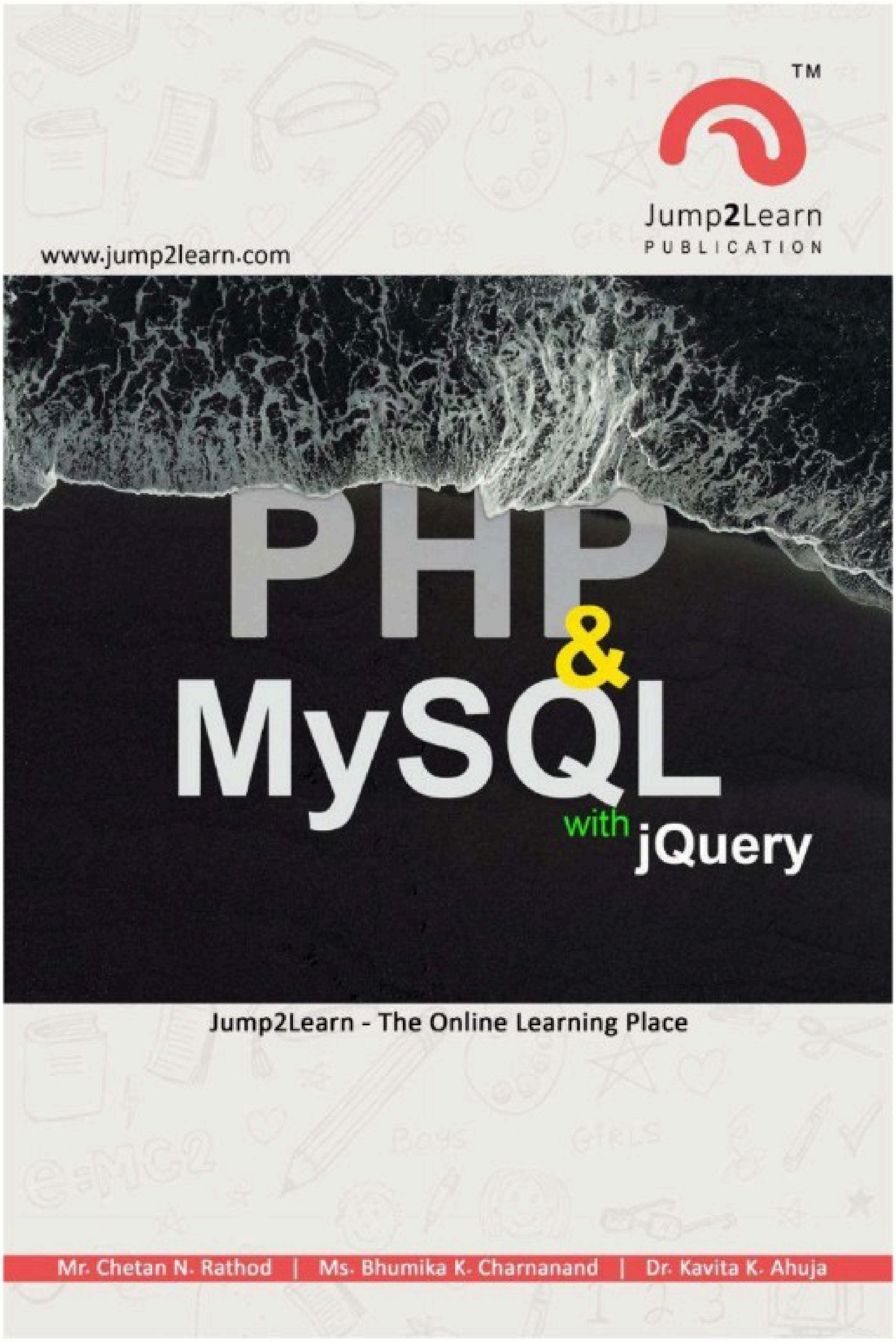
PROCEDURE
UPGRADE
SECURITY
VERIFICATION
DEPLOYMENT
EXECUTION
DEVELOPER
APPLICATION
ALGORITHM
ENGINEERING
FRAMEWORK
COMMAND
MYSQL
INFORMATION
FUNCTION
OPTIMIZATION
DATA
FEATURE
INTERPRETER
MAINTENANCE
REQUIREMENTS
METHODOLOGY
DOCUMENTATION
PROGRAM
SOURCE
DEVELOPMENT
PHP
BUG
PROGRAMMER
FIRMWARE
SQL
STEP
COMPUTER
DEBUGGER
SYSTEM
PROTOTYPING
CLIENT
CODING
SOFTWARE
INSTRUCTION
PROGRAMMING
EMBEDDED
COMPUTING
COMPILER
INSTALLATION
APPROACH
DATABASE
SOFTWARE
UPDATE
PLATFORM



TM

Jump2Learn PUBLICATION

www.jump2learn.com

PHP & MySQL with jQuery

Jump2Learn - The Online Learning Place

Mr. Chetan N. Rathod | Ms. Bhumika K. Charnanand | Dr. Kavita K. Ahuja

Unit - 5
jQuery

jQuery is the most useful and popular JavaScript library created by John Resig in the year 2006. It is an open source JavaScript library to achieve simplification between HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) document. jQuery is a cross-platform JavaScript library designed to simplify the client-side scripting of HTML.

Elaborating the terms, jQuery simplifies HTML document traversing and manipulation, browser event handling, DOM animations, Ajax interactions, and cross-browser JavaScript development. jQuery helps in loading web pages faster and it is SEO friendly. jQuery is widely famous with its philosophy of "Write less, do more".

The jQuery library contains the following features:

- HTML/DOM manipulation
- CSS manipulation
- HTML event Handling
- Effects and animations
- AJAX
- Utilities
- Lightweight

Installation of jQuery

There are two ways to install jQuery:

Local Installation:

The jQuery library can directly downloaded from their official website i.e. https://jquery.com/download/ and include it in your HTML code.

The jQuery library is a single JavaScript file, and you reference it with the HTML <script> tag (notice that the <script> tag should be inside the <head> section):

```
<head>
<script src="jquery-3.4.1.min.js"></script>
</head>
```

NOTE: Place the downloaded file in the same directory as the pages where you wish to use it.

Link to a CDN:

Add jQuery library in your HTML code from Content Delivery Network.

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.To use jQuery from Google or Microsoft, use one of the following:

Google CDN:-

```
<head>
        <script src="
        https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
        </script>
<head>
```

jQuery Syntax

The jQuery syntax is tailor-made for **selecting** HTML elements and performing some **action** on the element(s).

Syntax:

```
$(selector).action()
```

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)The Document Ready Event
- All jQuery methods need to be use inside a document ready event:

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

- This is to prevent any jQuery code from running before the document is finished loading (is ready).

jQuery Selectors

- **Selectors** are required in every statement using **jQuery**.
- jQuery Selectors are used to select and manipulate HTML elements by using the "$()" function.
- The point of a selector is to select the element on which to perform an action. jQuery selectors are used to "find" (or select) 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: $().

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

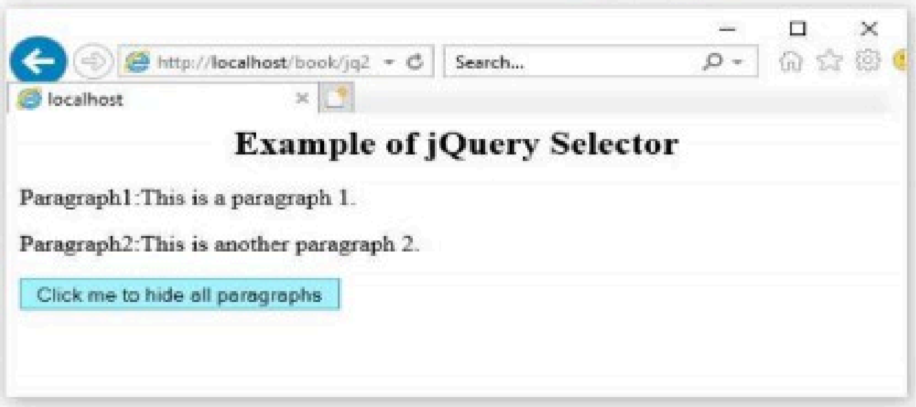
Common Selectors

$("p")	Selects all of the <p> elements
$("*")	Selects all elements
$(this)	Selects the current HTML element
$("p.intro")	Selects all <p> elements with class="intro"
$("p:first")	Selects the first <p> element
$("ul li:first")	Selects the first <li> element of the first <ul>
$("[href]")	Selects all elements with an href attribute
$("a[target='_blank']")	Selects all <a> elements with a target attribute value equal to "_blank"

$(":button")	Selects all <button> elements and <input> elements of type="button"
$("tr:even")	Selects all even <tr> elements
$("tr:odd")	Selects all odd <tr> elements

Example:

```
<?php
        echo"<center><h2> Example of jQuery Selector </h2></center>";
?>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
         $("button").click(function(){
         $("p").hide();
         });
        });
</script>
</head>
<body>
        <p>Paragraph1:This is a paragraph 1.</p>
        <p>Paragraph2:This is another paragraph 2.</p>
        <button>Click me to hide all paragraphs</button>
</body>
```



Example of jQuery Selector

Paragraph1:This is a paragraph 1.

