

jQuery is a fast, lightweight, and feature-rich JavaScript library that is based on the principle "*write less, do more*". Its easy-to-use APIs makes the things like HTML document traversal and manipulation, event handling, adding animation effects to a web page much simpler that works seamlessly across all the major browsers like Chrome, Firefox, Safari, Internet Explorer, etc.

jQuery also gives you the ability to create an Ajax based application in a quick and simple way. Big companies like Google, Microsoft and IBM are using the jQuery for their applications. So you can easily understand how popular and powerful the jQuery is?

jQuery was originally created by Genext Resig in early 2006. The jQuery project is currently run and maintained by a distributed group of developers as an open-source project.

## What You Can Do with jQuery

There are lot more things you can do with jQuery.

- You can easily select elements to perform manipulation.
- You can easily create effect like show or hide elements, sliding transition, and so on.
- You can easily create complex CSS animation with fewer lines of code.
- You can easily manipulate DOM elements and their attributes.
- You can easily implement Ajax to enable asynchronous data exchange between client and server.
- You can easily traverse all around the DOM tree to locate any element.
- You can easily perform multiple actions on an element with a single line of code.
- You can easily get or set dimensions of the HTML elements.

## Advantages of Using jQuery

If you're not familiar with jQuery, you might be wondering what makes jQuery so special. There are several advantages why one should opt for jQuery:

- **Save lots of time** — You can save lots of time and efforts by using the jQuery inbuilt effects and selectors and concentrate on other development work.
- **Simplify common JavaScript tasks** — jQuery considerably simplifies the common JavaScript tasks. Now you can easily create feature rich and interactive web pages with fewer lines of codes, a typical example is implementing Ajax to update the content of a page without refreshing it.

- **Easy to use** — jQuery is very easy to use. Anybody with the basic working knowledge of HTML, CSS and JavaScript can start development with jQuery.
- **Compatible with browsers** — jQuery is created with modern browsers in mind and it is compatible with all major modern browsers such as Chrome, Firefox, Safari, Internet Explorer, etc.
- **Absolutely Free** — And the best part is, it is completely free to download and use.

## Downloading jQuery

<https://jquery.com/download/>

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Simple HTML Document</title>
  <link rel="stylesheet" href="css/style.css">
  <script src="js/jquery-3.5.1.min.js"></script>
</head>
<body>
  <h1>Hello, World!</h1>
</body>
</html>
```

---

## Including jQuery from CDN

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

---

# Creating Your First jQuery Powered Web Page

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="utf-8">

  <title>My First jQuery Powered Web Page</title>

  <link rel="stylesheet" href="css/style.css">

  <script src="js/jquery-3.5.1.min.js"></script>

  <script>

    $(document).ready(function(){

      $("h1").css("color", "#0088ff");

    });

  </script>

</head>

<body>

  <h1>Hello, World!</h1>

</body>

</html>
```

---

## Standard jQuery Syntax

```
<script>

$(document).ready(function(){

  // Some code to be executed...

  alert("Hello World!");

});

</script>
```

## Explanation of code

If you are completely new to the jQuery, you might think what that code was all about. OK, let's go through each of the parts of this script one by one.

- The `<script>` element — Since jQuery is just a JavaScript library, so the jQuery code can be placed inside the `<script>` element. However, if you want to place it in an external JavaScript file, which is preferred, you just remove this part.
- The `$(document).ready(handler);` — This statement is typically known as ready event. Where the *handler* is basically a function that is passed to the `ready()` method to be executed safely as soon as the document is ready to be manipulated i.e. when the DOM hierarchy has been fully constructed.

Further, inside an event handler function you can write the jQuery statements to perform any action following the basic syntax, like: `$(selector).action();`

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery Document Ready Demo</title>
  <link rel="stylesheet" href="css/style.css">
  <script src="js/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function(){
      $("p").text("Hello World!");
    });
  </script>
</head>
<body>
  <p>Not loaded yet.</p>
</body>
</html>
```

---

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
  <meta charset="utf-8">
  <title>jQuery Click Handler Demo</title>
  <link rel="stylesheet" href="css/style.css">
  <script src="js/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function(){
      $("button").click(function(){
        $("p").text("Hello World!");
      });
    });
  </script>
</head>
<body>
  <p>Not loaded yet.</p>
  <button type="button">Replace Text</button>
</body>
</html>
```

---

# jQuery Selectors

## Selecting Elements with jQuery

JavaScript is most commonly used to get or modify the content or value of the HTML elements on the page, as well as to apply some effects like show, hide, animations etc. But, before you can perform any action you need to find or select the target HTML element.

Selecting the elements through a typical JavaScript approach could be very painful, but the jQuery works like a magic here. The ability of making the DOM elements selection simple and easy is one of the most powerful feature of the jQuery.

## Selecting Elements by ID

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Select Element by ID</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    // Highlight element with id mark
    $("#mark").css("background", "yellow");
});
</script>
</head>
<body>
    <p id="mark">This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <p>This is one more paragraph.</p>
    <p><strong>Note:</strong> The value of the id attribute must be unique in an
HTML document.</p>
```

```
</body>
```

```
</html>
```

---

## Selecting Elements by Class Name

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>jQuery Select Element by Class</title>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<script>
```

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

```
    // Highlight elements with class mark
```

```
    $(".mark").css("background", "yellow");
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<p class="mark">This is a paragraph.</p>
```

```
<p class="mark">This is another paragraph.</p>
```

```
<p>This is one more paragraph.</p>
```

```
</body>
```

```
</html>
```

---

## Selecting Elements by Name

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>jQuery Select Element by Name</title>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    // Highlight paragraph elements
    $("p").css("background", "yellow");
});
</script>
</head>
<body>
    <h1>This is heading</h1>
    <p>This is a paragraph.</p>
    <p>This is another paragraph.</p>
    <div>This is another block of text.</div>
</body>
</html>
```

---

## Selecting Elements by Attribute

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Select Element by Attribute</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    // Highlight paragraph elements
    $('input[type="text"]').css("background", "yellow");
});
</script>
</head>
```



```
<body>
  <form>
    <label>Name: <input type="text"></label>
    <label>Password: <input type="password"></label>
    <input type="submit" value="Sign In">
  </form>
</body>
</html>
```

---

## Selecting Elements by Compound CSS Selector

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Select Element by Compound Selector</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  // Highlight only paragraph elements with class mark
  $("p.mark").css("background", "yellow");

