **element** Selector Represents an HTML element name available in the DOM. For example **\$('p')** selects all paragraphs **<p>** in the document.
- The **#id** Selector Represents a HTML element available with the given ID in the DOM. For example **\$('#some-id')** selects the single element in the document that has **some-id** as element Id.
- The **.class** Selector Represents a HTML elements available with the given class in the DOM. For example **\$('.some-class')** selects all elements in the document that have a class of **some-class**.

# The element Selector

- The jQuery **element** selector selects HTML element(s) based on the element name. Following is a simple syntax of an element selector:

```
$(document).ready(function()  
{  
  $("Html Element Name" )});
```

- Please note while using element name as jQuery Selector, we are not giving angle braces along with the element. For example, we are giving only plain **p** instead of **<p>**.

- Examples

```
<html>
```

```
<head>
```

```
  <title></title>
```

```
<script>
```

```
  $(document).ready(function()  
  {  
    $("p").css("background-color", "red");  
  });
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<h1>jQuery element Selector</h1>
```

```
  <p>This is p tag</p>
```

```
</body>
```

```
</html>
```

# The #id Selector

- The jQuery **#id** selector selects an HTML element based on the element **id** attribute. Following is a simple syntax of a **#id** selector:

```
$(document).ready(function(){  
    $("#id of the element")  
});
```

- To use jQuery **#id** selector, you need to make sure that **id** attribute should be uniquely assigned to all the DOM elements. If your elements will have similar ids then it will not produce correct result.
- Examples

```
<html>  
<script>  
$(document).ready(function() {  
    $("#pp").css("background-color", "red"); });  
</script>  
<body>  
<h1>jQuery #id Selector</h1>  
<p id="pp">This is p tag</p>  
</body> </html>
```

# The .class Selector

- The jQuery **.class** selector selects HTML element(s) based on the element **class** attribute. Following is a simple syntax of a **.class** selector:

```
$(document).ready(function(){  
    $(".class of the element")  
});
```

- Because a class can be assigned to multiple HTML elements with in an HTML document, so it is very much possible to find out multiple elements with a single **.class** selector statement.
- Examples

```
<html>  
<script>  
    $(document).ready(function()  
{ $(".pp").css("background-color", "red"); });  
</script>  
<body>  
<h1>jQuery .class Selector</h1>  
<p class="pp">This is p tag</p>  
</body>  
</html>
```