JQuery

JQuery

- jQuery is a lightweight, "**write less, do more**", JavaScript library by Jhon Resig in 2006.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.
- The jQuery library contains the following features:
  - HTML/DOM manipulation
  - CSS manipulation
  - HTML event methods
  - Effects and animations
  - AJAX

Why jQuery?

- There are lots of other JavaScript frameworks out there, but jQuery seems to be the most popular, and also the most extendable.
- Many of the biggest companies on the Web use jQuery, such as:
  - Google
  - Microsoft
  - IBM
  - Netflix
- jQuery will run exactly the same in all major browsers.
- Why?
- Easy to manipulate DOM
- Cross browser platform
- Large community
- 1000s of plugin

- **Adding jQuery to Your Web Pages**
- There are several ways to start using jQuery on your web site. You can:
  - Download the jQuery library from jQuery.com
  - Include jQuery from a CDN, like Google
- **Downloading jQuery**
- There are two versions of jQuery available for downloading:
  - **Production version** - this is for your live website because it has been minified and compressed
  - **Development version** - this is for testing and development (uncompressed and readable code)

```
<head>
<script src="jquery-3.3.1.min.js"></script>
</head>
```

- **Do you wonder why we do not have type="text/javascript" inside the <script> tag?**

This is not required in HTML5. JavaScript is the default scripting language in HTML5 and in all modern browsers!

jQuery CDN

- If you don't want to download and host jQuery yourself, you can include it from a CDN (Content Delivery Network).
- Both Google and Microsoft host jQuery.
- **Google CDN:**

```
<head><script
              src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script></head>
```

- **Microsoft CDN:**

```
<head><script  src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-  3.3.1.min.js">
</script></head>
```

jQuery Syntax

- The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).
- Basic syntax is: **$(*selector*).*action*()**
  - A $ sign to define/access jQuery, ($ is simply an alias for jQuery)
  - A (*selector*) to "query (or find)" HTML elements
  - A jQuery *action*() to be performed on the element(s)
- Examples:
- $(this).hide() - hides the current element.
- $("p").hide() - hides all <p> elements.
- $(".test").hide() - hides all elements with class="test".
- $("#test").hide() - hides the element with id="test".

jQuery uses CSS syntax to select elements.

jQuery Selectors

jQuery selectors are used to "find" HTML elements based on their name, 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: $().

- **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")

- **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 HTML element:$("#test")

- **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")

The Document Ready Event

- It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.
- Here are some examples of actions that can fail if methods are run before the document is fully loaded:
- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet
- **Tip:** The jQuery team has also created an even shorter method for the document ready event:

```
$(document).ready(function()
{
  // jQuery methods go here...
})
```

```
$(function(){
  // jQuery methods go here...
})
```

Jquery Effects: Methods on elements

Jquery Effects are Hide, Show, Toggle, Slide, Fade, and Animate

- Hide : hide() Syntax - $(*selector*).hide(*speed,callback*);
- Show: show() Syntax - $(*selector*).show(*speed,callback*);
- Toggle: toggle();
- The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after
  the hide() or show() method completes (you will learn more about callback functions in a later chapter).
- **Slide:**
  - Syntax: $(*selector*).slideDown(*speed,callback*);
- The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
- The optional callback parameter is a function to be executed after the sliding completes.
  - slideDown()- method is used to slide down an element.
  - slideUp() - method is used to slide up an element.
  - slideToggle()- If the elements have been slid down, slideToggle() will slide them up; If the elements have been slid up, slideToggle() will slide them down.

Jquery Effects: Methods on elements –Cont.

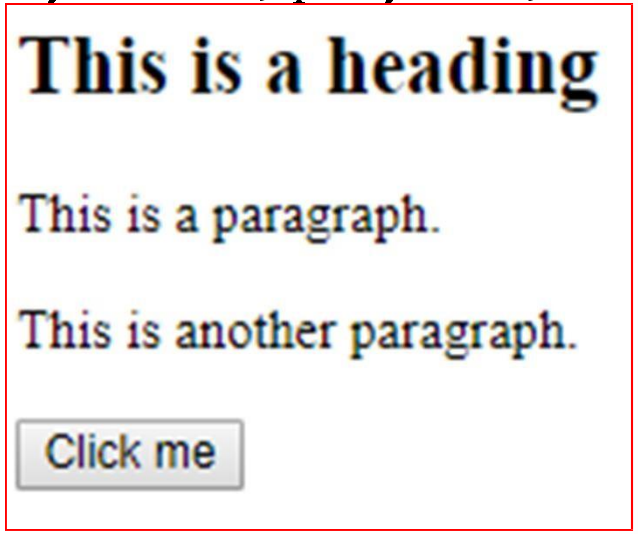
- **Fade:**
  - fadeIn() - method is used to fade in a hidden element.
    - Syntax: $(*selector*).fadeIn(*speed,callback*);
    - The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
    - The optional callback parameter is a function to be executed after the fading completes.
  - fadeOut()- method is used to fade out a visible element.
  - fadeToggle()- method toggles between the fadeIn() and fadeOut() methods.
  - fadeTo() - method allows fading to a given opacity (value between 0 and 1).
- **Animations** - The animate() Method is used to create custom animations
  - Syntax: $(*selector*).animate({*params*},*speed,callback*);
  - The required params parameter defines the CSS properties to be animated.
  - The optional speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds.
  - The optional callback parameter is a function to be executed after the animation completes.