  // Highlight only span elements inside the element with ID mark
  $("#mark span").css("background", "yellow");

  // Highlight li elements inside the ul elements
  $("ul li").css("background", "yellow");

  // Highlight li elements only inside the ul element with id mark
  $("ul#mark li").css("background", "red");
```

```
// Highlight li elements inside all the ul element with class mark
$("ul.mark li").css("background", "green");

// Highlight all anchor elements with target blank
$('a[target="_blank"]').css("background", "yellow");
});
</script>
</head>
<body>
  <p>This is a paragraph.</p>
  <p class="mark">This is another paragraph.</p>
  <p>This is one more paragraph.</p>
  <ul>
    <li>List item one</li>
    <li>List item two</li>
    <li>List item three</li>
  </ul>
  <ul id="mark">
    <li>List item one</li>
    <li>List <span>item two</span></li>
    <li>List item three</li>
  </ul>
  <ul class="mark">
    <li>List item one</li>
    <li>List item two</li>
    <li>List item three</li>
  </ul>
  <p>Go to <a href="https://www.genextcomputer.com/" target="_blank">Home
page</a></p>
</body>
```

```
</html>
```

---

## jQuery Custom Selector

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Custom Selector</title>
<style>
    /* Some custom style */
    *{
        padding: 5px;
    }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    // Highlight table rows appearing at odd places
    $("tr:odd").css("background", "yellow");

    // Highlight table rows appearing at even places
    $("tr:even").css("background", "orange");

    // Highlight first paragraph element
    $("p:first").css("background", "red");

    // Highlight last paragraph element
    $("p:last").css("background", "green");
```

```
// Highlight all input elements with type text inside a form
$("form :text").css("background", "purple");

// Highlight all input elements with type password inside a form
$("form :password").css("background", "blue");

// Highlight all input elements with type submit inside a form
$("form :submit").css("background", "violet");
});
</script>
</head>
<body>
  <table border="1">
    <thead>
      <tr>
        <th>No.</th>
        <th>Name</th>
        <th>Email</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>1</td>
        <td>Genext Crystal</td>
        <td>genextcrystal@mail.com</td>
      </tr>
      <tr>
        <td>2</td>
        <td>Iica Virtual</td>
        <td>iicavirtual@mail.com</td>
      </tr>
```

```
<tr>
  <td>3</td>
  <td>Genext Tiger</td>
  <td>genexttiger@mail.com</td>
</tr>
</tbody>
</table>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<p>This is one more paragraph.</p>
<form>
  <label>Name: <input type="text"></label>
  <label>Password: <input type="password"></label>
  <input type="submit" value="Sign In">
</form>
</body>
</html>
```

---

## jQuery Events

### What are Events

Events are often triggered by the user's interaction with the web page, such as when a link or button is clicked, text is entered into an input box or textarea, selection is made in a select box, key is pressed on the keyboard, the mouse pointer is moved etc. In some cases, the Browser itself can trigger the events, such as the page load and unload events.

jQuery enhances the basic event-handling mechanisms by offering the events methods for most native browser events, some of these methods are `ready()`, `click()`, `keypress()`, `focus()`, `blur()`, `change()`, etc.

### Mouse Events

```
<!DOCTYPE html>
```

```
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Click Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 20px;
    font: 20px sans-serif;
    background: khaki;
  }
</style>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).slideUp();
  });
});
</script>
</head>
<body>
  <p>Click on me and I'll disappear.</p>
  <p>Click on me and I'll disappear.</p>
  <p>Click on me and I'll disappear.</p>
</body>
</html>
```

---

## The `dblclick()` Method

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>
<meta charset="utf-8">
<title>Executing a Function on Double-click Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 20px;
    font: 20px sans-serif;
    background: khaki;
  }
</style>
<script>
$(document).ready(function(){
  $("p").dblclick(function(){
    $(this).slideUp();
  });
});
</script>
</head>
<body>
  <p>Double-click on me and I'll disappear.</p>
  <p>Double-click on me and I'll disappear.</p>
  <p>Double-click on me and I'll disappear.</p>
</body>
</html>
```

---

## The hover () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="utf-8">
<title>Executing a Function on Hover Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 20px;
    font: 20px sans-serif;
    background: #f2f2f2;
  }
  p.highlight{
    background: yellow;
  }
</style>
<script>
$(document).ready(function(){
  $("p").hover(function(){
    $(this).addClass("highlight");
  }, function(){
    $(this).removeClass("highlight");
  });
});
</script>
</head>
<body>
  <p>Place mouse pointer on me.</p>
  <p>Place mouse pointer on me.</p>
  <p>Place mouse pointer on me.</p>
</body>
</html>
```

---



## The mouseenter () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Mouseenter Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
    p{
        padding: 20px;
        font: 20px sans-serif;
        background: #f2f2f2;
    }
    p.highlight{
        background: yellow;
    }
</style>
<script>
$(document).ready(function(){
    $("p").mouseenter(function(){
        $(this).addClass("highlight");
    });
    $("p").mouseleave(function(){
        $(this).removeClass("highlight");
    });
});
</script>
</head>
```

```
<body>
  <p>Place mouse pointer on me.</p>
  <p>Place mouse pointer on me.</p>
  <p>Place mouse pointer on me.</p>
</body>
</html>
```

---

## The `mouseleave()` Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Mouseleave Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 20px;
    font: 20px sans-serif;
    background: #f2f2f2;
  }
  p.highlight{
    background: yellow;
  }
</style>
<script>
$(document).ready(function(){
  $("p").mouseenter(function(){
    $(this).addClass("highlight");
  });
```

```
    $("p").mouseleave(function(){
        $(this).removeClass("highlight");
    });
});
</script>
</head>
<body>
    <p>Place mouse pointer on me.</p>
    <p>Place mouse pointer on me.</p>
    <p>Place mouse pointer on me.</p>
</body>
</html>
```

---

## Keyboard Events

### The keypress () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Keypress Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
    p{
        padding: 10px;
        background: lightgreen;
        display: none;
    }
    div{
        margin: 20px 0;
```

