# JQuery

# What is jQuery?

- jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto − **Write less, do more**.
- jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

# Features

- **DOM manipulation** − jQuery made it easy to select DOM elements, traverse them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
- **Event handling** − jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- **AJAX Support** − jQuery helps you a lot to develop a responsive and feature-rich site using AJAX technology.

# Features

- **Animations** − jQuery comes with plenty of built-in animation effects which you can use in your websites.
- **Lightweight** − jQuery is very lightweight library - about 19KB in size ( Minified and gzipped ).
- **Cross Browser Support** − jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome & Opera 9.0+
- **Latest Technology** − The jQuery supports CSS3 selectors and basic XPath syntax.

# Relesae Notes

- https://blog.jquery.com/2021/03/02/jquery-3-6-0-released/

# Slim build

- Along with the regular version of jQuery that includes the ajax and animation effects modules, we've released a "slim" version that excludes these modules. The size of jQuery is very rarely a load performance concern these days, but the slim build is about 6k gzipped bytes smaller than the regular version. These files are also available in the npm package and on the CDN:
- https://code.jquery.com/jquery-3.6.0.slim.js
- https://code.jquery.com/jquery-3.6.0.slim.min.js

# jQuery Source Map

- What is jQuery Source Map? As the name suggests, it consists of a whole bunch of information that can be used **to map the code within a compressed file back to it's original source** . It is used by browser's debugger to help developers debug the minified version of script file.

- https://code.jquery.com/jquery-3.6.0.min.map

# How to use jQuery?

- **Local Installation −** You can download jQuery library on your local machine and include it in your HTML code.
- **CDN Based Version −** You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).

# Local Installation

- Go to the https://jquery.com/download/ to download the latest version available.
- Now put downloaded **jquery-3.6.0.min.js** file in a directory of your website, e.g. /jquery.

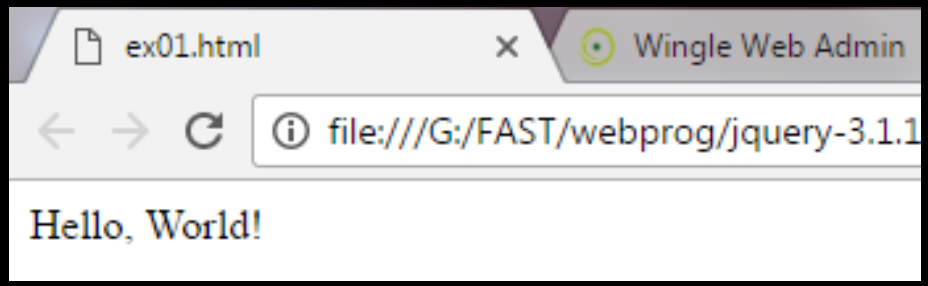
# How to call a jQuery library functions?

- If you want an event to work on your page, you should call it inside the $(document).ready() function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.
- To do this, we register a ready event for the document as follows :

```
$(document).ready(function() {
   // do stuff when DOM is ready
});
```

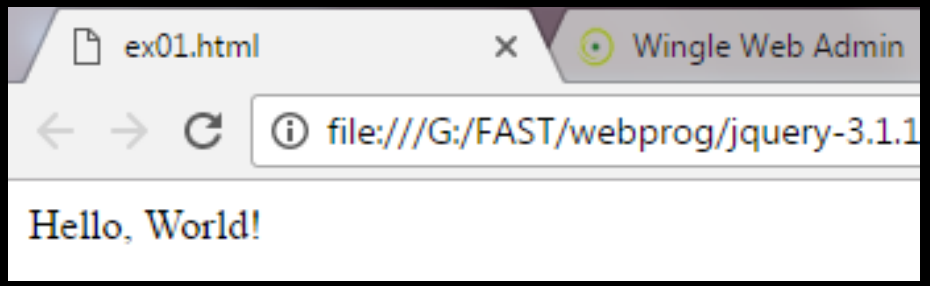
# Example 1

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript">
      $(document).ready(function(){
        document.write("Hello, World!");
      });
    </script>
  </head>
  <body>
    <h1>Hello</h1>
  </body>
</html>
```



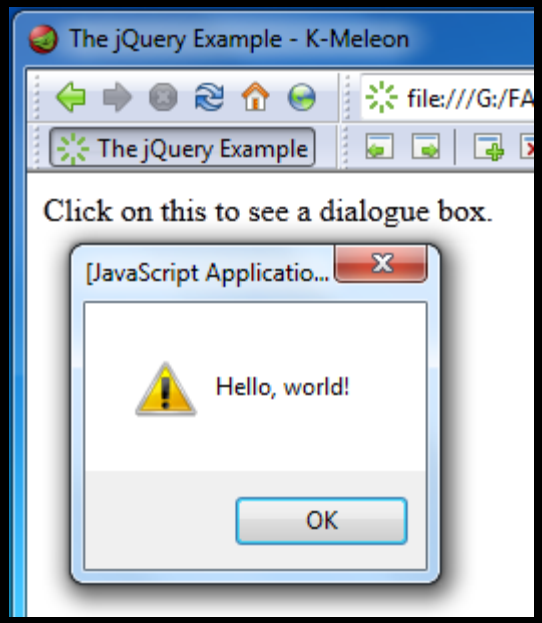
# Example 2 - CDN

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script src="https://ajax.googleapis.com/ajax/libs/
        jquery/3.6.0/jquery.min.js"></script>
    <script type = "text/javascript">
      $(document).ready(function(){
        document.write("Hello, World!");
      });
    </script>
  </head>
  <body>
    <h1>Hello</h1>
  </body>
</html>
```



# Example 3

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("div").click(function() {alert("Hello, world!");});
      });
    </script>
  </head>
  <body>
    <div id = "mydiv">
      Click on this to see a dialogue box.
    </div>
  </body>
</html>
```



# Example 4

```
<html>
   <head>
      <title>The jQuery Example</title>
      <script src = "jquery-3.6.0.js"></script>
      <script type = "text/javascript" language = "javascript">
         $(document).ready(function() {
            $("div").click(function() {
               $(this).hide();
            });
         });
      </script>
   </head>
   <body>
      <div id = "mydiv">
         Click on this to hide.
      </div>
   </body>
</html>
```

# How to use Custom Scripts?