- Note: A callback function is executed after the current effect is 100% finished.
  - Example: $("button").click(function(){ $("p").hide("slow", function(){ alert("The paragraph is now hidden"); }) });

- **jQuery Method Chaining:** To chain an action, you simply append the action to the previous action.
- The following example chains together the css(), slideUp(), and slideDown() methods. The "p1" element first changes to red, then it slides up, and then it slides down:
  - $("#p1").css("color", "red").slideUp(2000).slideDown(2000);

Example – element selector

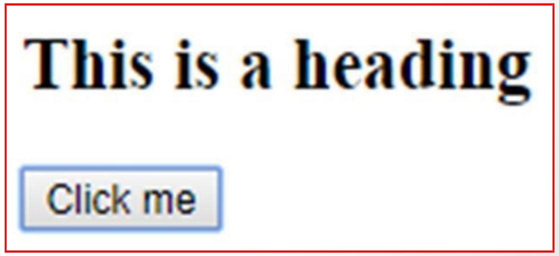
```
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script><script>
$(document).ready(function(){
$("button").click(function(){
$("p").hide();
});
});
</script></head><body>
<h2>This is a heading</h2><p>This is a paragraph.</p>
<p>This is another paragraph.</p><button>Click me</button>
</body></html>
```



This is a heading

This is a paragraph.

This is another paragraph.



This is a heading

Example – id selector

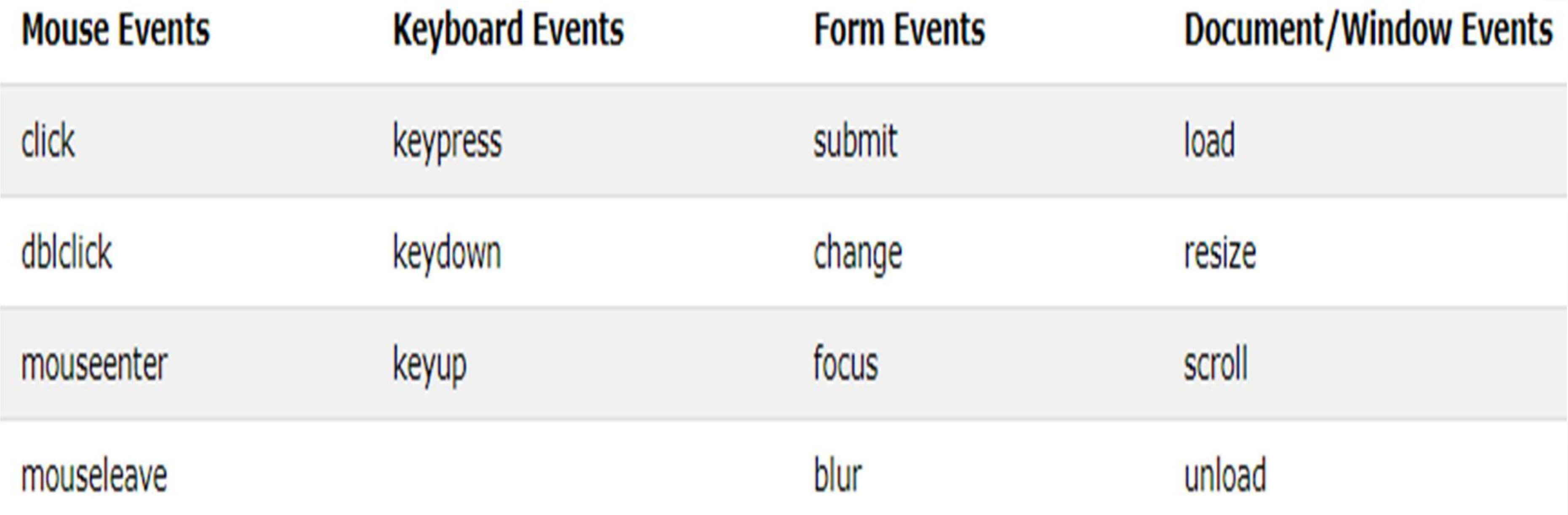
```
<html><head><script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script><script>
$(document).ready(function(){
$("button").click(function(){
$("#test").hide();
});
});
</script></head><body>
<h2>This is a heading</h2><p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>
<button>Click me</button></body></html>
```

Example – class selector

```
<html> <head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.
js"></script><script>
                                        $(document).ready(function(){
                                          $("button").click(function(){
$(".test").hide();
});
});
</script></head><body>
<h2 class="test">This is a heading</h2>
<p class="test">This is a paragraph.</p>
<p>This is another paragraph.</p>
<button>Click me</button></body></html>
```

jQuery Event Methods

- All the different visitor's actions that a web page can respond to are called events.
- An event represents the precise moment when something happens. "The keypress event is fired, the moment you press a key".