```
}
</style>
<script>
$(document).ready(function(){
    var i = 0;
    $('input[type="text"]').keypress(function(){
        $("span").text(i += 1);
        $("p").show().fadeOut();
    });
});
</script>
</head>
<body>
    <input type="text">
    <div>Keypress: <span>0</span></div>
    <div><strong>Note:</strong> Enter something inside the input box and see
the result.</div>
    <p>Keypress is triggered.</p>
</body>
</html>
```

---

## The keydown () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Keydown Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
p{
```

```

padding: 10px;
background: lightgreen;
display: none;
}
div{
margin: 20px 0;
}
</style>
<script>
$(document).ready(function(){
var i = 0;
$('input[type="text"]').keydown(function(){
    $("span").text(i += 1);
    $("p").show().fadeOut();
});
});
</script>
</head>
<body>

<div>Keydown: <span>0</span></div>
<div><strong>Note:</strong> Enter something inside the input box and see
the result.</div>
<p>Keydown is triggered.</p>
</body>
</html>

```

---

## The keyup () Method

```

<!DOCTYPE html>
<html lang="en">

```

```
<head>
<meta charset="utf-8">
<title>Executing a Function on Keyup Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 10px;
    background: lightgreen;
    display: none;
  }
  div{
    margin: 20px 0;
  }
</style>
<script>
$(document).ready(function(){
  var i = 0;
  $('input[type="text"]').keyup(function(){
    $("span").text(i += 1);
    $("p").show().fadeOut();
  });
});
</script>
</head>
<body>
  <input type="text">
  <div>Keyup: <span>0</span></div>
  <div><strong>Note:</strong> Enter something inside the input box and see
the result.</div>
  <p>Keyup is triggered.</p>
```

```
</body>
```

```
</html>
```

---

## Form Events

### The change () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Change Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("select").change(function(){
        var selectedOption = $(this).find(":selected").val();
        alert("You have selected - " + selectedOption);
    });
});
</script>
</head>
<body>
    <form>
        <label>City:</label>
        <select>
            <option>London</option>
            <option>Paris</option>
            <option>New York</option>
        </select>
    </form>
```

```
<p><strong>Note:</strong> Select any value from the dropdown select and see the result.</p>
</body>
</html>
```

---

## The focus () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Focus Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
    label{
        display: block;
        margin: 5px 0;
    }
    label span{
        display: none;
    }
</style>
<script>
$(document).ready(function(){
    $("input").focus(function(){
        $(this).next("span").show().fadeOut("slow");
    });
});
</script>
</head>
<body>
```



```
<form>

  <label>Email: <input type="text"> <span>focus fire</span></label>

  <label>Password: <input type="password"> <span>focus fire</span></label>

  <label><input type="submit" value="Sign In"> <span>focus
fire</span></label>

</form>

<p><strong>Note:</strong> Click on the form control or press the "Tab" key to
set focus.</p>

</body>
</html>
```

---

## The blur () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Blur Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  label{
    display: block;
    margin: 5px 0;
  }
  label span{
    display: none;
  }
</style>
<script>
$(document).ready(function(){
  $("input").blur(function(){
    $(this).next("span").show().fadeOut("slow");
```

```
});  
});  
</script>  
</head>  
<body>  
  <form>  
    <label>Email: <input type="text"> <span>blur fire</span></label>  
    <label>Password: <input type="password"> <span>blur fire</span></label>  
    <label><input type="submit" value="Sign In"> <span>blur  
fire</span></label>  
  </form>  
  <p><strong>Note:</strong> Click away from the form control or press the "Tab"  
key to remove focus.</p>  
</body>  
</html>
```

---

# The submit() Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Form Submit Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
    .error{
        color: red;
    }
</style>
<script>
$(document).ready(function(){
    $("form").submit(function(event){
        var regex = /^[a-zA-Z]+$/;
        var currentValue = $("#firstName").val();
        if(regex.test(currentValue) == false){
            $("#result").html('<p class="error">Not
valid!</p>').show().fadeOut(1000);
            // Preventing form submission
            event.preventDefault();
        }
    });
});
</script>
</head>
<body>
    <p><strong>Note:</strong> If try to submit any invalid value. It will produce
an error.</p>
```

```
<form action="/examples/html/action.php" method="post" id="users">
  <label for="firstName">First Name:</label>
  <input type="text" name="first-name" id="firstName">
  <input type="submit" value="Submit">
  <div id="result"></div>
</form>
</body>
</html>
```

---

## Document/Window Events

### The ready () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Executing a Function on Ready Event in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  $("p").text("The DOM is now loaded and can be manipulated.");
});
</script>
</head>
<body>
  <p>Not loaded yet.</p>
</body>
</html>
```

---

### The resize () Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Executing a Function on Resize Event in jQuery</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    padding: 20px;

    font: 20px sans-serif;

    background: #f0e68c;

  }

</style>

<script>

$(document).ready(function(){

  $(window).resize(function() {

    $(window).bind("resize", function(){

      $("p").text("Window width: " + $(window).width() + ", " + "Window height: " + $(window).height());

    });

  });

});

</script>

</head>

<body>

  <p>Open the output in a new tab and resize the browser window by dragging the corners.</p>

</body>

</html>
```

---

## The scroll () Method

```
<!DOCTYPE html>

<html lang="en">
```

```
<head>

<meta charset="utf-8">

<title>Executing a Function on Scroll Event in jQuery</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    width: 100%;

    padding: 50px 0;

    text-align: center;

    font: bold 34px sans-serif;

    background: #f0e68c;

    position: fixed;

    top: 50px;

    display: none;

  }

  .dummy-content{

    height: 600px;

    font: 34px sans-serif;

    text-align: center;

  }

</style>

<script>

$(document).ready(function(){

  $(window).scroll(function() {

    $("p").show().fadeOut("slow");

  });

});

</script>

</head>

<body>

  <p>Scroll Happened!</p>
```

```
<div class="dummy-content">Scroll the viewport.</div>
<div class="dummy-content">Scroll the viewport.</div>
<div class="dummy-content">Scroll the viewport.</div>
<div class="dummy-content">Scroll the viewport.</div>
<div class="dummy-content">Scroll the viewport.</div>
</body>
</html>
```

---

GENEX

# jQuery Show and Hide Effects

## jQuery show () and hide () Methods

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Show Hide Effects</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    padding: 15px;

    background: #F0E68C;

  }

</style>

<script>

$(document).ready(function(){

  // Hide displayed paragraphs

  $(".hide-btn").click(function(){

    $("p").hide();

  });

  // Show hidden paragraphs

  $(".show-btn").click(function(){

    $("p").show();

  });

});