```
<html> <head>
      <title>jQuery customized js file</title>
      <script src = "jquery-3.6.0.js"></script>
      <script type = "text/javascript" src = "ex05.js"></script>
   </head>
   <body>
      <div id = "mydiv"> Click on this to hide.</div>
   </body>
</html>
```

```
$(document).ready(function() {

   $("div").click(function() {
      $(this).hide();
   });

});
```

← ex05.js

# Context

- JavaScript famous keyword **this** always refers to the current context. Within a function **this** context can change, depending on how the function is called

```
$(document).ready(function() {
   // this refers to window.document
});

$("div").click(function() {
   // this refers to a div DOM element
});
```

# Callback

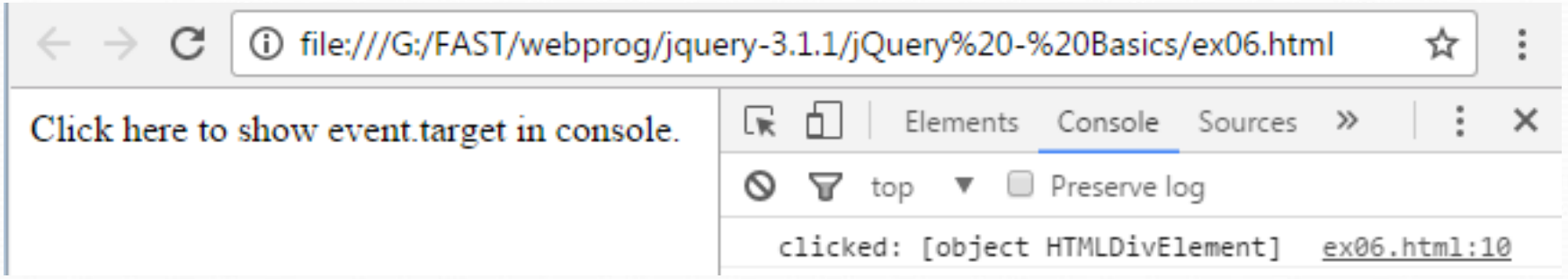
- A callback is a plain JavaScript function passed to some method as an argument or option. Some callbacks are just events, called to give the user a chance to react when a certain state is triggered.
- jQuery's event system uses such callbacks eve

```
$("body").click(function(event) {
   console.log("clicked: " + event.target);
});
```

- Most callbacks provide arguments and a context. In the event-handler example, the callback is called with one argument, an Event.

# Example - callback

```
<script src = "jquery-3.6.0.js"></script>
      <script type = "text/javascript">
            $(document).ready(function() {
           $("div").click(function(event) {
              console.log("clicked: " + event.target);
           });
         });
      </script>
.....
   <body>
      <div id = "mydiv"> Click here to show event.target in console.</div>
   </body>
```



# Callback

- To prevent a form submission, a submit event handler can return false as follows:

```
$("#myform").submit(function() {
   return false;
});
```

# jQuery - Selectors

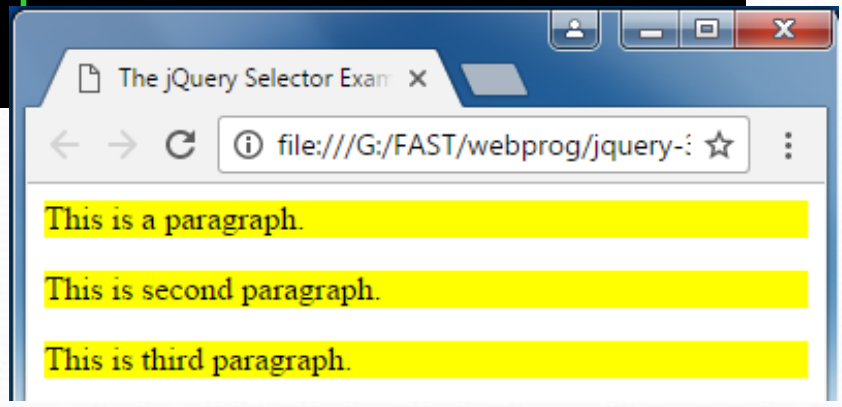
# jQuery - Selectors

- The jQuery library harnesses the power of CSS selectors to quickly and easily access elements or groups of elements in the DOM.
- **The $() factory function**
  - jQuery selectors start with the dollar sign and parentheses − **$()**.
  - The factory function $() is a synonym of jQuery() function.

| | **Selector & Description** |
|---|---|
| 1 | **Tag Name:** Represents a tag name available in the DOM. For example $('p') selects all paragraphs <p> in the document. |
| 2 | **Tag ID:** Represents a tag available with the given ID in the DOM. For example $('#some-id') selects the single element in the document that has an ID of some-id. |
| 3 | **Tag Class:** Represents a tag 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. |

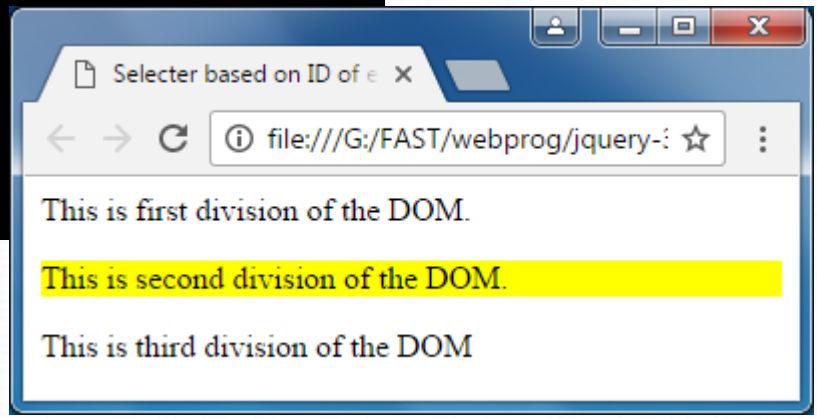
# jQuery - Selectors

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("p").css("background-color", "yellow");
      });
    </script>
        .. .. ..
    <div>
      <p class = "myclass">This is a paragraph.</p>
      <p id = "myid">This is second paragraph.</p>
      <p>This is third paragraph.</p>
    </div>
```