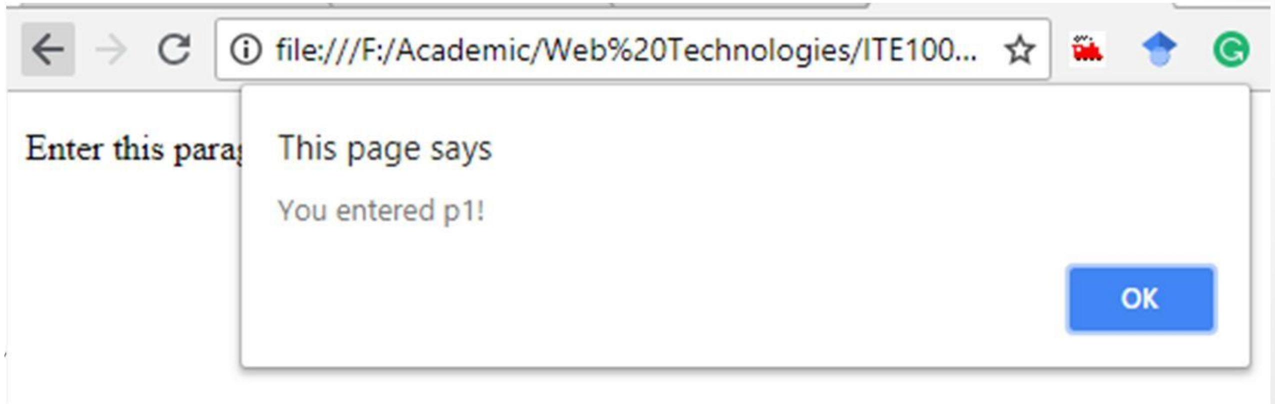


Mouse Events	Keyboard Events	Form Events	Document/Window Events
click	keypress	submit	load
dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

Event Methods

- **click()-**The function is executed when the user clicks on the HTML element.
- **dblclick()-**The function is executed when the user double-clicks on the HTML element
- **mouseenter()-**The function is executed when the mouse pointer enters the HTML element
- **mouseleave()-**The function is executed when the mouse pointer leaves the HTML element
- **mousedown()-**The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element
- **mouseup()-**The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element
- **hover()-**takes two functions and is a combination of the mouseenter() and mouseleave() methods
- **focus()-**The function is executed when the form field gets focus
- **blur()-**The function is executed when the form field loses focus
- **on()-**method attaches one or more event handlers for the selected elements

```
<html><head><script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.
js"></script>
<script>
$(document).ready(function(){
$("#p1").mouseenter(function(){  alert("You entered p1!");
});
}); </script></head><body>
<p id="p1">Enter this paragraph.</p></body></html>
```

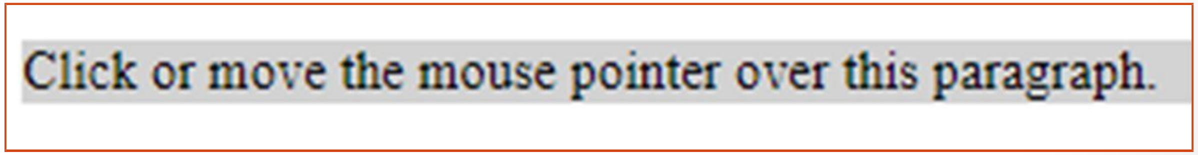


Enter this para

This page says

You entered p1!

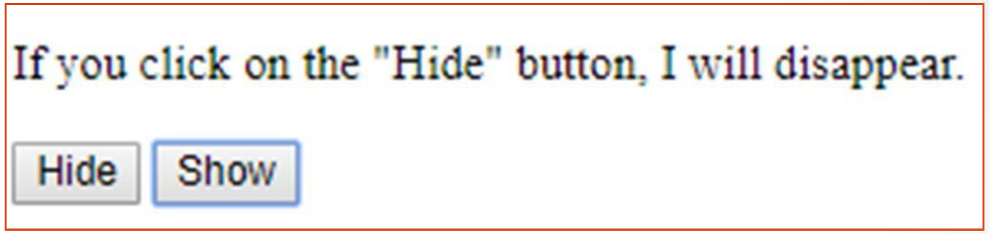
```
<html><head><script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script
><script>
$(document).ready(function(){
$("p").on({
mouseenter: function(){
$(this).css("background-color", "lightgray");
},
mouseleave: function(){
$(this).css("background-color", "lightblue");
},
click: function(){
$(this).css({"background-color”:”yellow”,”font-size”:”200%“});
}           });});
</script></head><body><p>Click or move the mouse pointer over this
paragraph.</p></body></html>
```



Click or move the mouse pointer over this paragraph.