</script>

</head>

<body>
```



```
<button type="button" class="hide-btn">Hide Paragraphs</button>
<button type="button" class="show-btn">Show Paragraphs</button>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

---

## According to timing function

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Animated Show Hide Effects</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 15px;
    background: #F0E68C;
  }
</style>
<script>
$(document).ready(function(){
  // Hide displayed paragraphs with different speeds
  $(".hide-btn").click(function(){
    $(".p.normal").hide();
    $(".p.fast").hide("fast");
    $(".p.slow").hide("slow");
    $(".p.very-fast").hide(50);
    $(".p.very-slow").hide(2000);
  });
  // Show hidden paragraphs with different speeds
```

```
$(".show-btn").click(function(){
    $(".p.normal").show();
    $(".p.fast").show("fast");
    $(".p.slow").show("slow");
    $(".p.very-fast").show(50);
    $(".p.very-slow").show(2000);
});
});
</script>
</head>
<body>
    <button type="button" class="hide-btn">Hide Paragraphs</button>
    <button type="button" class="show-btn">Show Paragraphs</button>
    <p class="very-fast">This paragraph will show/hide with very fast speed.</p>
    <p class="normal">This paragraph will show/hide with default speed.</p>
    <p class="fast">This paragraph will show/hide with fast speed.</p>
    <p class="slow">This paragraph will show/hide with slow speed.</p>
    <p class="very-slow">This paragraph will show/hide with very slow speed.</p>
</body>
</html>
```

---

## With alert function

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Show Hide Effects with Callback</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
    p{
        padding: 15px;
```

```
        background: #F0E68C;
    }
</style>

<script>
$(document).ready(function(){
    // Display alert message after hiding paragraphs
    $(".hide-btn").click(function(){
        $("p").hide("slow", function(){
            // Code to be executed
            alert("The hide effect is completed.");
        });
    });

    // Display alert message after showing paragraphs
    $(".show-btn").click(function(){
        $("p").show("slow", function(){
            // Code to be executed
            alert("The show effect is completed.");
        });
    });
});
</script>

</head>

<body>
    <button type="button" class="hide-btn">Hide Paragraphs</button>
    <button type="button" class="show-btn">Show Paragraphs</button>

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

</html>
```

---

# jQuery toggle () Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Toggle Effect</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    padding: 15px;

    background: #F0E68C;

  }

</style>

<script>

$(document).ready(function(){

  // Toggles paragraphs display

  $(".toggle-btn").click(function(){

    $("p").toggle();

  });

});

</script>

</head>

<body>

  <button type="button" class="toggle-btn">Toggle Paragraphs</button>

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

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

</body>

</html>
```

---

# jQuery Fading Effects

## jQuery fadeIn () and fadeOut () Methods

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Fade-In and Fade-Out Effects</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

p{

padding: 15px;

background: #DDA0DD;

}

</style>

<script>

$(document).ready(function(){

// Fadeing out displayed paragraphs

$(".out-btn").click(function(){

$(".p").fadeOut();

});

// Fading in hidden paragraphs

$(".in-btn").click(function(){

$(".p").fadeIn();

});

});

</script>

</head>

<body>
```

```
<button type="button" class="out-btn">Fade Out Paragraphs</button>
<button type="button" class="in-btn">Fade In Paragraphs</button>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

---

## With timing Function

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Fade-In and Fade-Out Effects with Different Speeds</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 15px;
    background: #DDA0DD;
  }
</style>
<script>
$(document).ready(function(){
  // Fading out displayed paragraphs with different speeds
  $(".out-btn").click(function(){
    $(".p.normal").fadeOut();
    $(".p.fast").fadeOut("fast");
    $(".p.slow").fadeOut("slow");
    $(".p.very-fast").fadeOut(50);
    $(".p.very-slow").fadeOut(2000);
  });
});
```

```
// Fading in hidden paragraphs with different speeds
$(".in-btn").click(function(){
    $(".p.normal").fadeIn();
    $(".p.fast").fadeIn("fast");
    $(".p.slow").fadeIn("slow");
    $(".p.very-fast").fadeIn(50);
    $(".p.very-slow").fadeIn(2000);
});
});
</script>
</head>
<body>
    <button type="button" class="out-btn">Fade Out Paragraphs</button>
    <button type="button" class="in-btn">Fade In Paragraphs</button>
    <p class="very-fast">This paragraph will fade in/out with very fast speed.</p>
    <p class="normal">This paragraph will fade in/out with default speed.</p>
    <p class="fast">This paragraph will fade in/out with fast speed.</p>
    <p class="slow">This paragraph will fade in/out with slow speed.</p>
    <p class="very-slow">This paragraph will fade in/out with very slow speed.</p>
</body>
</html>
```

---

## With alert Function

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Fade-In and Fade-Out Effects with Callback</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
p{
```

```
padding: 15px;
background: #DDA0DD;
}
</style>
<script>
$(document).ready(function(){
    // Display alert message after fading out paragraphs
    $(".out-btn").click(function(){
        $(".p").fadeOut("slow", function(){
            // Code to be executed
            alert("The fade-out effect is completed.");
        });
    });

    // Display alert message after fading in paragraphs
    $(".in-btn").click(function(){
        $(".p").fadeIn("slow", function(){
            // Code to be executed
            alert("The fade-in effect is completed.");
        });
    });
});
</script>
</head>
<body>
    <button type="button" class="out-btn">Fade Out Paragraphs</button>
    <button type="button" class="in-btn">Fade In Paragraphs</button>
    <p>This is a paragraph.</p>
</body>
</html>
```

---



# jQuery fadeToggle () Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Fade-Toggle Effect</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    padding: 15px;

    background: #DDA0DD;

  }

</style>

<script>

$(document).ready(function(){

  // Toggles paragraphs display with fading

  $(".toggle-btn").click(function(){

    $("p").fadeToggle();

  });

});

</script>

</head>

<body>

  <button type="button" class="toggle-btn">Fade Toggle Paragraphs</button>

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

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

</body>

</html>
```

---

## jQuery fadeTo () Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Fade-To Effect</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    display: none;

    padding: 15px;

    background: #DDA0DD;

  }

</style>

<script>

$(document).ready(function(){

  // Fade to paragraphs with different opacity

  $(".to-btn").click(function(){

    $(".p.none").fadeTo("fast", 0);

    $(".p.partial").fadeTo("slow", 0.5);

    $(".p.complete").fadeTo(2000, 1);