Paragraph2:This is another paragraph 2.

Click me to hide all paragraphs

jQuery Events

jQuery is tailor-made to respond to events in an HTML page. All the different visitors' actions that a web page can respond to are called events. An event represents the precise moment when something happens.

Events Examples:

- moving a mouse over an element
- selecting a radio button
- clicking on an element

Here are some common DOM events:

Mouse Events	Keyboard Events	Form Events	Document Events
Click	Keypress	Submit	Load
Dbclick	Keydown	Change	Resize
Mouseenter	Keyup	Focus	Scroll
Mouseleave		Blur	Unload

jQuery Event- click()

When you click on an element, the click event occurs by executing a function or a set of statements. The click() method attaches an event handler function to an HTML element.

Example

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

Here, a click event is assign to all the paragraphs on a page, you can do this:

```
$("p").click();
```

The next step is to define what should happen when the event fires. You must pass a function to the event:

```
$("p").click(function(){
        // action goes here!!
});
```

In our example, the paragraph click on will hide on the page.

```
$(this).hide();
```

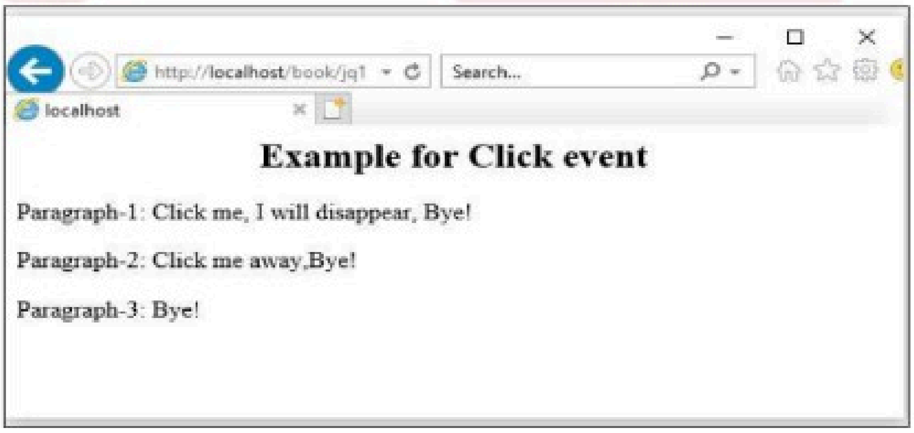
Commonly Used jQuery Event Methods-

$(document).ready()

The $(document).ready() method allows us to execute a function when the document is fully loaded.

Example:

```
<?php
        echo "<center><h2>Example for Click event</h2></center>";
?>
<form name="f1">
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
        $(document).ready(function(){
                $("p").click(function(){
                                $(this).hide();
                });
        });
</script>
</head>
<body>
        <p>Paragraph-1: Click me, I will disappear, Bye!</p>
        <p>Paragraph-2: Click me away,Bye!</p>
        <p>Paragraph-3: Bye!</p>
</body>
</form>
```



Example for Click event

Paragraph-1: Click me, I will disappear, Bye!

Paragraph-2: Click me away,Bye!

Paragraph-3: Bye!

jQuery Event- mouseenter()

The mouseenter() method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer enters the HTML element:

```
$("#p1").mouseenter(function(){
                alert("You entered p1!"); });
```

jQuery Event- mouseleave()

The mouseleave() method attaches an event handler function to an HTML element.

The function is executed when the mouse pointer leaves the HTML element:

```
$("#p1").mouseleave(function(){
        alert("Bye! Your Mouse leave from p1!");
});
```

jQuery Event- mousedown()

The mousedown() method attaches an event handler function to an HTML element. The function is executed, when the left, middle or right mouse button is pressed down, while the mouse is over the HTML element:

```
$("#p1").mousedown(function(){
        alert("Your Mouse down over p1!");
});
```

jQuery Event- hover()

The hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.
The first function is executed when the mouse enters the HTML element, and the second function is executed when the mouse leaves the HTML element:

```
            $("#p1").hover(function(){
            alert("You entered p1!");
    },
            function(){
                    alert("Bye! You now leave p1!");
            });
```

jQuery Event- focus()

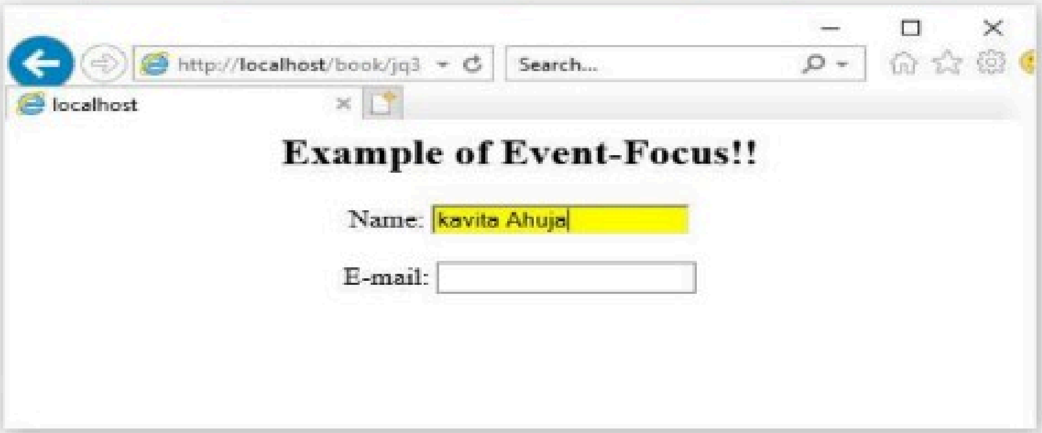
The focus() method attaches an event handler function to an HTML form field.
The function is executed when the form field gets focus:

```
$("input").focus(function()
{
        $(this).css("background-color", "#cccccc");
 });
```

Example:

```
<?php

        echo "<center><h2> Example of Event-Focus!! <center></h2>";

?>
```

```
<html>
<head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
        </script>
<script>
                $(document).ready(function()
                  {
                          $("input").focus(function()
                           {
                            $(this).css("background-color", "yellow");
                          });
                });
</script>
</head>
<body>
        Name: <input type="text" name="Username:"><br>
        <br>
        E-mail: <input type="text" name="E-mail:">
</body>
</html>
```



Example of Event-Focus!!

Name: kavita Ahuja

E-mail:

jQuery Event- blur()

The blur() method attaches an event handler function to an HTML form field.

The function is executed when the form field loses focus:

```
$("input").blur(function(){
$(this).css("background-color", "#ffffff");
});
```

jQuery Event- The on() Method

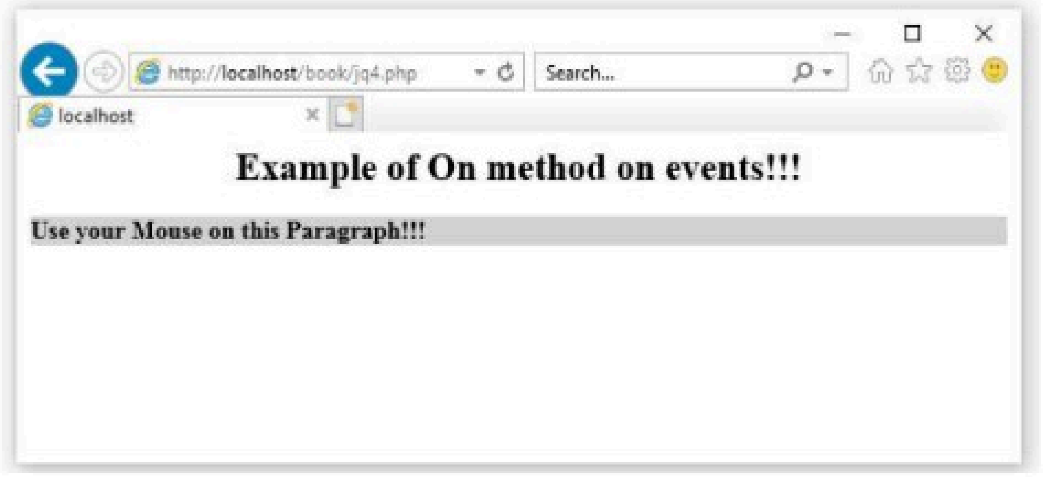
The on() method attaches one or more event handlers for the selected elements.