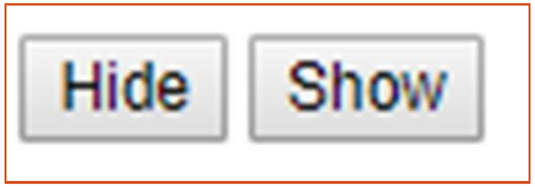
jQuery Effects - Hide and Show

```
$("#hide").click(function(){
$("p").hide();
});
```

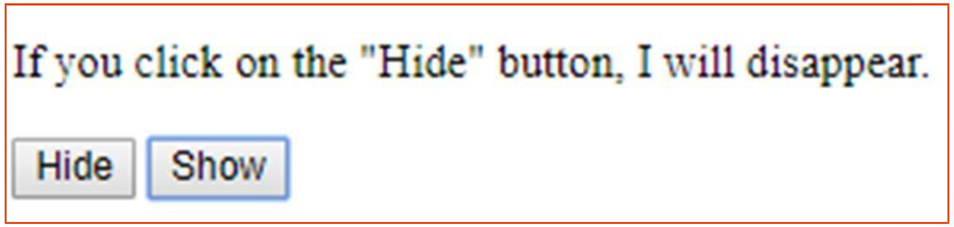


If you click on the "Hide" button, I will disappear.

```
$("#show").click(function(){
$("p").show();
});
```



```
$("button").click(function(){
$("p").hide(1000);
});
```

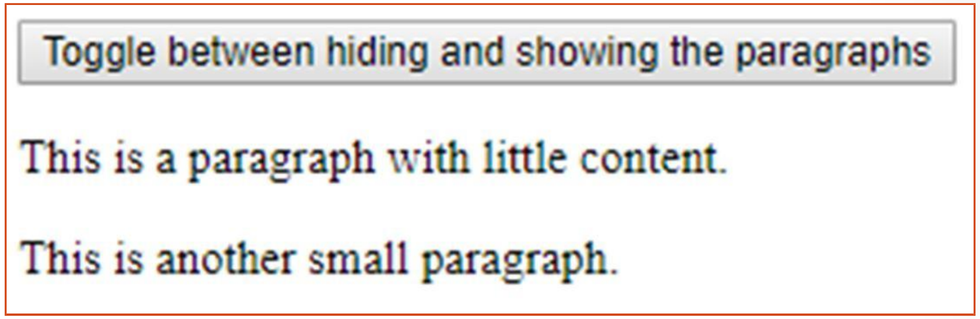


If you click on the "Hide" button, I will disappear.

The optional speed parameter specifies the speed of the hiding/showing, and can take the following values: "slow", "fast", or milliseconds.

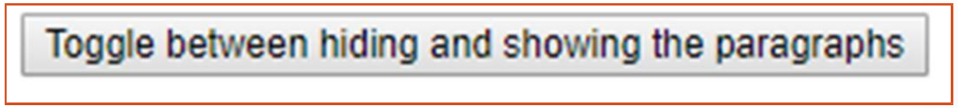
jQuery Effects - toggle

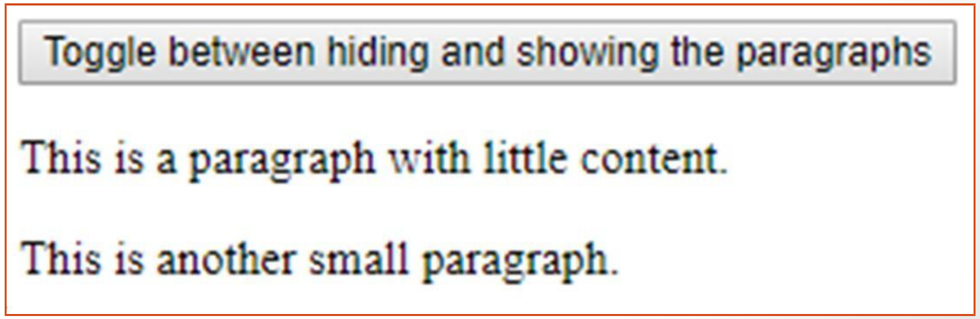
```
<html> <head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.
j  s"></script> <script>
$(document).ready(function(){
$("button").click(function(){
$("p").toggle();
}); });
</script> </head> <body>
<button>Toggle between hiding and showing the paragraphs</button>
<p>This is a paragraph with little content.</p> <p>This is another
small paragraph.</p>
</body> </html>
```



This is a paragraph with little content.