  });

});

</script>

</head>

<body>

  <button type="button" class="to-btn">Fade To Hidden Paragraphs</button>

  <p class="none">This is a paragraph.</p>

  <p class="partial">This is another paragraph.</p>

  <p class="complete">This is one more paragraph.</p>

</body>

</html>
```

---

# jQuery Sliding Effects

## jQuery `slideUp()` and `slideDown()` Methods

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Slide-Up and Slide-Down Effects</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

p{

padding: 15px;

background: #B0C4DE;

}

</style>

<script>

$(document).ready(function(){

// Slide up displayed paragraphs

$(".up-btn").click(function(){

$(".p").slideUp();

});

// Slide down hidden paragraphs

$(".down-btn").click(function(){

$(".p").slideDown();

});

});

</script>

</head>

<body>
```

```
<button type="button" class="up-btn">Slide Up Paragraphs</button>
<button type="button" class="down-btn">Slide Down Paragraphs</button>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
</body>
</html>
```

---

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Slide-Up and Slide-Down Effects with Different Speeds</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  p{
    padding: 15px;
    background: #B0C4DE;
  }
</style>
<script>
$(document).ready(function(){
  // Sliding up displayed paragraphs with different speeds
  $(".up-btn").click(function(){
    $(".p.normal").slideUp();
    $(".p.fast").slideUp("fast");
    $(".p.slow").slideUp("slow");
    $(".p.very-fast").slideUp(50);
    $(".p.very-slow").slideUp(2000);
  });

  // Sliding down hidden paragraphs with different speeds
  $(".down-btn").click(function(){
```

```
    $(".p.normal").slideDown();
    $(".p.fast").slideDown("fast");
    $(".p.slow").slideDown("slow");
    $(".p.very-fast").slideDown(50);
    $(".p.very-slow").slideDown(2000);
  });
});
</script>
</head>
<body>
  <button type="button" class="up-btn">Slide Up Paragraphs</button>
  <button type="button" class="down-btn">Slide Down Paragraphs</button>
  <p class="very-fast">This paragraph will fade in/out with very fast speed.</p>
  <p class="normal">This paragraph will fade in/out with default speed.</p>
  <p class="fast">This paragraph will fade in/out with fast speed.</p>
  <p class="slow">This paragraph will fade in/out with slow speed.</p>
  <p class="very-slow">This paragraph will fade in/out with very slow speed.</p>
</body>
</html>
```

---

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>Example of jQuery Slide-Up and Slide-Down Effects with Callback</title>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <style>
    p{
      padding: 15px;
      background: #B0C4DE;
    }
  </style>
```

```
<script>
$(document).ready(function(){
    // Display alert message after sliding up paragraphs
    $(".up-btn").click(function(){
        $(".p").slideUp("slow", function(){
            // Code to be executed
            alert("The slide-up effect is completed.");
        });
    });

    // Display alert message after sliding down paragraphs
    $(".down-btn").click(function(){
        $(".p").slideDown("slow", function(){
            // Code to be executed
            alert("The slide-down effect is completed.");
        });
    });
});
</script>
</head>
<body>
    <button type="button" class="up-btn">Slide Up Paragraphs</button>
    <button type="button" class="down-btn">Slide Down Paragraphs</button>
    <p>This is a paragraph.</p>
</body>
</html>
```

---

## jQuery slideToggle () Method

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="utf-8">

<title>Example of jQuery Slide-Toggle Effect</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

    p{

        padding: 15px;

        background: #B0C4DE;

    }

</style>

<script>

$(document).ready(function(){

    // Toggles paragraphs display with sliding

    $(".toggle-btn").click(function(){

        $("p").slideToggle();

    });

});

</script>

</head>

<body>

    <button type="button" class="toggle-btn">Slide Toggle Paragraphs</button>

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

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

</body>

</html>
```

---

# jQuery Animation Effects

## jQuery animate () Method

### Syntax

The basic syntax of the jQuery `animate()` method can be given with:

```
$(selector).animate({ properties }, duration, callback);
```

The parameters of the `animate()` method have the following meanings:

- The required properties parameter defines the **CSS properties** to be animated.
- The optional duration parameter specifies how long the animation will run. Durations can be specified either using one of the predefined string `'slow'` or `'fast'`, or in a number of milliseconds; higher values indicate slower animations.
- The optional **callback** parameter is a function to call once the animation is complete.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Animation Effects</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  img{
    position: relative; /* Required to move element */
  }
</style>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("img").animate({
      left: 300
    });
  });
});
</script>
</head>
```



```
<body>

  <button type="button">Start Animation</button>

  <p>

  </p>

</body>

</html>
```

---

## Animate Multiple Properties At Once

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Multiple Properties Animation</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  .box{

    width: 100px;

    height: 100px;

    background: #9d7ede;

    margin-top: 30px;

    border-style: solid; /* Required to animate border width */

    border-color: #6f40ce;

  }

</style>

<script>

$(document).ready(function(){

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

    $(".box").animate({

      width: "300px",

      height: "300px",
```

```
        marginLeft: "150px",
        borderWidth: "10px",
        opacity: 0.5
    });
});
});
</script>
</head>
<body>
    <button type="button">Start Animation</button>
    <div class="box"></div>
</body>
</html>
```

---

## Animate Multiple Properties One by One or Queued Animations

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Queued Animation</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
    .box{
        width: 100px;
        height: 100px;
        background: #9d7ede;
        margin-top: 30px;
        border-style: solid; /* Required to animate border width */
```

```
        border-color: #6f40ce;
    }
</style>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $(".box")
            .animate({width: "300px"})
            .animate({height: "300px"})
            .animate({marginLeft: "150px"})
            .animate({borderWidth: "10px"})
            .animate({opacity: 0.5});
    });
});
</script>
</head>
<body>
    <button type="button">Start Animation</button>
    <div class="box"></div>
</body>
</html>
```

---

## Animate Properties with Relative Values

You can also define the relative values for the animated properties. If a value is specified with a leading `+=` or `-=` prefix, then the target value is calculated by adding or subtracting the given number from the current value of the property.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Animation with Relative Values</title>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<style>
```

```
.box{  
    width: 100px;  
    height: 100px;  
    background: #9d7ede;  
    margin-top: 30px;  
    position: relative; /* Required to move element */  
}
```