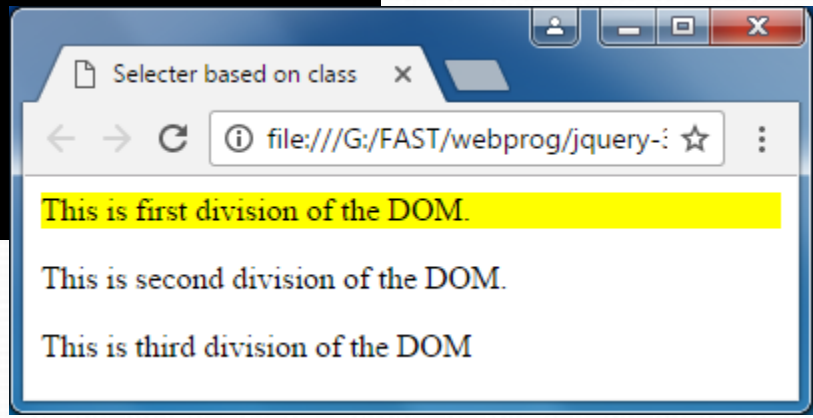
# Selecter based on ID

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("#div2").css("background-color", "yellow");
      });
    </script>
    ....
    <div class = "big" id = "div1">
      <p>This is first division of the DOM.</p>
    </div>
    <div class = "medium" id = "div2">
      <p>This is second division of the DOM.</p>
    </div>
    <div class = "small" id = "div3">
      <p>This is third division of the DOM</p>
    </div>
```



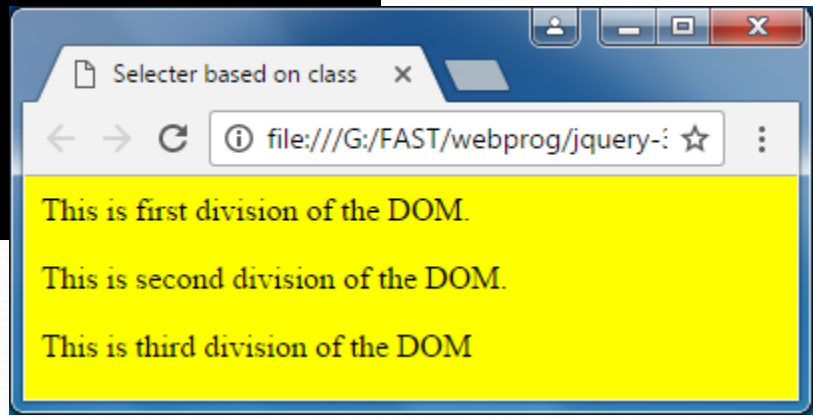
# Selecter based on class

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $(".big").css("background-color", "yellow");
      });
    </script>
    ....
    <div class = "big" id = "div1">
      <p>This is first division of the DOM.</p>
    </div>
    <div class = "medium" id = "div2">
      <p>This is second division of the DOM.</p>
    </div>
    <div class = "small" id = "div3">
      <p>This is third division of the DOM</p>
    </div>
```



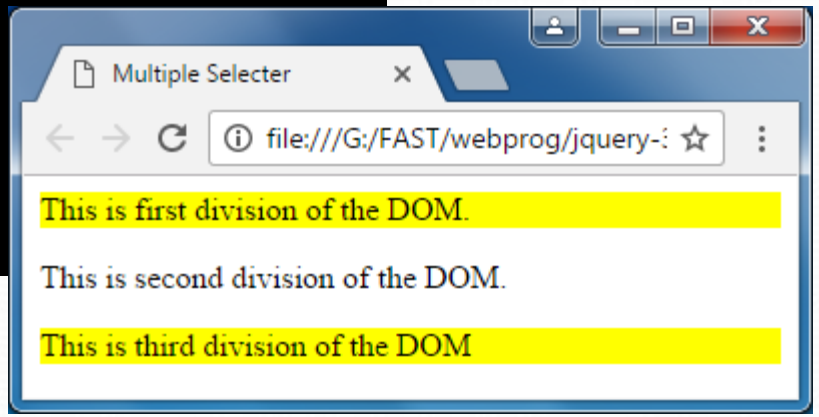
# Universal Selecter (*)

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("*").css("background-color", "yellow");
      });
    </script>
    ....
    <div class = "big" id = "div1">
      <p>This is first division of the DOM.</p>
    </div>
    <div class = "medium" id = "div2">
      <p>This is second division of the DOM.</p>
    </div>
    <div class = "small" id = "div3">
      <p>This is third division of the DOM</p>
    </div>
```



# Multiple Selecter

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $(".big, #div3").css("background-color", "yellow");
      });
    </script>
    ….
    <div class = "big" id = "div1">
      <p>This is first division of the DOM.</p>
    </div>
    <div class = "medium" id = "div2">
      <p>This is second division of the DOM.</p>
    </div>
    <div class = "small" id = "div3">
      <p>This is third division of the DOM</p>
    </div>
```



# jQuery - Attributes

# jQuery - Attributes

- Consider the following HTML markup for an image element:

```
<img id = "imageid" src = "image.gif" alt = "Image" class = "myclass"
   title = "This is an image"/>
```

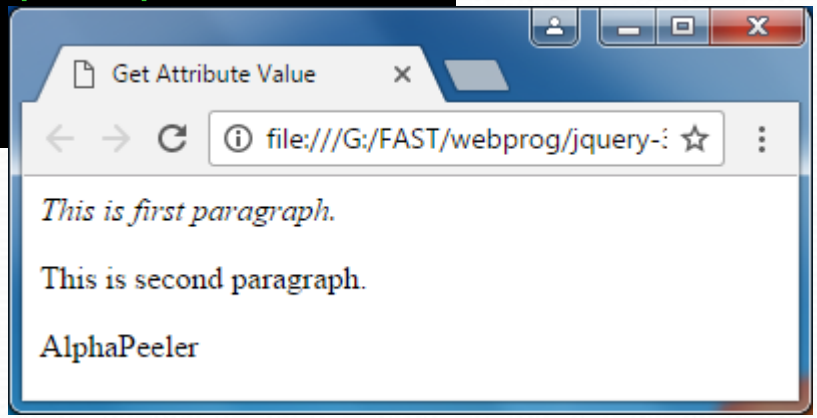
- the tag name is img, and the markup for id, src, alt, class, and title represents the element's attributes, each of which consists of a name and a value.
- jQuery gives us the means to easily manipulate an element's attributes and gives us access to the element so that we can also change its properties.

# jQuery - Attributes

- **Get Attribute Value**
- The **attr()** method can be used to either fetch the value of an attribute from the first element in the matched set or set attribute values onto all matched elements.
- **Set Attribute Value**
- The **attr(name, value)** method can be used to set the named attribute onto all elements in the wrapped set using the passed value.