This is another small paragraph.





This is a paragraph with little content.

This is another small paragraph.

jQuery Effects - Fading

- fadeIn() method is used to fade in a hidden element

```
$("button").click(function(){
   $("#div1").fadeIn();
   $("#div2").fadeIn("slow");
   $("#div3").fadeIn(3000); });
```

- fadeOut() method is used to fade out a visible element

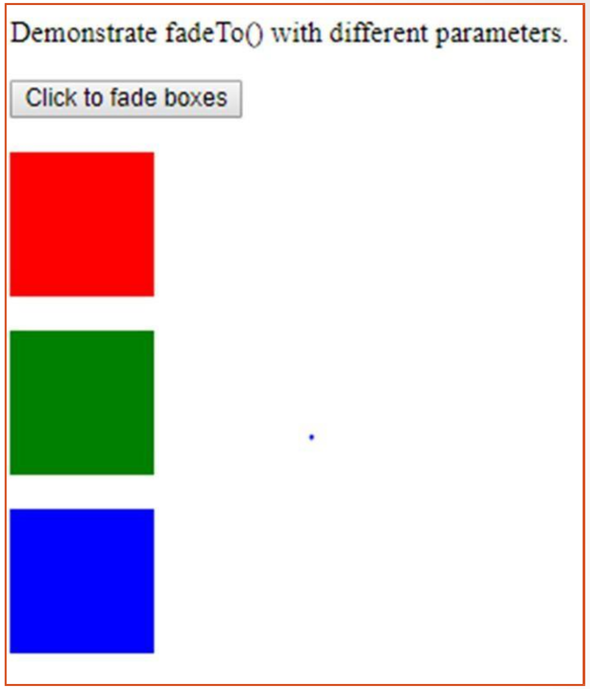
```
$("button").click(function(){
   $("#div1").fadeOut();
   $("#div2").fadeOut("slow");
   $("#div3").fadeOut(3000); });
```

- fadeToggle() method toggles between the fadeIn() and fadeOut() methods

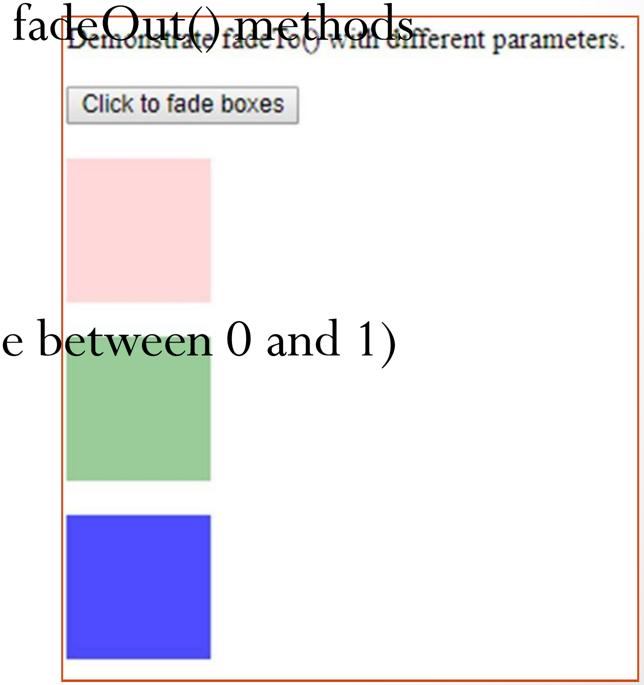
```
$("button").click(function(){
   $("#div1").fadeToggle();
   $("#div2").fadeToggle("slow");
   $("#div3").fadeToggle(3000); });
```

- fadeTo() method allows fading to a given opacity (value between 0 and 1)

```
$("button").click(function(){
   $("#div1").fadeTo("slow",0.15);
   $("#div2").fadeTo("slow",0.4);
   $("#div3").fadeTo("slow",0.7);});
```



Demonstrate fadeTo() with different parameters.