```
</style>
```

```
<script>
```

```
$(document).ready(function(){  
    $("button").click(function(){  
        $(".box").animate({  
            top: "+=50px",  
            left: "+=50px",  
            width: "+=50px",  
            height: "+=50px"  
        });  
    });  
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
<button type="button">Start Animation</button>
```

```
<p><strong>Note:</strong> Click the "Start Animation" button multiple times to see how the relative value works.</p>
```

```
<div class="box"></div>
```

```
</body>
```

```
</html>
```

---

# Animate Properties with Pre-defined Values

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Animation with Pre-defined Values</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

    .box{

        width: 80%;

        height: 200px;

        background: #9d7ede;

        margin-top: 30px;

    }

</style>

<script>

$(document).ready(function(){

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

        $(".box").animate({

            width: 'toggle'

        });

    });

});

</script>

</head>

<body>

    <button type="button">Toggle Animation</button>

    <div class="box"></div>

</body>

</html>
```

---

# jQuery Stop Animations

## jQuery stop () Method

The jQuery `stop()` method is used to stop the jQuery animations or effects currently running on the selected elements before it completes.

The basic syntax of the jQuery `stop()` method can be given with:

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

The parameters in the above syntax have the following meanings:

- The optional *stopAll* Boolean parameter specifies whether to remove queued animation or not. Default value is `false`, that means only the current animation will be stopped, rest of the animations in the queue will run afterwards.
- The optional *goToEnd* Boolean parameter specifies whether to complete the current animation immediately. Default value is `false`.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Stop Currently Running Animations</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
  img{
    position: relative; /* Required to move element */
  }
</style>
<script>
$(document).ready(function(){
  // Start animation
  $(".start-btn").click(function(){
    $("img").animate({left: "+=150px"}, 2000);
  });
});
```

```
// Stop running animation
$(".stop-btn").click(function(){
    $(".img").stop();
});

// Start animation in the opposite direction
$(".back-btn").click(function(){
    $(".img").animate({left: "-=150px"}, 2000);
});

// Reset to default
$(".reset-btn").click(function(){
    $(".img").animate({left: "0"}, "fast");
});
});
</script>
</head>
<body>
    <button type="button" class="start-btn">Start</button>
    <button type="button" class="stop-btn">Stop</button>
    <button type="button" class="back-btn">Back</button>
    <button type="button" class="reset-btn">Reset</button>
    <p>
        
    </p>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
```

---

```
<meta charset="utf-8">

<title>jQuery Stop Current Animation and Play Next Animation in Queue</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

    .box{

        width: 300px;

        height: 200px;

        background: #9d7ede;

        margin-top: 20px;

        border: 3px solid #6f40ce;

    }

</style>

<script>

$(document).ready(function(){

    // Kill and toggle the current sliding animation

    $(".toggle-btn").on("click", function(){

        $(".box").stop().slideToggle(1000);

    });

});

</script>

</head>

<body>

    <p><strong>Note:</strong> Click the "Slide Toggle" button to start the animation, then click again before the animation is completed to understand this example.</p>

    <button type="button" class="toggle-btn">Slide Toggle</button>

    <div class="box"></div>

</body>

</html>
```

---

## Creating Smooth Hover Effect

```
<!DOCTYPE html>
```



```
<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Smooth Hover Effect</title>

<style>

    .box{

        width: 500px;

        height: 300px;

        border: 5px solid #000;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $(".box").hover(function(){

        $(this).find("img").stop(true, true).fadeOut();

    }, function(){

        $(this).find("img").stop(true, true).fadeIn();

    });

});

</script>

</head>

<body>

    <div class="box">

    </div>

    <p><strong>Note:</strong> Place and remove the mouse pointer over the image to see the effect.</p>

</body>

</html>
```

---

# jQuery Chaining

## jQuery Method Chaining

The jQuery provides another robust feature called method chaining that allows us to perform multiple action on the same set of elements, all within a single line of code.

This is possible because most of the jQuery methods return a jQuery object that can be further used to call another method.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Method Chaining</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

/* Some custom styles to beautify this example */

p {

width: 200px;

padding: 40px 0;

font: bold 24px sans-serif;

text-align: center;

background: #aaccaa;

border: 1px solid #63a063;

box-sizing: border-box;

}

</style>

<script>

$(document).ready(function(){

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

        $(".p").animate({width: "100%"}).animate({fontSize: "46px"}).animate({borderWidth: 30});

    });

});
```

```
$(".reset").click(function(){
    $("p").removeAttr("style");
});
});
</script>
</head>
<body>
    <p>Hello World!</p>
    <button type="button" class="start">Start Chaining</button>
    <button type="button" class="reset">Reset</button>
</body>
</html>
```

---

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Example of jQuery Method Chaining</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<style>
/* Some custom styles to beautify this example */
p {
    width: 200px;
    padding: 40px 0;
    font: bold 24px sans-serif;
    text-align: center;
    background: #aaccaa;
    border: 1px solid #63a063;
    box-sizing: border-box;
}
</style>
<script>
```

```
$(document).ready(function(){
    $(".start").click(function(){
        $("p")
            .animate({width: "100%"})
            .animate({fontSize: "46px"})
            .animate({borderWidth: 30});
    });
    $(".reset").click(function(){
        $("p").removeAttr("style");
    });
});
</script>
</head>
<body>
    <p>Hello World!</p>
    <button type="button" class="start">Start Chaining</button>
    <button type="button" class="reset">Reset</button>
</body>
</html>
```

---

## jQuery Callback Functions

JavaScript statements are executed line by line. But, since jQuery effect takes some time to finish the next line code may execute while the previous effect is still running. To prevent this from happening jQuery provides a callback function for each effect method.

A callback function is a function that is executed once the effect is complete. The callback function is passed as an argument to the effect methods and they typically appear as the last argument of the method. For example, the basic syntax of the jQuery `slideToggle()` effect method with a callback function can be given with:

```
$(selector).slideToggle(duration, callback);
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>

<meta charset="utf-8">

<title>Example of jQuery Effect Method without Callback Function</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  p{

    background:yellow;

    font-size: 24px;

    padding:20px;

  }

</style>

<script>

$(document).ready(function(){

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

    $("p").slideToggle("slow");

    alert("The slide toggle effect has completed.");

  });

});