# Get Attribute Value

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        var title = $("em").attr("title");
        $("#divid").text(title);
      });
    </script>
        ......
    <div>
      <em title = "AlphaPeeler">This is first paragraph.</em>
      <p id = "myid">This is second paragraph.</p>
      <div id = "divid"></div>
    </div>
```



# Set Attribute Value

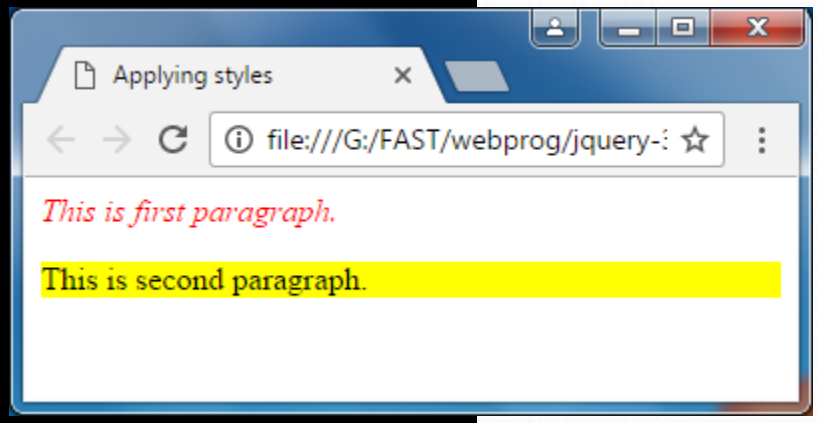
```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("#myimg").attr("src", "/jquery/images/jquery.jpg");
      });
    </script>
    ......
    <div>
      <img id = "myimg" src = "jquery.jpg" alt = "image" />
    </div>
```

# Applying Styles

- The **addClass( classes )** method can be used to apply defined style sheets onto all the matched elements. You can specify multiple classes separated by space.

# Applying Styles

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("em").addClass("selected");
        $("#myid").addClass("highlight");
      });
    </script>
    <style>
      .selected { color:red; }
      .highlight { background:yellow; }
    </style>
     .....
    <em>This is first paragraph.</em>
    <p id = "myid">This is second paragraph.</p>
```



# Attribute Methods

- Following table lists down few useful methods which you can use to manipulate attributes and properties:
- attr( properties ) : Set a key/value object as properties to all matched elements.
- attr( key, fn ): Set a single property to a computed value, on all matched elements.
- removeAttr( name ): Remove an attribute from each of the matched elements.
- hasClass( class ): Returns true if the specified class is present on at least one of the set of matched elements.

# Attribute Methods

- removeClass( class ): Removes all or the specified class(es) from the set of matched elements.
- toggleClass( class ): Adds the specified class if it is not present, removes the specified class if it is present.
- html( ): Get the html contents (innerHTML) of the first matched element.
- html( val ): Set the html contents of every matched element.
- text( ): Get the combined text contents of all matched elements.
- text( val ): Set the text contents of all matched elements.
- val( ): Get the input value of the first matched element.

# Attribute Methods

- val( val ): Set the value attribute of every matched element if it is called on <input> but if it is called on <select> with the passed <option> value then passed option would be selected, if it is called on check box or radio box then all the matching check box and radiobox would be checked.

# attr( properties )

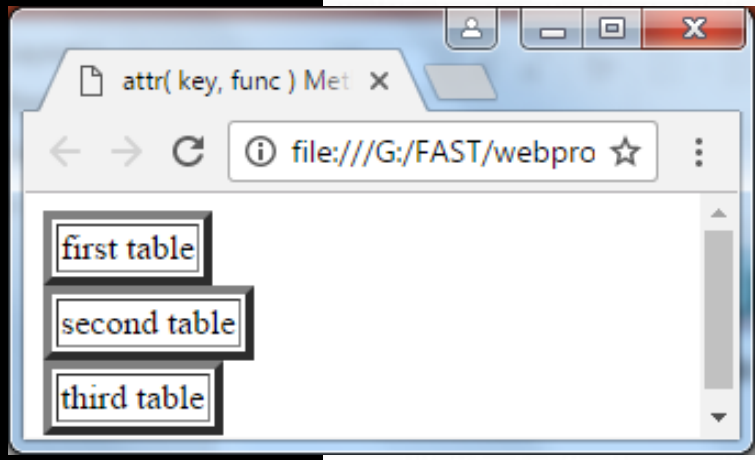
- Set a key/value object as properties to all matched elements.

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("img").attr({
          src: "/images/jquery.jpg",
          title: "jQuery",
          alt: "jQuery Logo"
        });
      });
    </script>
    .....
    <div class = "division" id="divid">
      <p>Following is the logo of jQuery</p>
      <img src = "wrong src" title = "none" alt = "none" />
    </div>
```

# attr( key, func ) Method

- sets a single property to a computed value, on all matched elements.
- **key** − The name of the property to set.
- **func** − A function returning the value to set. This function would have one argument which is index of current element.

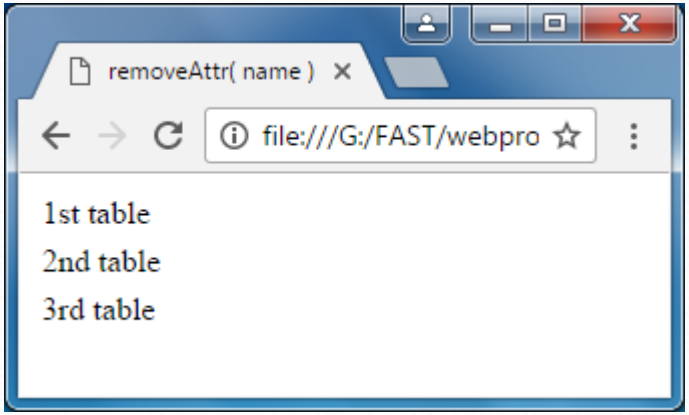
```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
       $("table").attr("border", function(index) {
            return "4px";
         })
      });
    </script>
     .....
  <table> <tr><td>first table</td></tr></table>
  <table><tr><td>second table</td></tr></table>
  <table><tr><td>third table</td></tr></table>
```



# removeAttr( name )

- removes an attribute from each of the matched elements.

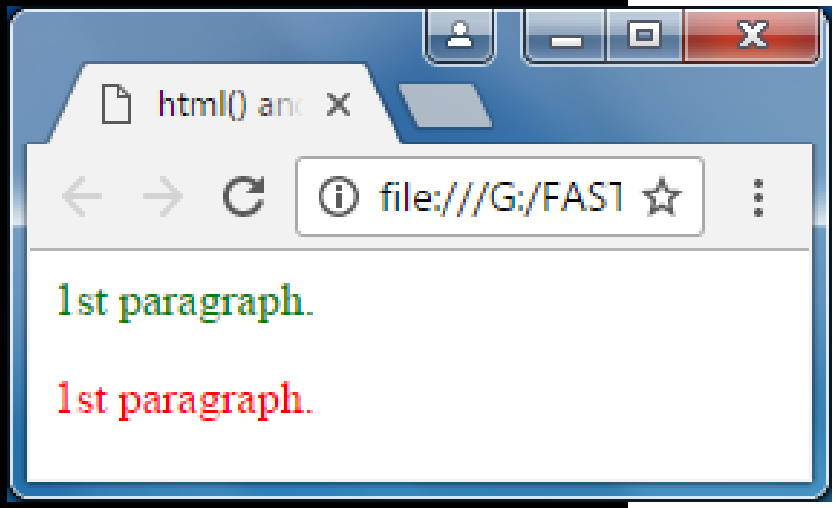
```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("table").removeAttr("border");
      });
    </script>
     .....
    <table border = "2"><tr><td>1st table</td></tr></table>
    <table border = "3"><tr><td>2nd table</td></tr></table>
    <table border = "4"><tr><td>3rd table</td></tr></table>
```



# html() & html( val )

- **html():** Get html contents (innerHTML) of the first matched element.
- **html(val):** Set the html contents of every matched element.

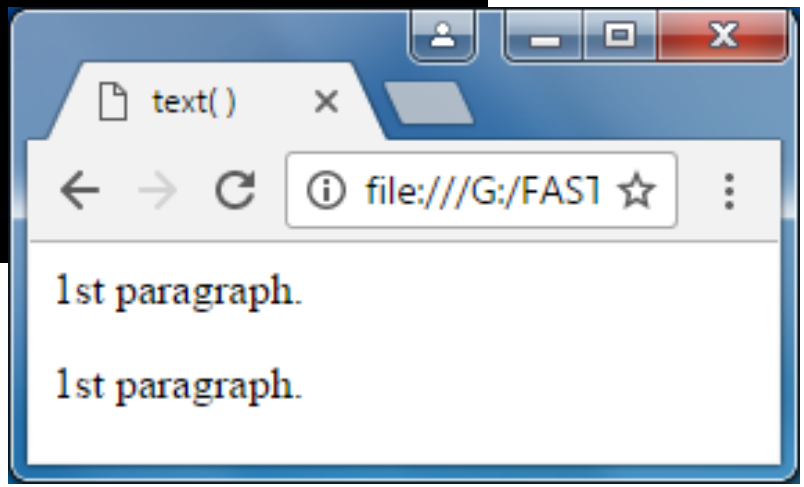
```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        var content = $("p").html();
        $("#pid2").html( content );
      });
    </script>
    <style>
      .red { color:red; }
      .green { color:green; }
    </style>
     …..
    <p class = "green" id = "pid1">1st paragraph.</p>
    <p class = "red" id = "pid2">2nd paragraph.</p>
```



# text( ) & text(val)

- text(): gets combined text contents of all matched elements
- **text(val):** Set the text contents of every matched element.

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        var content = $("p#pid1").text();
        $("#pid2").text(content);
      });
    </script>
     …..
    <p id = "pid1">1st paragraph.</p>
    <p id = "pid2">2nd paragraph.</p>
```