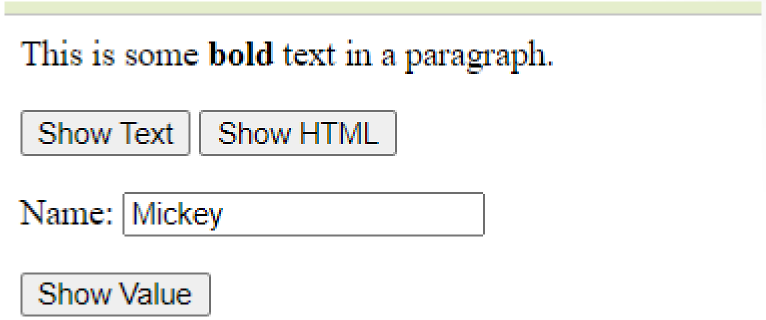
Demonstrate fadeTo() with different parameters.

jQuery DOM Manipulation

- jQuery contains powerful methods for changing and manipulating HTML elements and attributes.
- One very important part of jQuery is the possibility to manipulate the DOM.
- jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.
- Get Content - text(), html(), and val()
    - Three simple, but useful, jQuery methods for DOM manipulation are:
    - text() - Sets or returns the text content of selected elements
    - html() - Sets or returns the content of selected elements (including HTML markup)
    - val() - Sets or returns the value of form fields

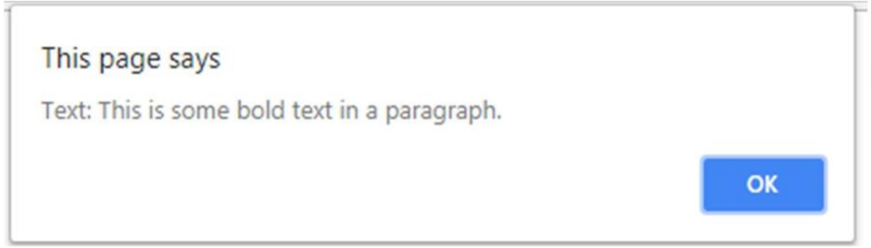
Example -Get Content - text(), html(), and val()

```
<html> <head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery
/3.3.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
$("#btn1").click(function(){
alert("Text: " + $("#test").text()); });
$("#btn2").click(function(){
alert("HTML: " + $("#test").html()); });
$("#btn3").click(function(){
alert("Value: " + $("#t1").val()); });
});
</script>
</head>
<body>
<p id="test">This is some <b>bold</b> text in a
paragraph.</p>
<button id="btn1">Show Text</button>
<button id="btn2">Show HTML</button>
<p>Name: <input type="text" id="t1"
value="Mickey Mouse"></p>
<button id="btn3">Show Value</button>
</body>
</html>
```



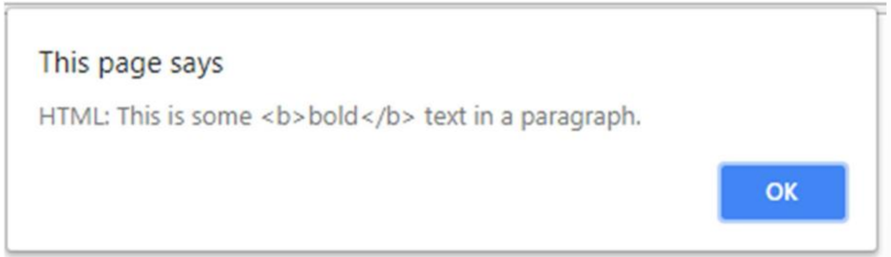
This is some **bold** text in a paragraph.

Name: Mickey



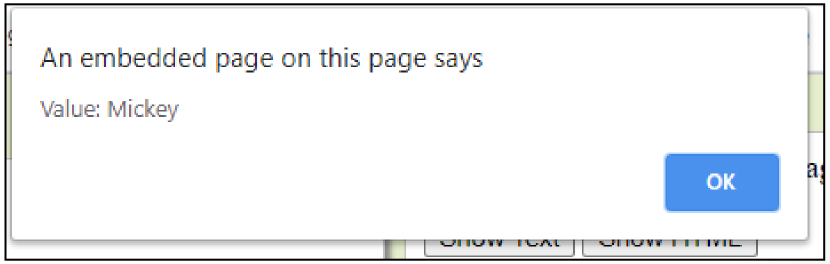
This page says

Text: This is some bold text in a paragraph.



This page says

HTML: This is some <b>bold</b> text in a paragraph.



An embedded page on this page says

Value: Mickey

Add New HTML Content

Five jQuery methods that are used to add new content:

- add() - to add element to the existing group of elements
- append() - Inserts content at the end of the selected elements
- prepend() - Inserts content at the beginning of the selected elements
- after() - Inserts content after the selected elements
- before() - Inserts content before the selected elements