</script>

</head>

<body>

  <p>This is paragraph.</p>

  <button type="button">Click me</button>

</body>

</html>
```

---

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Effect Method with Callback Function</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<style>

  p{

    background:yellow;

    font-size: 24px;

    padding:20px;

  }

</style>

<script>

$(document).ready(function(){

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

    $("p").slideToggle("slow", function(){

      alert("The slide toggle effect has completed.");

    });

  });

});

</script>

</head>

<body>

  <p>This is paragraph.</p>

  <button type="button">Click me</button>

</body>

</html>
```

---

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Example of jQuery Callback Executed Multiple Times</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<style>

  h1{

    display:none;
```

```
background:red;

padding:20px;

}

p{

background:yellow;

font-size: 24px;

padding:20px;

}

</style>

<script>

$(document).ready(function(){

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

        $("h1, p").slideToggle("slow", function(){

            alert("The slide toggle effect has completed.");

        });

    });

});

</script>

</head>

<body>

    <h1>This is heading</h1>

    <p>This is paragraph.</p>

    <button type="button">Click me</button>

</body>

</html>
```

---

## jQuery Getters & Setter

### jQuery Get or Set Contents and Values

Some jQuery methods can be used to either assign or read some value on a selection. A few of these methods are `text()`, `html()`, `attr()`, and `val()`.

When these methods are called with no argument, it is referred to as a *getters*, because it gets (or reads) the value of the element. When these methods are called with a value as an argument, it's referred to as a *setter* because it sets (or assigns) that value.

## jQuery `text()` Method

The jQuery `text()` method is either used to get the combined text contents of the selected elements, including their descendants, or set the text contents of the selected elements.

### Get Contents with `text()` Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Get Text Contents of the Elements</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $(".btn-one").click(function(){

        var str = $("p").text();

        alert(str);

    });

    $(".btn-two").click(function(){

        var str = $("p:first").text();

        alert(str);

    });

    $(".btn-three").click(function(){

        var str = $("p.extra").text();

        alert(str);

    });

});

</script>

</head>
```



```
<body>

  <button type="button" class="btn-one">Get All Paragraph's Text</button>

  <button type="button" class="btn-two">Get First Paragraph's Text</button>

    <button type="button" class="btn-three">Get Last Paragraph's Text</button>

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

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

    <p class="extra">This is one more paragraph.</p>

</body>

</html>
```

---

## Set Contents with `text()` Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Set Text Contents of the Elements</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

  $(".btn-one").click(function(){

    $("p").text("This is demo text.");

  });

  $(".btn-two").click(function(){

    $("p:first").text("This is another demo text.");

  });

  $(".btn-three").click(function(){

    $("p.empty").text("This is one more demo text.");

  });

});

</script>

</head>
```

```
<body>

  <button type="button" class="btn-one">Set All Paragraph's Text</button>

  <button type="button" class="btn-two">Set First Paragraph's Text</button>

  <button type="button" class="btn-three">Set Empty Paragraph's Text</button>

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

  <p>This is another test paragraph.</p>

  <p class="empty"></p>

</body>

</html>
```

---

## jQuery `html()` Method

The jQuery `html()` method is used to get or set the HTML contents of the elements.

### Get HTML Contents with `html()` Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Get HTML Contents of an Element</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

  $(".btn-one").click(function(){

    var str = $("p").html();

    alert(str);

  });

  $(".btn-two").click(function(){

    var str = $("#container").html();

    alert(str);

  });

});

});
```

```
</script>
</head>
<body>
  <button type="button" class="btn-one">Get Paragraph's HTML Contents</button>
  <button type="button" class="btn-two">Get Container's HTML Contents</button>
  <div id="container">
    <h1>Hello World!</h1>
    <p>The quick <b>brown fox</b> jumps over the lazy dog.</p>
  </div>
</body>
</html>
```

---

## Set HTML Contents with `html()` Method

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <title>jQuery Set HTML Contents of the Element</title>
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script>
    $(document).ready(function(){
      $("button").click(function(){
        $("body").html("<p>Hello World!</p>");
      });
    });
  </script>
</head>
<body>
  <button type="button">Write Message</button>
</body>
</html>
```

---

# jQuery `attr()` Method

You can use the jQuery `attr()` method to either get the value of an element's attribute or set one or more attributes for the selected element.

## Get Attribute Value with `attr()` Method

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Get an Element's Attribute Value</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $(".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="http://www.genextcomputer.com/">Tutorial Republic</a></p>

</body>
```

</html>

---

## Set Attributes with `attr()` Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Set Element's Attribute Value</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $('input[type="checkbox"]').attr("checked", "checked");
    });
});
</script>
</head>
<body>
    <p><label><input type="checkbox"></label> I agree with terms and conditions.</p>
    <button type="button">Check</button>
</body>
</html>
```

---

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Set Multiple Attribute for the Elements</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
```

```

$("button").click(function(){
    $("img").attr({
        "class" : "frame",
        "title" : "Hot Air Balloons"
    });
});

});
</script>
<style>
    .frame{
        border: 6px solid #000;
    }
</style>
</head>
<body>
    <button type="button">Set Attributes for Image</button>
    <p>
        
    </p>
</body>
</html>

```

---

## jQuery val () Method

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Get a Form Field Value</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){

```

```
$("#button.get-name").click(function(){
    var name = $("#name").val();
    alert(name);
});
$("#button.get-comment").click(function(){
    var comment = $("#comment").val();
    alert(comment);
});
$("#button.get-city").click(function(){
    var city = $("#city").val();
    alert(city);
});
});
</script>
</head>
<body>
    <form>
        <table>
            <tr>
                <td>Name:</td>
                <td>
                    <input type="text" id="name">
                </td>
            </tr>
            <tr>
                <td>Comments:</td>
                <td>
                    <textarea rows="4" cols="30" id="comment"></textarea>
                </td>
            </tr>
            <tr>
```

```
<td>City:</td>
<td>
  <select id="city">
    <option>London</option>
    <option>Paris</option>
    <option>New York</option>
  </select>
</td>
</tr>
</table>
</form>
<p><strong>Note:</strong> Fill the above form and click the following button to get the value.</p>
<button type="button" class="get-name">Get Name</button>
<button type="button" class="get-comment">Get Comment</button>
<button type="button" class="get-city">Get City</button>
</body>
</html>
```

---

## Set the Values of Form Fields with `val()` Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Set Form Fields Values</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    var text = $(this).text();
    $('input[type="text"]').val(text);
  });
});
```



```
});  
</script>  
</head>  
<body>  
  <button type="button">Discovery</button>  
  <button type="button">Atlantis</button>  
  <button type="button">Endeavour</button>  
  <p><strong>Note:</strong> Click the above buttons to set the value of following input box.</p>  
  <p>  
    <input type="text">  
  </p>  
</body>  
</html>
```

---

## jQuery Insert New Content

jQuery provides several methods, like `append()`, `prepend()`, `html()`, `text()`, `before()`, `after()`, `wrap()` etc. that allows us to insert new content inside an existing element.