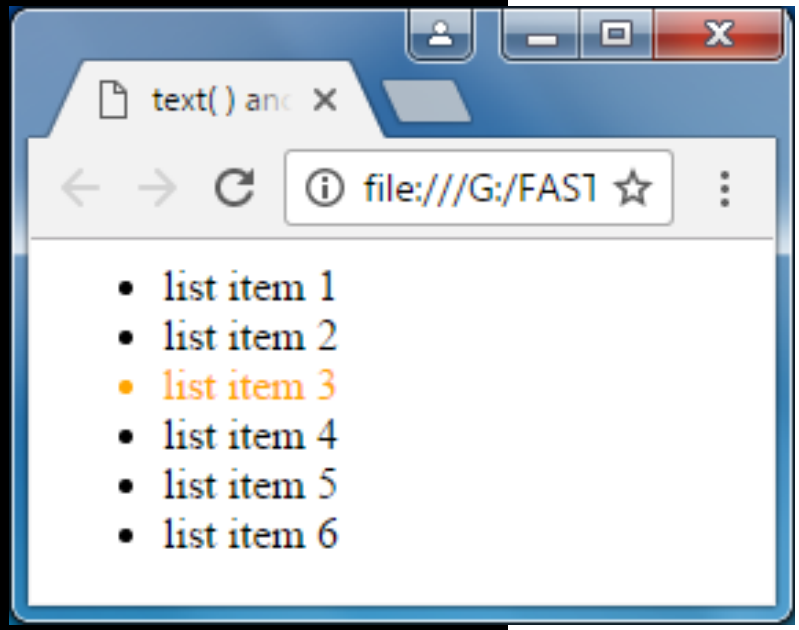
# DOM Traversing

- jQuery is a very powerful tool which provides a variety of DOM traversal methods to help us select elements in a document randomly as well as in sequential method.

# Find Elements by index

- E.g, html list has its index, and can be located by using **eq(index)** method. Every child element starts its index from 0, thus, *list item 2* would be accessed by using **$("li").eq(1)** and so on.

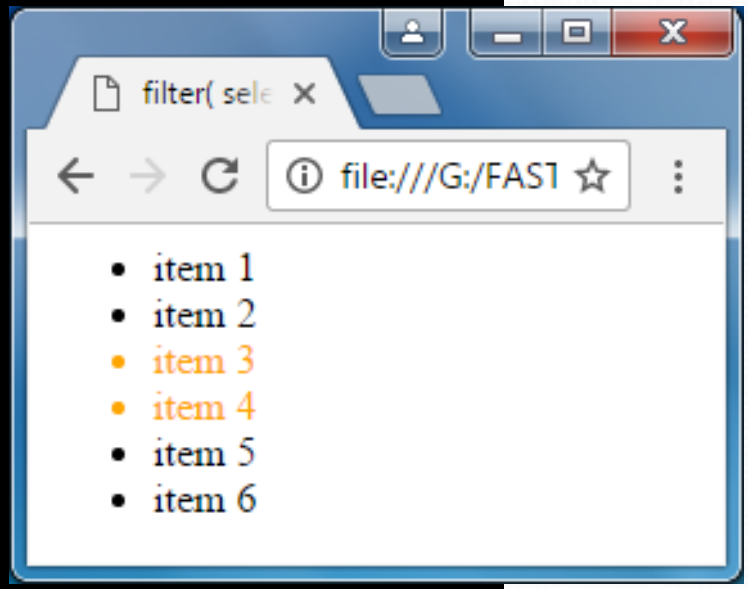
```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").eq(2).addClass("selected");
      });
    </script>
    <style>
      .selected { color:orange; }
    </style>
     .....
    <ul>
      <li>list item 1</li>
      <li>list item 2</li>
      <li>list item 3</li>
      <li>list item 4</li>
      <li>list item 5</li>
      <li>list item 6</li>
```



# Find Elements by filter

- **filter( selector )** method can be used to filter out all elements from the set of matched elements