Attach a click event to a <p> element:

```
$("p").on("click", function(){
        $(this).hide();
});
```

Attach multiple event handlers to a <p> element:
Example:

```
<?php
        echo "<center><h2> Example of On method on events!!!</center></h2>";
?>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
        $(document).ready(function(){
                 $("p").on({
                  mouseenter: function(){
                        $(this).css("background-color", "lightgray");
                },
                  mouseleave: function()
                {
                   $(this).css("background-color", "blue");
                },
                  click: function(){
                         $(this).css("background-color", "yellow");
                }
                });
        });
</script>
</head>
<body>
                <p><b>Use your Mouse on this Paragraph!!!</b></p>
</body>
        </html>
```

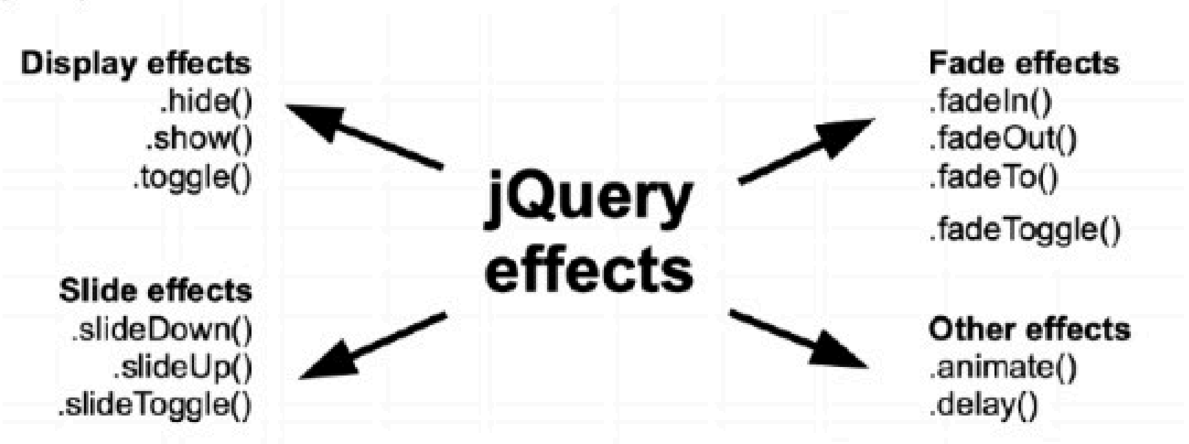


Example of On method on events!!!

Use your Mouse on this Paragraph!!!

jQuery Effects

jQuery provide features of adding effects on a web page. There are four categories of jQuery effects:



- jQuery Hide
- jQuery Show
- jQuery Fade
- jQuery Slide
- jQuery Animate
- jQuery stop
- jQuery Callback
- jQuery Chaining

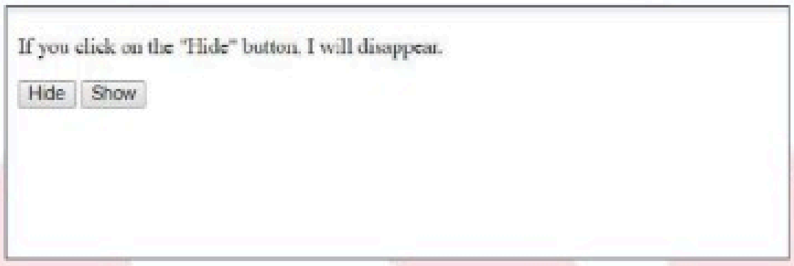
jQuery hide() and show()

With jQuery, you can hide and show HTML elements with the hide() and show() methods:

Example:

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

Output:



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

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
                $(document).ready(function(){
                $("#hide").click(function(){
                          $("p").hide();
                  });
                         $("#show").click(function(){
                         $("p").show();
                  });
        });
</script>
</head>

        <body>

                <p>If you click on the "Hide" button, I will disappear.</p>

                <button id="hide">Hide</button>

                <button id="show">Show</button>

        </body></html>
```

Syntax:

```
$(selector).hide(speed,callback);
$(selector).show(speed,callback);
```

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

The following example demonstrates the speed parameter with hide():

Example

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

jQuery Fading Methods

With jQuery you can fade an element in and out of visibility.

jQuery has the following fade methods:

- fadeIn()
- fadeOut()
- fadeToggle()
- fadeTo()

jQuery fadeIn() Method

The jQuery fadeIn() method is used to fade in a hidden element. fadeIn() is a method to create the fading effect by changing the opacity of the selected element to bring the visibility of the element from a hidden state to a visible state.

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.

The following example demonstrates the fadeIn() method with different parameters:

Example

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

jQuery fadeOut() Method

The jQuery fadeOut() method is used to fade out a visible element.

Syntax:

$(*selector*).fadeOut(*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.

The following example demonstrates the fadeOut() method with different parameters:

Example:

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

jQuery fadeToggle() Method

The jQuery fadeToggle() method toggles between the fadeIn() and fadeOut() methods. If the elements are faded out, fadeToggle() will fade them in. If the elements are faded in, fadeToggle() will fade them out.

Syntax:

$(*selector*).fadeToggle(*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.

The following example demonstrates the fadeToggle() method with different parameters:

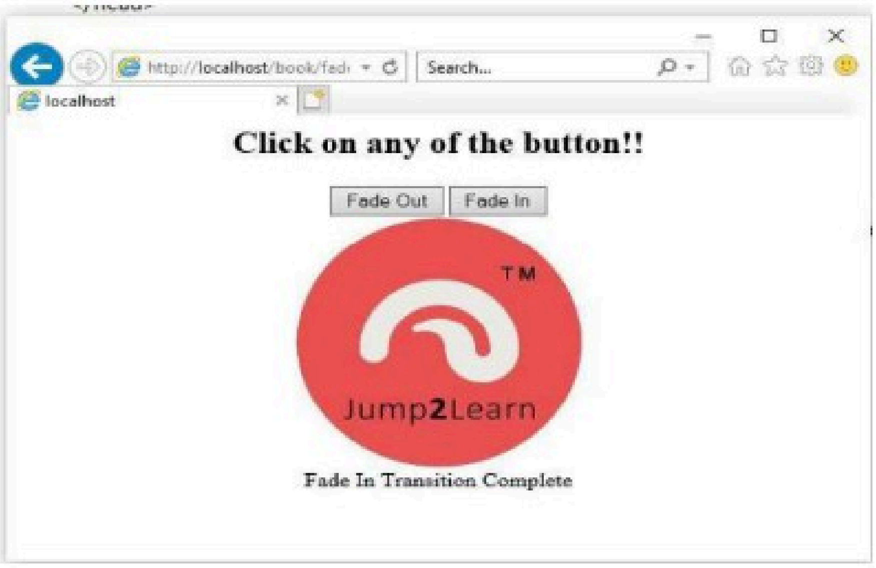
Example:

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

Example:

```
                    <?php
                            echo "<center><h2> Click on any of the
button!!</h2><center>";
                    ?>
<head>
                    <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
    $(document).ready(function() {
    $("#in").click(function(){
    $(".target").fadeIn( 'slow', function(){
                                    $(".log").text('Fade In Transition Complete');
      });
    });
    $("#out").click(function(){
      $(".target").fadeOut( 'slow', function(){
        $(".log").text('Fade Out Transition Complete');
      });
    });
  });
                <script>
            </head>
  <body>
        <button id = "out"> Fade Out </button>
                        <button id = "in"> Fade In</button>
    <div class = "target">
      <img src = "C:\wamp\www\book\images\logo.jpg" height=200px width=200px
alt = "jQuery" />
    </div>
```

```
                    <div class = "log"></div>
  </body>
```



Click on any of the button!!

Fade In Transition Complete

jQuery fadeTo() Method
The jQuery fadeTo() method allows fading to a given opacity (value between 0 and 1).
Syntax:

```
                                $(selector).fadeTo(speed,opacity,callback);
```

The required speed parameter specifies the duration of the effect. It can take the following values: "slow", "fast", or milliseconds. The required opacity parameter in the fadeTo() method specifies fading to a given opacity (value between 0 and 1). The optional callback parameter is a function to be executed after the function completes.
The following example demonstrates the fadeTo() method with different parameters:

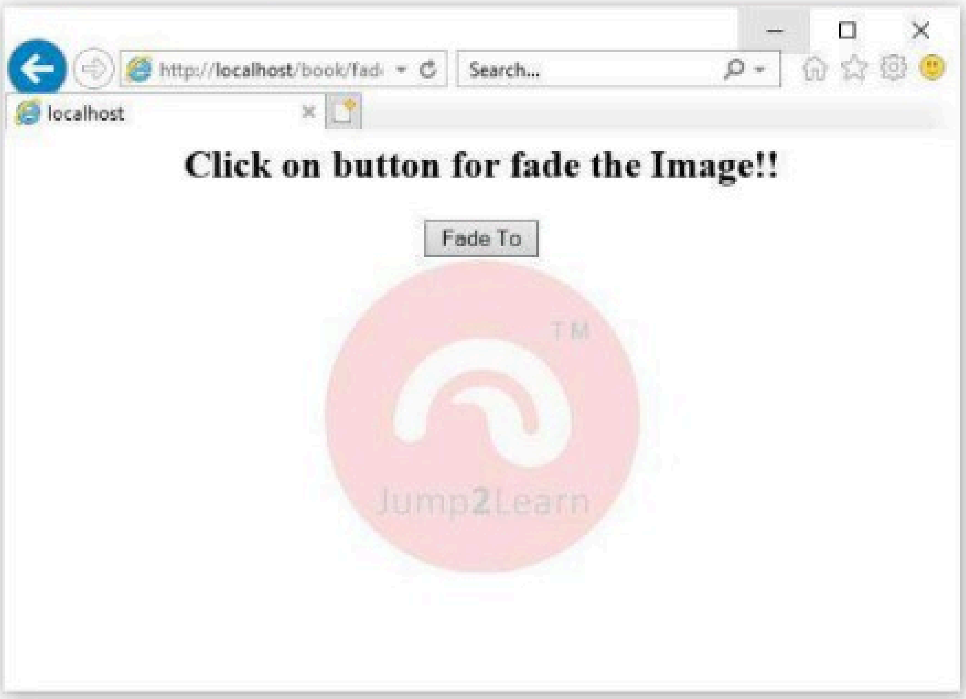
Example:

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

Example:

```
<?php
        echo "<center><h2> Click on button for fade the Image!!</h2><center>";
?>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">

              </script>

              <script>
```

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

                        $("#to").click(function(){
          $(".target").fadeTo( 'slow', 0.20);
                   });
      });
  </script>
  </head>
  <body>
    <button id = "to"> Fade To</button>
        <div class = "target">
 <img src = "C:\wamp\www\book\images\logo.jpg" height=200px width=200px alt =
"jQuery" />
    </div>
    <div class = "log"></div>
  </body>
```


http://localhost/book/fad
Search...
localhost
Click on button for fade the Image!!
Fade To
TM
Jump2Learn

jQuery Sliding Methods- Slide

With jQuery you can create a sliding effect on elements.

jQuery has the following slide methods:

- **slideDown()**
- **slideUp()**

- **slideToggle()**

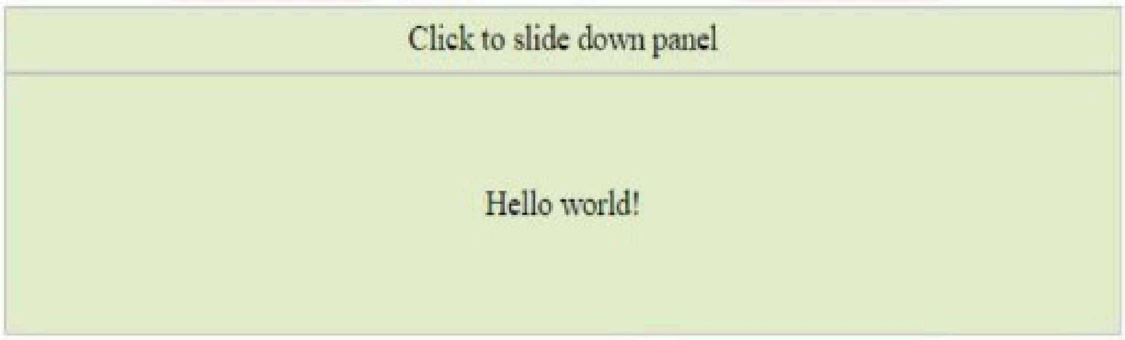
jQuery slideDown() Method

The jQuery slideDown() method is used to slide down an element.

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.



```
<html>
<head>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">
</script>
<script>
      $(document).ready(function(){
        $("#flip").click(function(){
        $("#panel").slideDown("slow");
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  text-align: center;
  background-color: #e5eecc;
  border: solid 1px #c3c3c3;
}


                  #panel
                   {
                    padding: 50px;
                    display: none;
                  }
                  </style>
                  </head>
                  <body>


<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>

</body>
</html>
```

jQuery slideUp() Method

The jQuery slideUp() method is used to slide up an element.

Syntax:

$(selector).slideUp(speed,callback);

jQuery slideToggle() Method

The jQuery slideToggle() method toggles between the slideDown() and slideUp() methods. If the elements have been slid down, slideToggle() will slide them up. If the elements have been slid up, slideToggle() will slide them down.

Syntax:

$(selector).slideToggle(speed,callback);

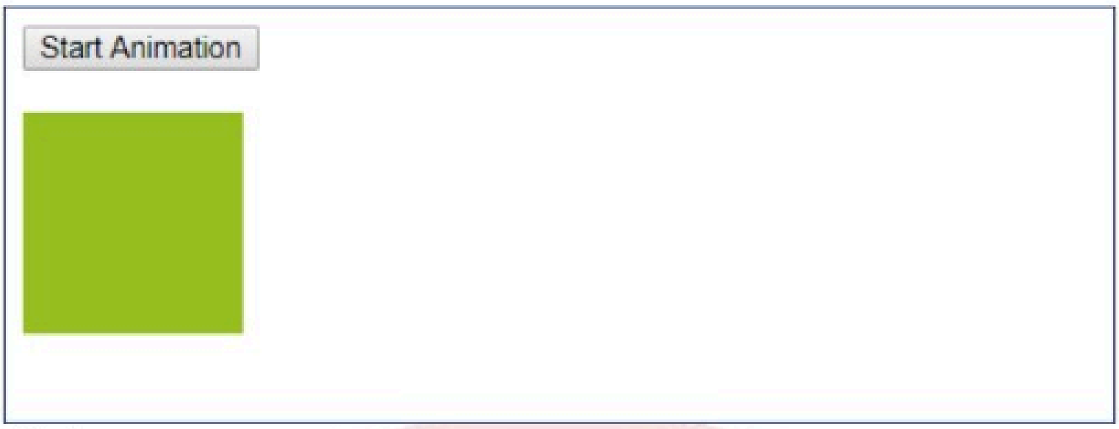
jQuery Effects - Animation

With jQuery, you can create custom animations. jQuery Animations - The animate() Method. The jQuery animate() method is used to create custom animations.

Syntax:

$(*selector*).animate({*params*},*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 animation completes. The following example demonstrates a simple use of the animate() method; it moves a <div> element to the right, until it has reached a left property of 250px:



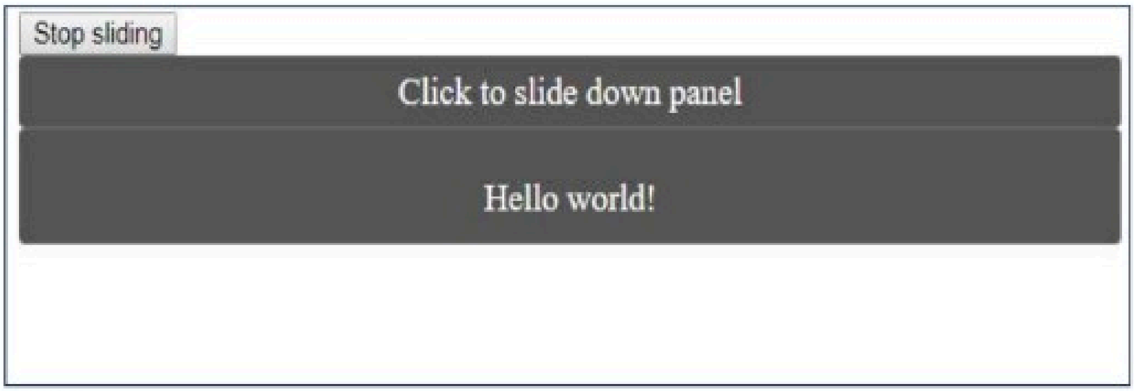
```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("div").animate({left: '250px'});
  });
});
</script>
</head>
<body>
<button>Start Animation</button>
<br><br>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

jQuery Stop Animations

The jQuery stop() method is used to stop an animation or effect before it is finished. The stop() method works for all jQuery effect functions, including sliding, fading and custom animations.

Syntax:

```
$(selector).stop(stopAll,goToEnd);
```



```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown(5000);
  });
  $("#stop").click(function(){
    $("#panel").stop();
  }); });
</script>
<style>
#panel, #flip {
  padding: 5px;
  font-size: 18px;
  text-align: center;
  background-color: #555;
  color: white;
  border: solid 1px #666;
  border-radius: 3px;
}
#panel {
  padding: 50px;
  display: none;

            }

            </style>

            </head>

            <body>
```

```
        <button id="stop">Stop sliding</button>

        <div id="flip">Click to slide down panel</div>

        <div id="panel">Hello world!</div>

        </body>

        </html>
```

jQuery Callback Functions

JavaScript statements are executed line by line. However, with effects, the next line of code can be run even though the effect is not finished. This can create errors. To prevent this, you can create a callback function. A callback function is executed after the current effect is finished.

Syntax: $(*selector*).hide(*speed,callback*);

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide("slow", function(){
      alert("The paragraph is now hidden");
    });
  });
});
</script>
</head>
<body>
<button>Hide</button>
<p>This is a paragraph with little content.</p>
</body>
</html>
```

jQuery - Chaining

With jQuery, you can chain together actions/methods. Chaining allows us to run multiple jQuery methods (on the same element) within a single statement. Until now we have been writing jQuery statements one at a time (one after the other). However, there is a technique called chaining, that allows us to run multiple jQuery commands, one after the other, on the same element(s).