- Example: The several new elements are created with text/HTML, jQuery, and JavaScript/DOM. Then these new elements to the body element with the append() method.

  -
    ```
    function appendText() {
      var txt1 = "<p>Text.</p>";               // Create element with HTML
      var txt2 = $("<p></p>").text("Text.");   // Create with jQuery
      var txt3 = document.createElement("p");  // Create with DOM
      txt3.innerHTML = "Text.";
      $("body").append(txt1, txt2, txt3);      // Append the new elements
    }
    ```

Add element using jQuery

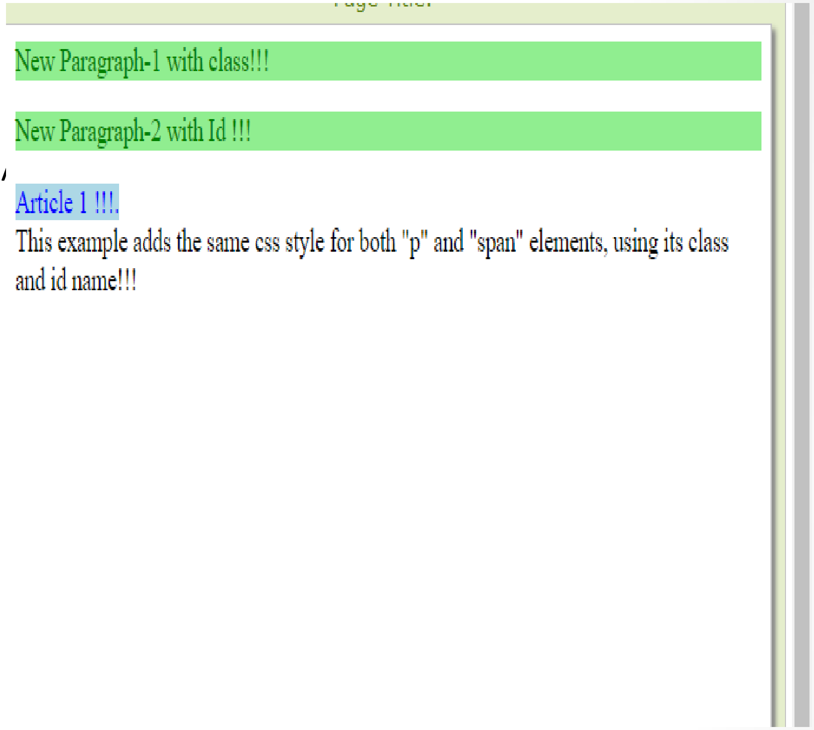
- The **jQuery add()** method is used to add element to the existing group of elements. This method can add element to the whole document, or just inside context element if the context parameter is defined.
- $(selector).add(element, context_parameter)

Example: add() and chaining

```
<html>

<head>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></
script>
   <script>
      $(document).ready(function() {
         $(".heading").add("#para1").add("#span1").
         css("background-color", "lightgreen");
                 $("#span1").css("background-color", "lightblue");
      });
   </script>
</head>

<body>
   <p style="color:green" class="heading">New Paragraph-1 with
class!!!</p>
   <p style="color:green" id="para1">New Paragraph-2 with Id !!!</p>
   <span style="color:blue" id="span1">Article 1 !!!.</span>
   <div>This example adds the different css style for both
      "p" and "span" elements, using its class and
       id name!!! </div>
</body>
</html>
```



New Paragraph-1 with class!!!

New Paragraph-2 with Id !!!

Article 1 !!!.
This example adds the same css style for both "p" and "span" elements, using its class and id name!!!

Remove Elements/Content

- To remove elements and content, there are mainly two jQuery methods:
  - remove() - Removes the selected element (and its child elements)

    ```
    $("#div1").remove();
     $("p").remove(".p1");  //passing a parameter of a particular class/id
    ```

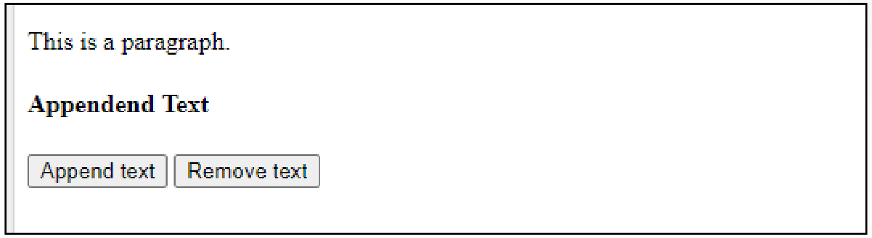
  - empty() - Removes the child elements from the selected element

    ```
    $("#div1").empty();
    ```

Example – append() and remove()

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js
"></script>
<script>
function appendText() {
  var txt1 = "<p>Text1.</p>";        // Create text with HTML
  var txt2 = $("<p></p>").text("Text2.");  // Create text with jQuery
 txt2.attr({class : "p1"});
 var txt3 = document.createElement("p");
  txt3.innerHTML = "Text3.";
// Create text with DOM
  $("body").append(txt1, txt2, txt3);   // Append new elements
  }
  function removeText() {
    $("p").remove(".p1");   // remove p elements with  class name p1
  }
</script>
</head>
<body>
<p>This is a paragraph.</p>
<h4> Appendend Text </h4>
<button onclick="appendText()">Append text</button>
<button onclick="removeText()">Remove text</button>
</body>
</html>
```