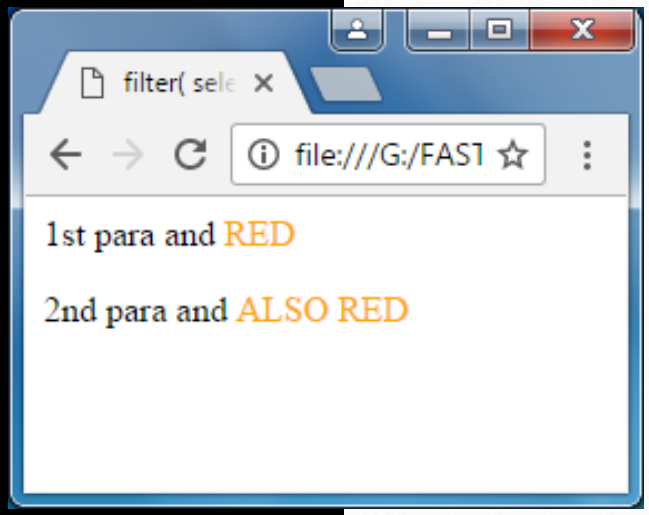
```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").filter(".middle").addClass("selected");
      });
    </script>
    <style>
      .selected { color:orange; } </style>
     .....
   <ul>
     <li class = "top">item 1</li>
     <li class = "top">item 2</li>
     <li class = "middle">item 3</li>
     <li class = "middle">item 4</li>
     <li class = "bottom">item 5</li>
     <li class = "bottom">item 6</li>
   </ul>
```



# Locating descendant Elements

- **find( selector )** method can be used to locate all the descendant elements of a particular type of elements

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("p").find("span").addClass("selected");
      });
    </script>
    <style>
      .selected { color:orange; }
    </style>
     …..
   <p>1st para and <span>RED</span></p>
   <p>2nd para and <span>ALSO RED</span></p>
```



# JQuery DOM Filter Methods

- eq( index ) Reduce the set of matched elements to a single element.
- filter( selector ) Removes all elements from the set of matched elements that do not match the specified selector(s).
- filter( fn ) Removes all elements from the set of matched elements that do not match the specified function.
- is( selector ) Checks the current selection against an expression and returns true, if at least one element of the selection fits the given selector.
- map( callback ) Translate a set of elements in the jQuery object into another set of values in a jQuery array (which may, or may not contain elements).
- not( selector ) Removes elements matching the specified selector from the set of matched elements.
- slice( start, [end] ) Selects a subset of the matched elements.

# JQuery DOM Traversing Methods

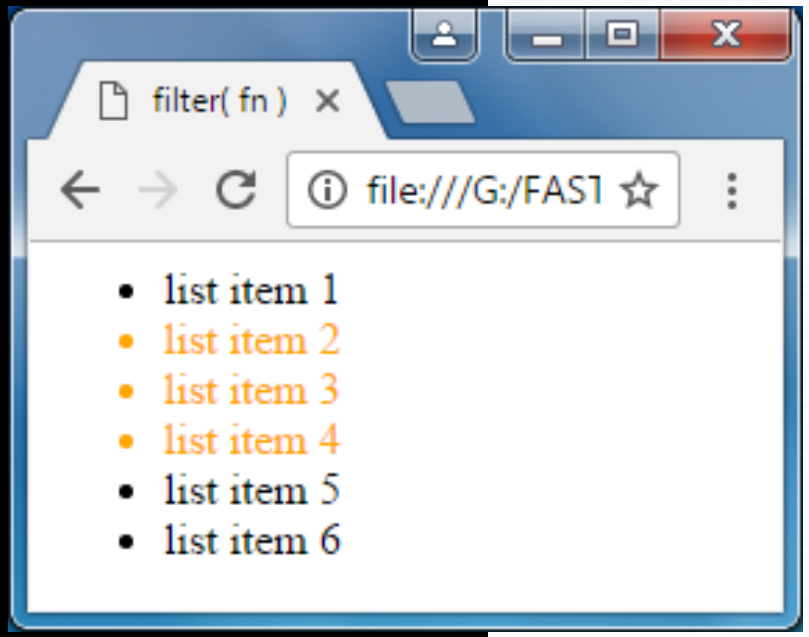
- add( selector ): Adds more elements, matched by the given selector, to the set of matched elements.
- andSelf( ) : Add the previous selection to the current selection.
- children( [selector]): Get a set of elements containing all of the unique immediate children of each of the matched set of elements.
- closest( selector ): Get a set of elements containing the closest parent element that matches the specified selector, the starting element included.
- contents( ): Find all the child nodes inside the matched elements (including text nodes), or the content document, if the element is an iframe.
- end( ): Revert the most recent 'destructive' operation, changing the set of matched elements to its previous state.
- find( selector ): Searches for descendant elements that match the specified selectors.
- next( [selector] ): Get a set of elements containing the unique next siblings of each of the given set of elements.

# JQuery DOM Traversing Methods

- nextAll( [selector] ): Find all sibling elements after the current element.
- offsetParent( ): Returns a jQuery collection with the positioned parent of the first matched element.
- parent( [selector] ): Get the direct parent of an element. If called on a set of elements, parent returns a set of their unique direct parent elements.
- parents( [selector] ) : Get a set of elements containing the unique ancestors of the matched set of elements (except for the root element).
- prev( [selector] ) : Get a set of elements containing the unique previous siblings of each of the matched set of elements.
- prevAll( [selector] ) : Find all sibling elements in front of the current element.
- siblings( [selector] ): Get a set of elements containing all of the unique siblings of each of the matched set of elements.

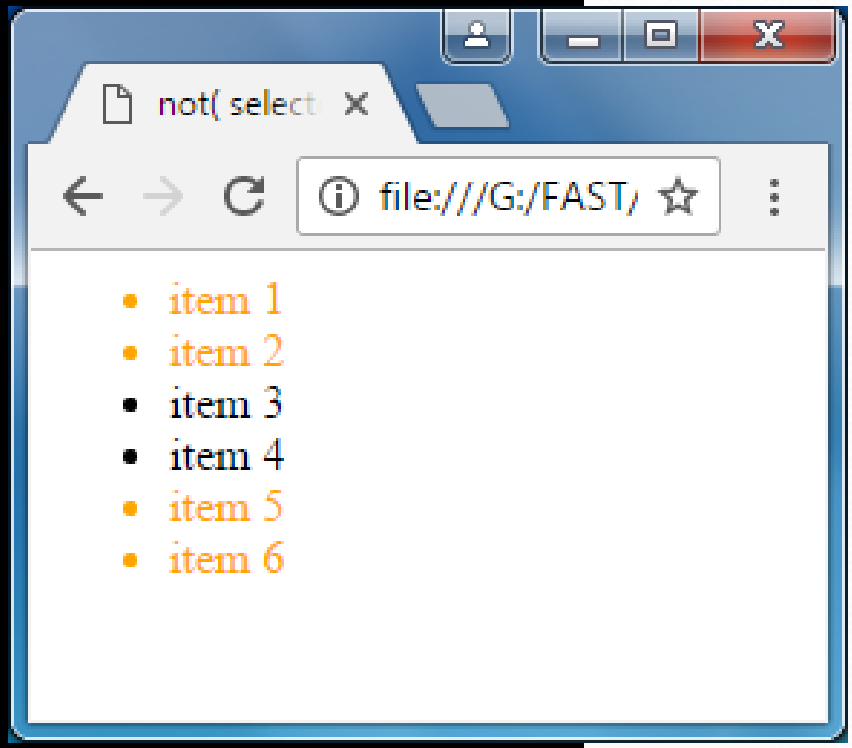
# filter( fn )

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").filter(function (index) {
          return index == 1 || $(this).attr("class") == "middle";
        }).addClass("selected");
      });
    </script>
    <style>
      .selected { color:orange; }
    </style>
     .....
    <ul>
      <li class = "top">list item 1</li>
      <li class = "top">list item 2</li>
      <li class = "middle">list item 3</li>
      <li class = "middle">list item 4</li>
      <li class = "bottom">list item 5</li>
      <li class = "bottom">list item 6</li>
    </ul>
```