Tip: This way, browsers do not have to find the same element(s) more than once.

- ✓ 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:

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "red").slideUp(2000).slideDown(2000);
  });
});
</script>
</head>
<body>
<p id="p1">jQuery is fun!!</p>
<button>Click me</button>
</body></html>
```

jQuery HTML

Get:

Three simple, but useful, jQuery methods for DOM(DOM = Document Object Model). The DOM defines a standard for accessing HTML and XML documents) 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

```
$("#btn1").click(function(){
      alert("Text: " + $("#test").text());
    });
    $("#btn2").click(function(){
      alert("HTML: " + $("#test").html());
    });
```

- **Get Attributes - val()**

The following example demonstrates how to get the value of an input field with the jQuery val() method:

Example:

```
                $("#btn1").click(function(){
                  alert("Value: " + $("#test").val());
        });
```

- **Get Attributes - attr()**

The jQuery attr() method is used to get attribute values.
The following example demonstrates how to get the value of the href attribute in a link:

Example:

```
$("button").click(function(){
        alert($("#w3s").attr("href"));
      });
```

Set:

Set Content - text(), html(), and val()
We will use the same three methods from the previous page to set content:

- 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
- The following example demonstrates how to set content with the jQuery text(), html() and val() methods:

Example:

```
                 $("#btn1").click(function(){
                           $("#test1").text("Hello world!");
                           });
                 $("#btn2").click(function(){
                   $("#test2").html("<b>Hello world!</b>");
```

```
                        });
                        $("#btn3").click(function(){
                          $("#test3").val("Dolly Duck");
                        });
```

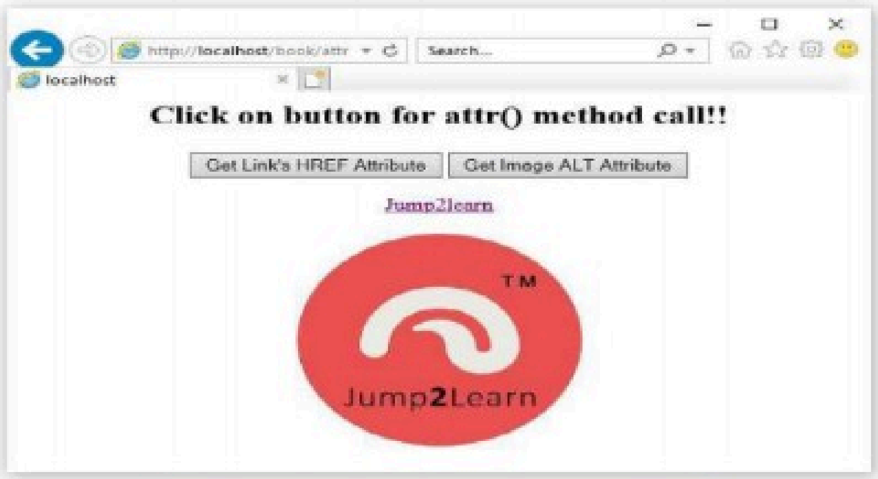
Set Attributes - attr()

The jQuery attr() method is also used to set/change attribute values.

The following example demonstrates how to change (set) the value of the href attribute in a link:

Example:

```
                    <?php
            echo "<center><h2> Click on button for attr() method call!!</h2>
<center>";
                    ?>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
     $(".btn-one").click(function(){
            var str = $("a").attr("href");
        alert(str);
  });
        $(".btn-two").click(function(){
        var str = $("img#sky").attr("alt");
              alert(str);
  });
});
   </script>
  </head>
  <body>
   <button type="button" class="btn-one">Get Link's HREF Attribute</button>
   <button type="button" class="btn-two">Get Image ALT Attribute</button>
   <p><a href="https://jump2learn.com/">Jump2learn</a></p>
   <img height=200px width=200px id="sky"
src="C:\wamp\www\book\images\logo.jpg" alt="jquery">
</body>
```



Click on button for attr() method call!!

Get Link's HREF Attribute | Get Image ALT Attribute

Jump2learn

jQuery - Add Elements

Add New HTML Content

There are four jQuery methods that are used to add new content:

- **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

jQuery append() Method

The jQuery append() method inserts content AT THE END of the selected HTML elements.

Example:

```
$("p").append("Some appended text.");
```

jQuery prepend() Method

The jQuery prepend() method inserts content AT THE BEGINNING of the selected HTML elements.

Example:

```
$("p").prepend("Some prepended text.");
```

jQuery after() and before() Methods

The jQuery after() method inserts content AFTER the selected HTML elements.

The jQuery before() method inserts content BEFORE the selected HTML elements.

Example:

```
$("img").after("Some text after");
$("img").before("Some text before");
```

jQuery - Remove Elements

With jQuery, it is easy to remove existing HTML elements.

Remove Elements/Content

To remove elements and content, there are mainly two jQuery methods:

- remove()- Removes the selected element (and its child elements)
- empty()- Removes the child elements from the selected element

jQuery remove() Method

The jQuery remove() method removes the selected element(s) and its child elements.

Example:

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

jQuery empty() Method

The jQuery empty() method removes the child elements of the selected element(s).

Example:

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

jQuery - Get and Set CSS Classes

jQuery has several methods for CSS manipulation. The CSS class methods are-

- addClass()- Adds one or more classes to the selected elements
- removeClass() - Removes one or more classes from the selected elements
- toggleClass() - Toggles between adding/removing classes from the selected elements
- css() - Sets or returns the style attribute

Example:

The following stylesheet will be used for all the examples:

```
.important {
        font-weight: bold;
        font-size: xx-large;
    }

.blue {
        color: blue;
  }
```

jQuery addClass() Method

The following example shows how to add class attributes to different elements. Of course you can select multiple elements, when adding classes:

Example:

```
$("button").click(function(){
        $("h1, h2, p").addClass("blue");
        $("div").addClass("important");
});
```

jQuery removeClass() Method

The following example shows how to remove a specific class attribute from different elements:

```
        $("button").click(function()
    {
         $("h1, h2, p").removeClass("blue");
   });
```

jQuery toggleClass() Method

The following example will show how to use the jQuery toggleClass() method. This method toggles between adding/removing classes from the selected elements:

Example:

```
 $("button").click(function()
{
        $("h1, h2, p").toggleClass("blue");
 });
```

jQuery - css() Method

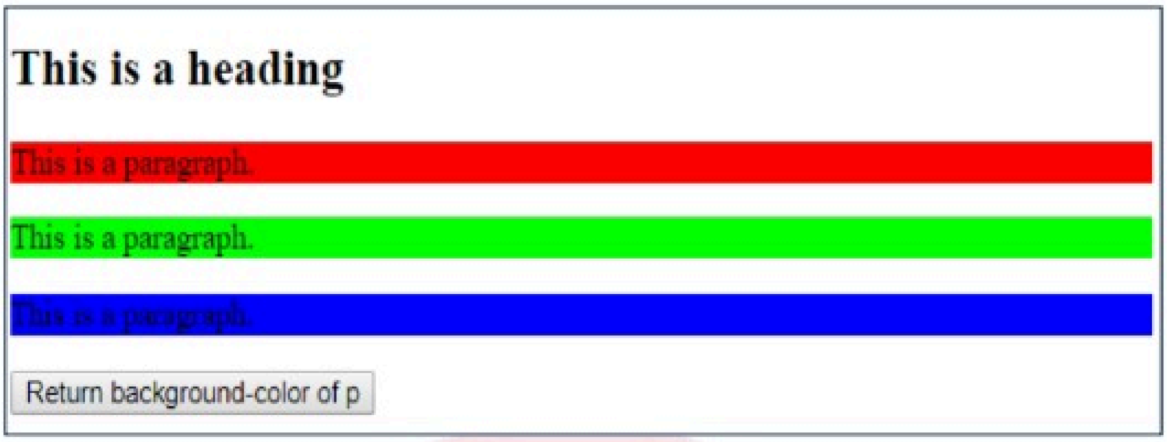
The css()method sets or returns one or more style properties for the selected elements.

Return a CSS Property

To return the value of a specified CSS property, use the following syntax:

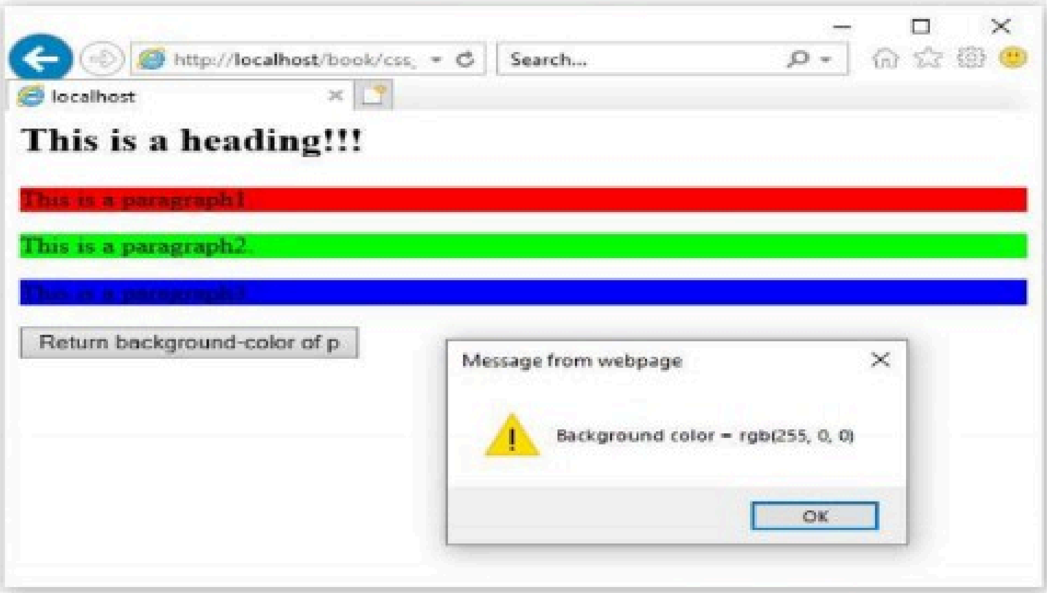
```
css("propertyname");
```

The following example will return the background-color value of the FIRST matched element:



```
<html>

<head>

<script
  src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></scri
  pt>

<script>

$(document).ready(function(){

 $("button").click(function(){

   alert("Background color = " + $("p").css("background-color"));

 });

});

</script>

</head>

<body>

<h2>This is a heading</h2>

<p style="background-color:#ff0000">This is a paragraph.</p>
```

```
<p style="background-color:#00ff00">This is a paragraph.</p>

<p style="background-color:#0000ff">This is a paragraph.</p>

<button>Return background-color of p</button>

</body>

</html>
```



This is a heading!!!

This is a paragraph1.

This is a paragraph2.

This is a paragraph3.

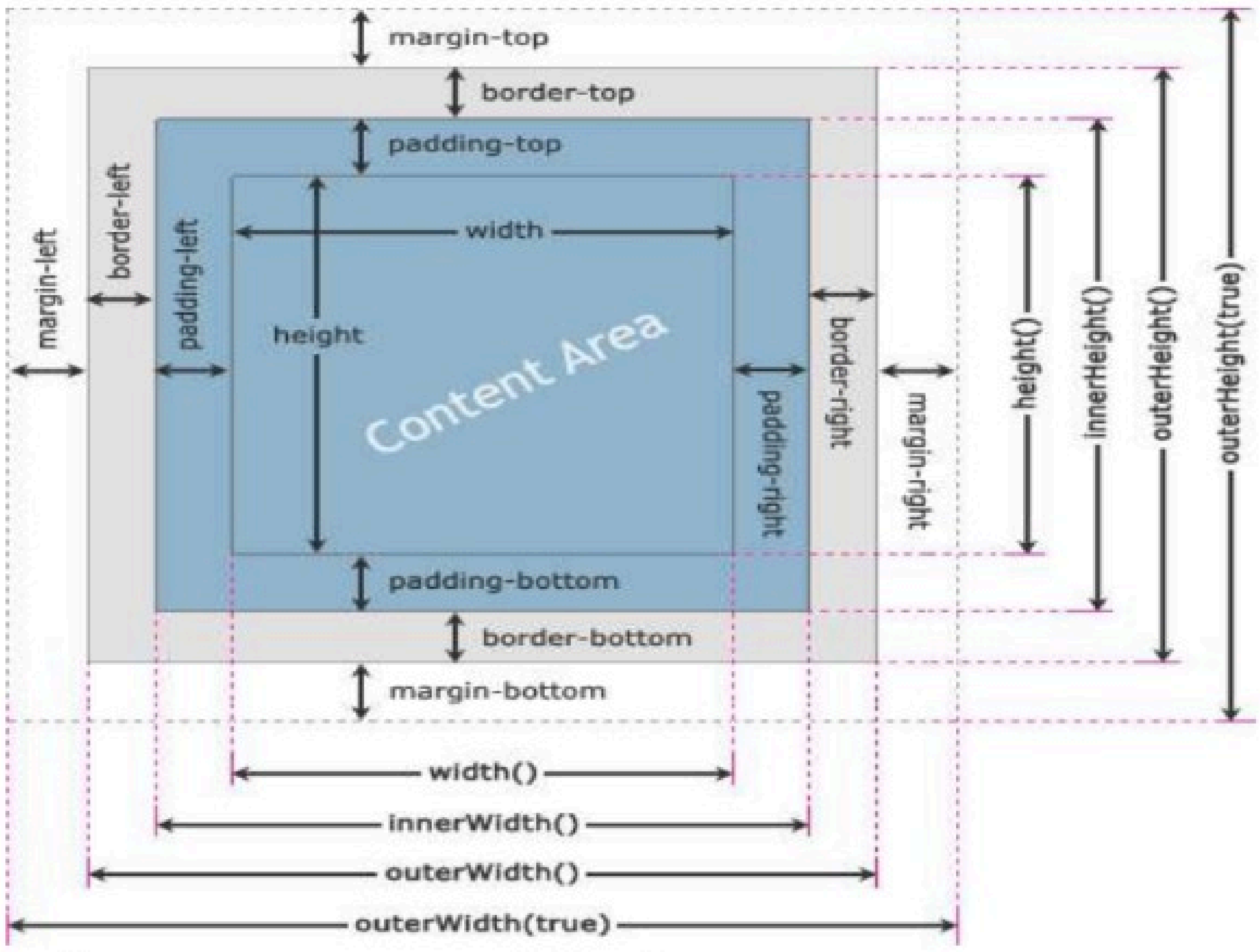
Background color = rgb(255, 0, 0)

jQuery - Dimensions

With jQuery, it is easy to work with the dimensions of elements and browser window. jQuery has several important methods for working with dimensions:

- width()
- height()
- innerWidth()
- innerHeight()
- outerWidth()
- outerHeight()

jQuery Dimensions



jQuery width() and height() Methods

The width() method sets or returns the width of an element (excludes padding, border and margin). The height() method sets or returns the height of an element (excludes padding, border and margin). The following example returns the width and height of a specified <div> element:

Example:

```
$("button").click(function(){
        var txt = "";
        txt += "Width: " + $("#div1").width() + "</br>";
```

```
                          txt += "Height: " + $("#div1").height();
                           $("#div1").html(txt);
            });
```

jQuery innerWidth() and innerHeight() Methods

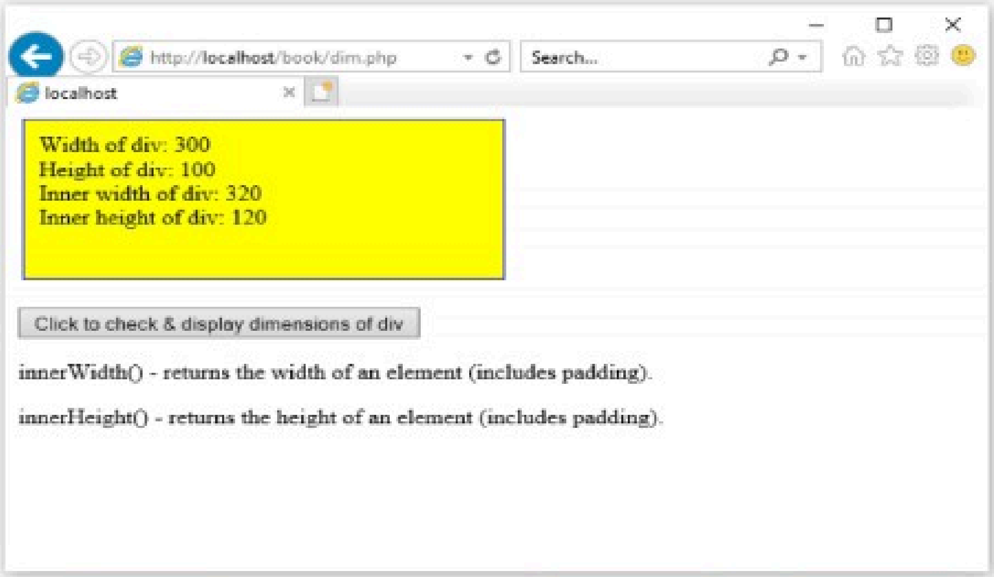
The innerWidth() method returns the width of an element (includes padding).

The innerHeight() method returns the height of an element (includes padding).
The following example returns the inner-width/height of a specified <div> element:

Example:

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
        var txt = "";
        txt += "Width of div: " + $("#div1").width() + "</br>";
        txt += "Height of div: " + $("#div1").height() + "</br>";
        txt += "Inner width of div: " + $("#div1").innerWidth() + "</br>";
        txt += "Inner height of div: " + $("#div1").innerHeight();
        $("#div1").html(txt);
  });
});
</script>
</head>
<style>
#div1 {
  height: 100px;
  width: 300px;
  padding: 10px;
  margin: 3px;
  border: 1px solid blue;
  background-color: yellow;
}
</style>
<body>
<div id="div1"></div>
            <br>

            <button> Click to check & display dimensions of div</button>
```

```
<p>innerWidth() - returns the width of an element (includes
    padding).</p>

<p>innerHeight() - returns the height of an element (includes
    padding).</p>

</body>
```