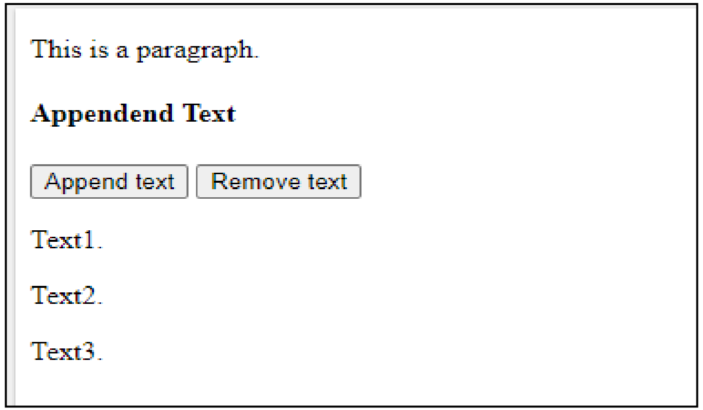
- Output



This is a paragraph.

Appendend Text

- After append text



This is a paragraph.

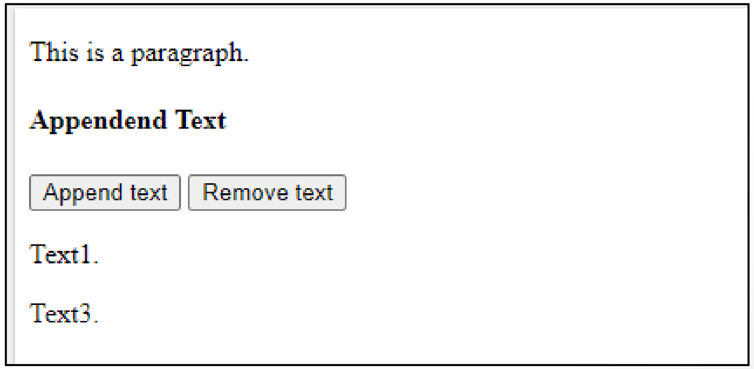
Appendend Text

Text1.

Text2.

Text3.

- After remove text



This is a paragraph.

Appendend Text

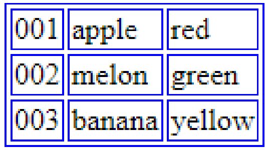
Text1.

Text3.

Dynamic table creation with jQuery and json data

```
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<style> #mytable,td{ border:1px solid blue;}</style>
<script>
$(document).ready(function ()
{
var obj=[ { item : "001", name : "apple", category : "fruit", color : "red" },
{ item : "002", name : "melon", category : "fruit", color : "green" },
{ item : "003", name : "banana", category : "fruit", color : "yellow" } ]
var tbl=$("<table/>").attr("id","mytable"); //table creation at dynamically
$("#div1").append(tbl);
for(var i=0;i<obj.length;i++) { var tr="<tr>";
var td1="<td>"+obj[i]["item"]+"</td>";
var td2="<td>"+obj[i]["name"]+"</td>";
var td3="<td>"+obj[i]["color"]+"</td></tr>";
$("#mytable").append(tr+td1+td2+td3); }})
</script>
</head>
<body> <div id="div1"> </div> </body>
</html>
```

output



001	apple	red
002	melon	green
003	banana	yellow

Exercise

1.Use a jQuery method to hide the <p> element when it is clicked on.

2. Use a jQuery method to hide the <p> element when it is clicked on. The speed should be "slow".

3. There is a hidden <p> element in the document. Use a jQuery method to show the <p> element with a click of a button.

4. Toggle between hiding and showing the <p> element when you click on the "Toggle" button.

5. Use a jQuery method to slide up the <div> element.The duration of the effect should be "slow".

Exercise –Cont.

1. Develop a quiz application using Jquery methods. The application should be developed in such a way that after the user attempted a quiz question, the answer along with the explanation should be displayed when the user clicks the answer button. Use sliding methods to develop the same
2. Develop a animated application using Jquery, the application should be developed in such a way that the user pressed the button, the object along with text should be move from left to right, right to left , top to bottom and bottom to top with particular set of intervals after completing the task it should be hide.

Exercise –cont.

3.Use jQuery selectors to identify elements with these properties in a hypothetical page:

- All p tags that have no children, but only if they don't have a class of ignore
- Any element with the text "REPLACE_ME" in it.
- All div tags with a child that has a class of special
- All heading elements (h1, h2, h3, h4, h5, h6)
- Every other visible li. Use the DOM API to target the #square and periodically change it's position in a random direction. Use jQuery selectors instead of the DOM API.