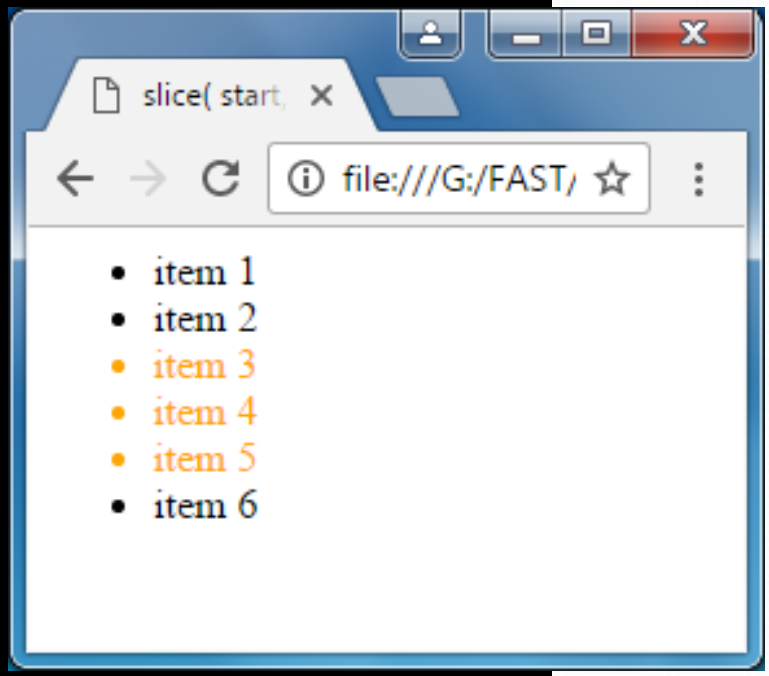
# not( selector )

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").not(".middle").addClass("selected");
      });
    </script>
    <style>
      .selected { color:orange; }
    </style>
     .....
   <ul>
     <li class = "top">item 1</li>
     <li class = "top">item 2</li>
     <li class = "middle">item 3</li>
     <li class = "middle">item 4</li>
     <li class = "bottom">item 5</li>
     <li class = "bottom">item 6</li>
   </ul>
```



# slice( start, end )

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").slice(2, 5).addClass("selected");
      });
    </script>
    <style>
      .selected { color:orange; }
    </style>
     .....
   <ul>
     <li class = "top">item 1</li>
     <li class = "top">item 2</li>
     <li class = "middle">item 3</li>
     <li class = "middle">item 4</li>
     <li class = "bottom">item 5</li>
     <li class = "bottom">item 6</li>
   </ul>
```

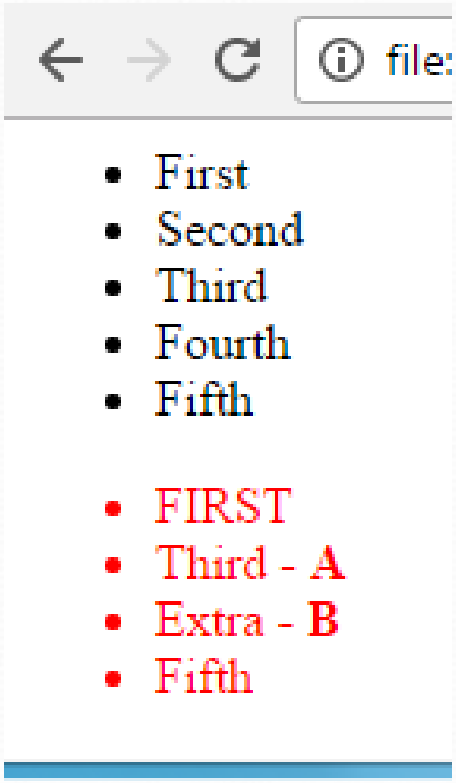


# map(callback)

```
<script src = "jquery-3.6.0.js"></script>
      <script type = "text/javascript">
$(document).ready(function() {
   var mappedItems = $("li").map(function (index) {
       var replica = $("<li>").text($(this).text()).get(0);
       if (index == 0) {
          // make the first item all caps
          $(replica).text($(replica).text().toUpperCase());
       } else if (index == 1 || index == 3) {
          // delete the second and fourth items
          replica = null;
       } else if (index == 2) {
          // make two of the third item and add some text
          replica = [replica,$("<li>").get(0)];
          $(replica[0]).append("<b> - A</b>");
          $(replica[1]).append("Extra <b> - B</b>");
       }
       return replica;
             });
             $("#results").append(mappedItems);
 });</script>
```

# map(callback)

```
<style> #results { color:red; }
</style>
</head>
  <body>
    <ul>
        <li>First</li>
        <li>Second</li>
        <li>Third</li>
        <li>Fourth</li>
        <li>Fifth</li>
      </ul>
      <ul id = "results">
      </ul>
  </body>
</html>
```