Width of div: 300
Height of div: 100
Inner width of div: 320
Inner height of div: 120

Click to check & display dimensions of div

innerWidth() - returns the width of an element (includes padding).

innerHeight() - returns the height of an element (includes padding).

jQuery outerWidth() and outerHeight() Methods

The outerWidth() method returns the width of an element (includes padding and border).

The outerHeight() method returns the height of an element (includes padding and border).

The following example returns the outer-width/height of a specified <div> element:

Example:

```
$("button").click(function(){
  var txt = "";
  txt += "Outer width: " + $("#div1").outerWidth() + "</br>";
  txt += "Outer height: " + $("#div1").outerHeight();
  $("#div1").html(txt);
});
```

The outerWidth(true) method returns the width of an element (includes padding, border, and margin).

The outerHeight(true) method returns the height of an element (includes padding, border, and margin).

Example:

```
$("button").click(function(){
var txt = "";
txt += "Outer width (+margin): " + $("#div1").outerWidth(true) + "</br>";
txt += "Outer height (+margin): " + $("#div1").outerHeight(true);
            $("#div1").html(txt);
});
```

jQuery More width() and height()

The following example returns the width and height of the document (the HTML document) and window (the browser viewport):
Example:

```
$("button").click(function(){
var txt = "";
txt += "Document width/height: " + $(document).width();
txt += "x" + $(document).height() + "\n";
txt += "Window width/height: " + $(window).width();
txt += "x" + $(window).height();
alert(txt);
});
```

The following example sets the width and height of a specified <div> element:
Example:

```
$("button").click(function(){
 $("#div1").width(500).height(500);
});
```

jQuery Traversing

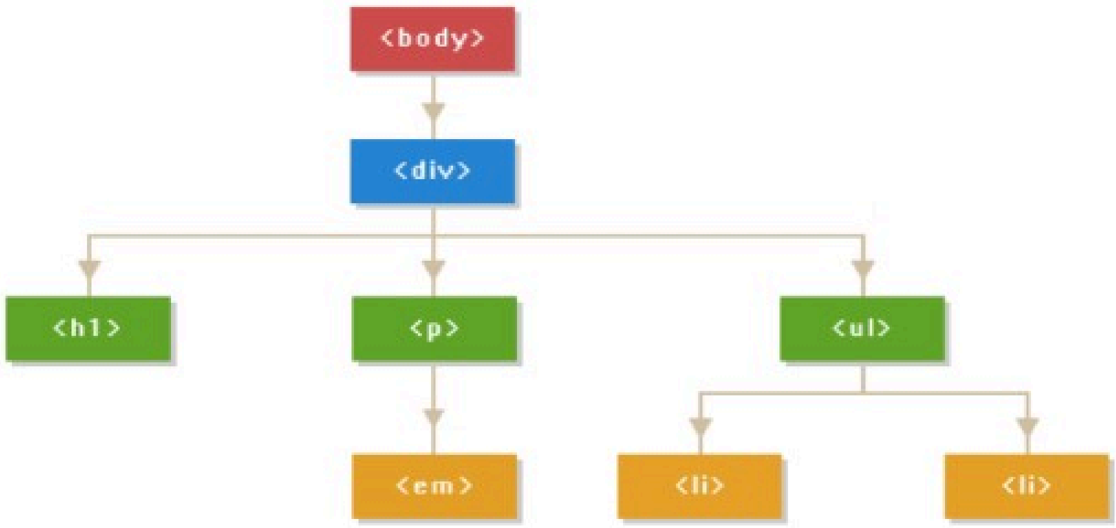
jQuery traversing, which means "move through", are used to "find" (or select) HTML elements based on their relation to other elements. Start with one selection and move through that selection until you reach the elements you desire.
The Document Object Model (DOM) is a standard convention for accessing and manipulating elements within HTML and XML documents. Elements in the DOM are organized into a tree-like data structure that can be traversed to navigate, locate, or modify elements and/or content within an XML/HTML document.
DOM tree traversal may be accomplished through the use of six basic properties. All properties, except childNodes, contain a reference to another node object. The childNodes property contains a reference to an array of nodes-

- previousSibling
- nextSibling
- childNodes
- firstChild
- lastChild
- parentNode

Given the following HTML:

```
<head>
<meta charset="utf-8">
<title>HTML DOM Tree Sample</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
</head>
<body>
   <div class="container">
                        <h1>Hello World</h1>
                        <p>This is a <em>simple paragraph</em>.</p>
          <ul>
                            <li>Item One</li>
                           <li>Item Two</li>
          </ul>
   </div>
</body>
</html>
```

The HTML code in the above example can be represented as below Tree:



The above diagram showing the parent/child relationships between the elements:

- The <body> element is the **parent** of the <div> element, and an ancestor of everything inside of it. The enclosed <div> element is the parent of <h1>, <p> and <ul> elements, and a child of the <body> element.
- The elements <h1>, <p> and <ul> are **siblings,** since they share the same parent.
- The <h1> element is a **child** of the <div> element and a **descendant** of the <body> element. This element does not have any children.
- The <p> element is the **parent** of <em> element, child of the <div> element and a descendant of the <body> element. The containing <em> element is a child of this <p> element and a descendant of the <div> and <body> element.
- Similarly, the <ul> element is the **parent** of the <li> elements, child of the <div> element and a descendant of the <body> element.
- The containing <li> elements are the **child** of this <ul> element and a **descendant** of the <div> and <body> element. Also, both the <li> elements are **siblings.**

jQuery Traversing - Ancestors

- With jQuery you can traverse up the DOM tree to find ancestors of an element.
- An ancestor is a parent, grandparent, great-grandparent, and so on.

- Traversing Up the DOM Tree

Three useful jQuery methods for traversing up the DOM tree are:

- parent()
- parents()
- parentsUntil()

jQuery parent() Method

- The parent() method returns the direct parent element of the selected element.
- This method only traverse a single level up the DOM tree.
- The following example returns the direct parent element of each <span> elements:

Example:

```
$(document).ready(function(){
        $("span").parent();
});
```

jQuery parents() Method

The parents() method returns all ancestor elements of the selected element, all the way up to the document's root element (<html>).
The following example returns all ancestors of all <span> elements:

Example:

```
$(document).ready(function(){
  $("span").parents();
});
```

You can also use an optional parameter to filter the search for ancestors.
The following example returns all ancestors of all <span> elements that are <ul> elements:

Example:

```
$(document).ready(function(){
$("span").parents("ul");
});
```

jQuery parentsUntil() Method

The parentsUntil() method returns all ancestor elements between two given arguments.

The following example returns all ancestor elements between <span> and <div> element:

Example:

```
$(document).ready(function()
   {
       $("span").parentsUntil("div");
   });
```

jQuery Traversing - Descendants

With jQuery you can traverse down the DOM tree to find descendants of an element. A descendant is a child, grandchild, great-grandchild, and so on.

Traversing Down the DOM Tree
Two useful jQuery methods for traversing down the DOM tree are:
children()
find()
jQuery children() Method
The children() method returns all direct children of the selected element.
This method only traverses a single level down the DOM tree.
The following example returns all elements that are direct children of each <div> elements.
Example:

```
        $(document).ready(function(){
   $("div").children();
   });
```

You can also use an optional parameter to filter the search for children.
The following example returns all <p> elements with the class name "first", that are direct children of <div>:
Example:

```
    $(document).ready(function(){
 $("div").children("p.first");
});
```

jQuery find() Method

The find() method returns descendant elements of the selected element, all the way down to the last descendant.