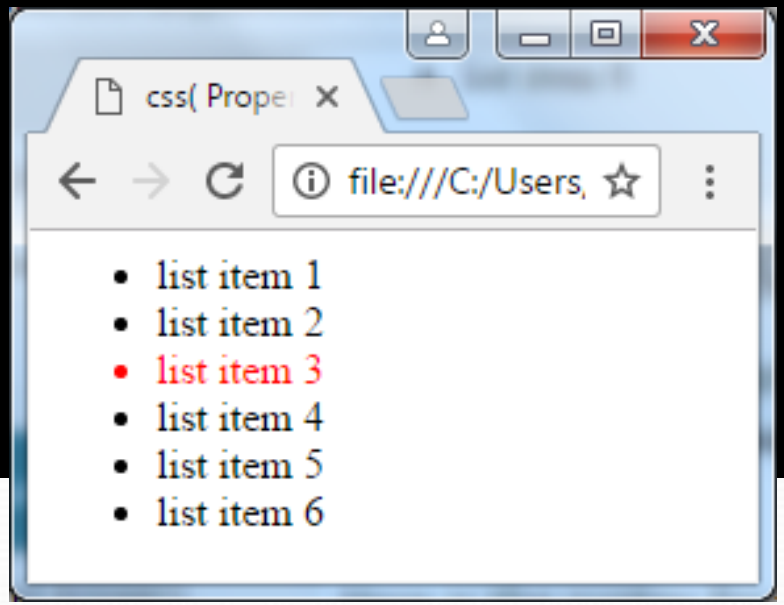


# jQuery - CSS Selectors Methods

- The jQuery library supports nearly all of the selectors included in Cascading Style Sheet (CSS) specifications 1 through 3, as outlined on the World Wide Web Consortium's site.
- Apply CSS Properties
- To apply any CSS property, use JQuery method css( PropertyName, PropertyValue ).
- Here is the syntax for the method:
- selector.css( PropertyName, PropertyValue );

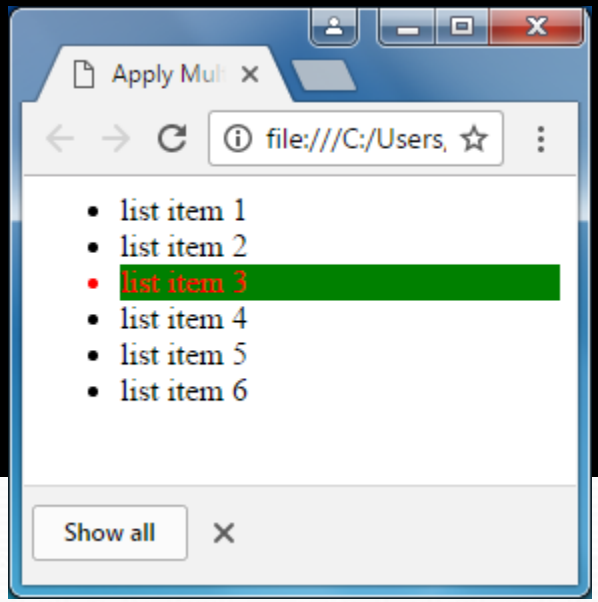
# css( PropertyName, PropertyValue ).

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").eq(2).css("color", "red");
      });
    </script>
     .....
   <ul>
     <li>list item 1</li>
     <li>list item 2</li>
     <li>list item 3</li>
     <li>list item 4</li>
     <li>list item 5</li>
     <li>list item 6</li>
   </ul>
```



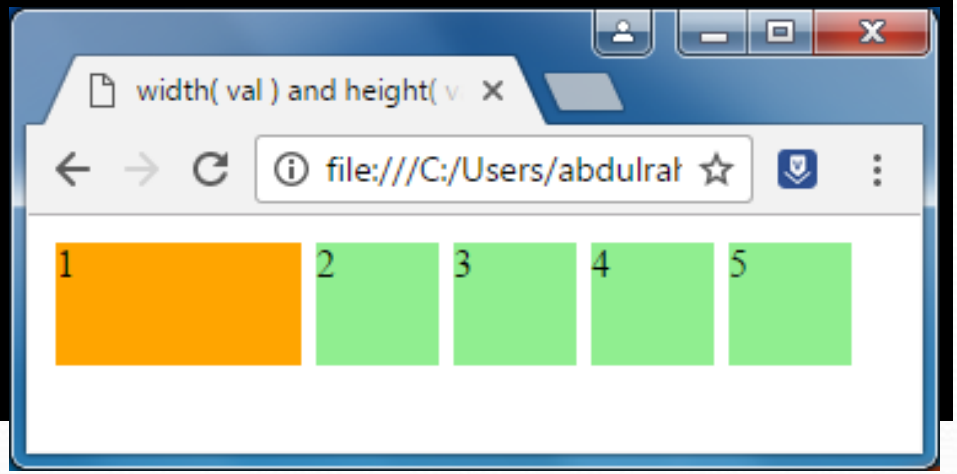
# Apply Multiple CSS Properties

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("li").eq(2).css({"color":"red", "background-color":"green"});
      });
    </script>
     ....
          <ul>
            <li>list item 1</li>
            <li>list item 2</li>
            <li>list item 3</li>
            <li>list item 4</li>
            <li>list item 5</li>
            <li>list item 6</li>
          </ul>
```



# width( val ) and height( val )

```
<script src = "jquery-3.6.0.js"></script>
    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("div:first").width(100);
        $("div:first").css("background-color", "orange");
      });
    </script>
    <style>
      div{ width:50px; height:50px; float:left;
          margin:3px; background:lightgreen; cursor:pointer; }
    </style>
     .....
    <div>1</div>
    <div>2</div>
    <div>3</div>
    <div>4</div>
    <div>5</div>
```



# JQuery CSS Methods

- css( name ): Return a style property on the first matched element.
- css( name, value ): Set a single style property to a value on all matched elements.
- css( properties ): Set a key/value object as style properties to all matched elements.
- height( val ): Set the CSS height of every matched element.
- height( ): Get the current computed, pixel, height of the first matched element.
- innerHeight( ): Gets the inner height (excludes the border and includes the padding) for the first matched element.
- innerWidth( ): Gets the inner width (excludes the border and includes the padding) for the first matched element.
- offset( ) : Get the current offset of the first matched element, in pixels, relative to the document.

# JQuery CSS Methods

- offsetParent( ): Returns a jQuery collection with the positioned parent of the first matched element.
- outerHeight( [margin] ): Gets the outer height (includes the border and padding by default) for the 1st matched element.
- outerWidth( [margin] ) : Get the outer width (includes the border and padding by default) for the 1st matched element.
- position( ) : Gets the top and left position of an element relative to its offset parent.
- scrollLeft( val ) : When a value is passed in, the scroll left offset is set to that value on all matched elements.
- scrollLeft( ): Gets scroll left offset of 1st matched element.
- scrollTop( val ): When a value is passed in, the scroll top offset is set to that value on all matched elements.
- scrollTop( ) : Gets scroll top offset of 1st matched element.
- width( val ) : Set the CSS width of every matched element.
- width( ) : Get the current computed, pixel, width of the first matched element.