The following example returns all <span> elements that are descendants of <div>:

Example:

```
$(document).ready(function(){
        $("div").find("span");
});
```

The following example returns all descendants of <div>:

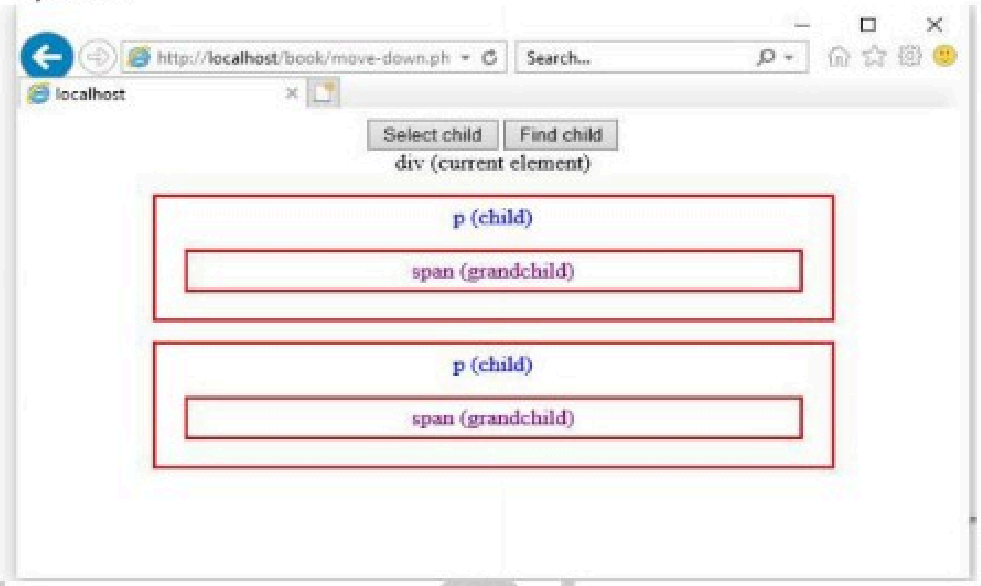
Example:

```
$(document).ready(function(){
        $("div").find("*");
});
```

Example:

```
<!DOCTYPE html>
<html>
<head>
<style>
.descendants * {
  display: block;
  border: 2px solid lightgrey;
  color: lightgrey;
  padding: 5px;
  margin: 15px;
}
</style>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#b1").click(function(){
  $("div").children().css({"color": "blue", "border": "2px solid red"});
  $("#b2").click(function(){
  $("div").find("span").css({"color": "purple", "border": "2px solid red"});
});
});
});
</script>
</head>
```

```
<body>
<center>
<input type="submit" id="b1" value="Select child">
<input type="submit" id="b2" value="Find child">
<div class="descendants" style="width:500px;">div (current element)
 <p>p (child)
   <span>span (grandchild)</span>
          </p>
          <p>p (child)
            <span>span (grandchild)</span>
          </p>
        </div>
        </center>
        </body>
        </html>
```



div (current element)

p (child)

span (grandchild)

p (child)

span (grandchild)

jQuery Traversing - Siblings

With jQuery you can traverse sideways in the DOM tree to find siblings of an element. Siblings share the same parent. Traversing Sideways in The DOM Tree

There are many useful jQuery methods for traversing sideways in the DOM tree:

- siblings()
- next()
- nextAll()
- nextUntil()
- prev()

- prevAll()
- prevUntil()

jQuery siblings() Method

The siblings() method returns all sibling elements of the selected element.

The following example returns all sibling elements of <h2>:

Example:

```
$(document).ready(function(){
        $("h2").siblings();
});
```

You can also use an optional parameter to filter the search for siblings.

The following example returns all sibling elements of <h2> that are <p> elements:

Example:

```
    $(document).ready(function(){
$("h2").siblings("p");
});
```

jQuery next() Method

The next() method returns the next sibling element of the selected element.
The following example returns the next sibling of <h2>:
Example:

```
    $(document).ready(function(){
$("h2").next();
});
```

jQuery nextAll() Method

The nextAll() method returns all next sibling elements of the selected element.
The following example returns all next sibling elements of <h2>:
Example:

```
    $(document).ready(function(){
    $("h2").nextAll();
});
```

jQuery nextUntil() Method

The nextUntil() method returns all next sibling elements between two given arguments.

The following example returns all sibling elements between a <h2> and a <h6> element:

Example:

```
(document).ready(function(){
        $("h2").nextUntil("h6");
});
```

jQuery prev(), prevAll() & prevUntil() Methods

The prev(), prevAll() and prevUntil() methods work just like the methods above but with reverse functionality: they return previous sibling elements (traverse backwards along sibling elements in the DOM tree, instead of forward).

jQuery Traversing - Filtering

The first(), last(), eq(), filter() and not() Methods used for traversing. The most basic filtering methods are first(), last() and eq() which allow you to select a specific element based on its position in a group of elements. Other filtering methods, like filter() and not() allow you to select elements that match, or do not match, a certain criteria.

jQuery first() Method

The first() method returns the first element of the specified elements.

The following example selects the first <div> element:

Example:

```
        $(document).ready(function(){
$("div").first();
});
```

jQuery last() Method

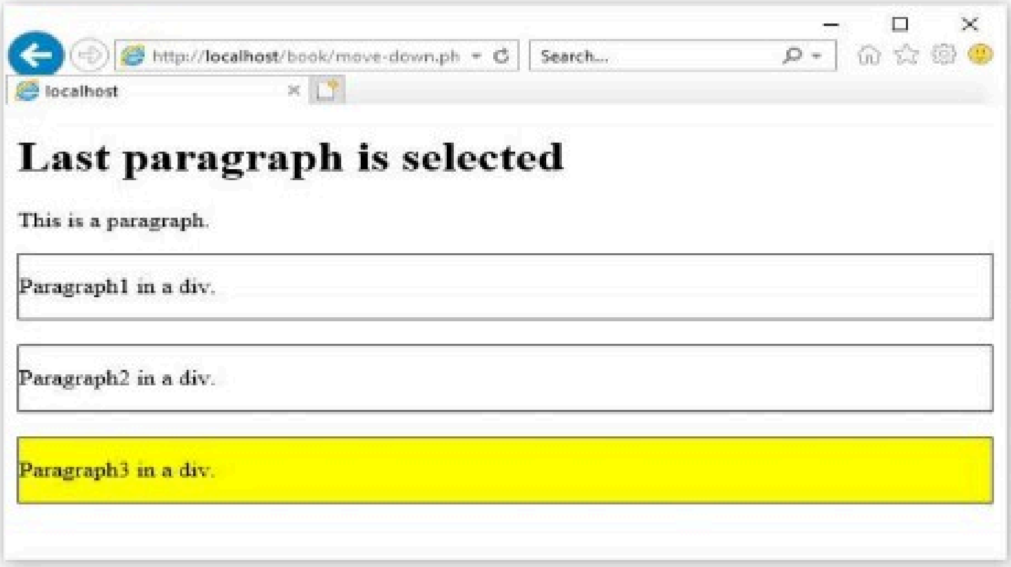
The last() method returns the last element of the specified elements.
The following example selects the last <div> element:

Example:

```
<!DOCTYPE html>

<html>
```

```
<head>

<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">

</script>

<script>

                $(document).ready(function(){

                        $("div").last().css("background-color", "yellow");

                });

</script>
</head>
<body>

                <h1> Last paragraph is selected</h1>

                <p> This is a paragraph.</p>

                <div style="border: 1px solid black;">

                        <p>Paragraph1 in a div.</p>

                 </div>

                <br>

                <div style="border: 1px solid black;">

                        <p>Paragraph2 in a div.</p>

                </div>

                <br>

                <div style="border: 1px solid black;">

                        <p>Paragraph3 in a div.</p>

                </div>
```

```
                    </body>

              </html>
```



http://localhost/book/move-down.ph

Last paragraph is selected

This is a paragraph.

Paragraph1 in a div.

Paragraph2 in a div.

Paragraph3 in a div.

jQuery eq() method

The eq()method returns an element with a specific index number of the selected elements. The index numbers start at 0, so the first element will have the index number 0 and not 1. The following example selects the second <p> element (index number 1):

Example:

```
                    $(document).ready(function(){
                            $("p").eq(1);
              });
```

jQuery filter() Method

The filter() method used to specify a criteria. Elements that do not match the criteria are removed from the selection, and those that match will be returned.
The following example returns all <p> elements with class name "intro":
Example:

```
                    $(document).ready(function(){
                      $("p").filter(".intro");
              });
```

jQuery not() Method

The not() method returns all elements that do not match the criteria. The not() method is the opposite of filter() method.

The following example returns all <p> elements that do not have class name "intro":

Example:

```
$(document).ready(function(){
        $("p").not(".intro");
});
```