The jQuery `html()` and `text()` methods have already covered in the previous chapter, so in this chapter, we will discuss about the rest of them.

## jQuery `append()` Method

The jQuery `append()` method is used to insert content to the end of the selected elements.

```
<!DOCTYPE html>  
<html lang="en">  
  <head>  
    <meta charset="utf-8">  
    <title>Inserting HTML Contents At the End of the Elements in jQuery</title>  
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>  
  <script>
```

```
$(document).ready(function(){  
    // Append all paragraphs on document ready  
    $("p").append(' <a href="#">read more...</a>');  
  
    // Append a div container on button click  
    $("button").click(function(){  
        $("#container").append("This is demo text.");  
    });  
});  
</script>  
</head>  
<body>  
    <button type="button">Insert Text</button>  
    <div id="container">  
        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.</p>  
        <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti.</p>  
    </div>  
</body>  
</html>
```

---

## jQuery `prepend()` Method

The `prepend()` method is used to insert content to the beginning of the selected elements.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="utf-8">  
    <title>Inserting HTML Contents At the Start of the Elements in jQuery</title>  
    <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<script>

$(document).ready(function(){

    // Prepend all paragraphs on document ready

    $("p").prepend("<strong>Note:</strong> ");


    // Prepend a div container on button click

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

        $("#container").prepend("This is demo text.");

    });

});

</script>

</head>

<body>

    <button type="button">Insert Text</button>

    <div id="container">

        <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante.</p>

        <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti.</p>

    </div>

</body>

</html>
```

---

## Insert Multiple Elements with `append()` & `prepend()` Method

The jQuery `append()` and `prepend()` also supports passing in multiple arguments as input.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Append or Prepend Multiple Elements in jQuery</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```

<script>
$(document).ready(function(){
    $("button").click(function(){
        var newHeading = "<h1>Important Note:</h1>";
        var newParagraph = document.createElement("p");
            newParagraph.innerHTML = "<em>Lorem Ipsum is dummy text...</em>";
        var newImage = $('');
        $("body").append(newHeading, newParagraph, newImage);
    });
});
</script>
</head>
<body>
    <button type="button">Insert Contents</button>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante, metus ac nisl bibendum.</p>
</body>
</html>

```

---

## jQuery before() Method

The jQuery `before()` method is used to insert content before the selected elements.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Inserting HTML Contents Before or After the Elements in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    // Add content after a div container on document ready
    $("#container").after("<p>&mdash; The End &mdash;</p>");

```

```
// Add content before a div container on document ready
$("#container").before("<p>&mdash; Demo Text &mdash;</p>");

// Add content after heading on button click
$("button.insert-after").click(function(){
    $("h1").after('');
});

// Add content before heading on button click
$("button.insert-before").click(function(){
    $("h1").before('');
});
});
</script>
<style>
h1{
    display: inline-block; /* To place marker image and heading in one line */
}
body{
    text-align: center;
}
</style>
</head>
<body>
<h1>Hello World</h1>

<hr>

<button type="button" class="insert-before">Insert Before</button>

<button type="button" class="insert-after">Insert After</button>

<hr>

<div id="container">
```

<p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>

<p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

</div>

</body>

</html>

---

## jQuery after() Method

The jQuery `after()` method is used to insert content after the selected elements.

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Inserting HTML Contents Before or After the Elements in jQuery</title>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<script>
```

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

```
    // Add content after a div container on document ready
```

```
    $("#container").after("<p>&mdash; The End &mdash;</p>");
```

```
    // Add content before a div container on document ready
```

```
    $("#container").before("<p>&mdash; Demo Text &mdash;</p>");
```

```
    // Add content after heading on button click
```

```
    $("button.insert-after").click(function(){
```

```
        $("h1").after('');
```

```

});

// Add content before heading on button click
$("button.insert-before").click(function(){
    $("h1").before('');
});
});
</script>
<style>
h1{
    display: inline-block; /* To place marker image and heading in one line */
}
body{
    text-align: center;
}
</style>
</head>
<body>
<h1>Hello World</h1>
<hr>
<button type="button" class="insert-before">Insert Before</button>
<button type="button" class="insert-after">Insert After</button>
<hr>
<div id="container">
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla. Vivamus nisl leo, blandit at bibendum eu, tristique eget risus. Integer aliquet quam ut elit suscipit, id interdum neque porttitor. Integer faucibus ligula.</p>
    <p>Quis quam ut magna consequat faucibus. Pellentesque eget nisi a mi suscipit tincidunt. Ut tempus dictum risus. Pellentesque viverra sagittis quam at mattis. Suspendisse potenti. Aliquam sit amet gravida nibh, facilisis gravida odio. Phasellus auctor velit at lacus blandit, commodo iaculis justo viverra. Etiam vitae est arcu. Mauris vel congue dolor. Aliquam eget mi mi. Fusce quam tortor, commodo ac dui quis, bibendum viverra erat. Maecenas mattis lectus enim, quis tincidunt dui molestie euismod. Curabitur et diam tristique, accumsan nunc eu, hendrerit tellus.</p>

```

```
</div>
</body>
</html>
```

---

## Insert Multiple Elements with `before()` & `after()` Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Inserting Multiple Elements Before or After the Elements in jQuery</title>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        var newHeading = "<h2>Important Note:</h2>";
        var newParagraph = document.createElement("p");
        newParagraph.innerHTML = "<em>Lorem Ipsum is dummy text...</em>";
        var newImage = $('');
        $("p").before(newHeading, newParagraph, newImage);
    });
});
</script>
</head>
<body>
    <button type="button">Insert Contents</button>

    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante, metus ac nisl bibendum.</p>
</body>
</html>
```

---



# jQuery `wrap()` Method

The jQuery `wrap()` method is used to wrap an HTML structure around the selected elements.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Wrapping HTML Around the Elements in jQuery</title>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    // Wrap div container with another div on document ready

    $(".container").wrap('<div class="wrapper"></div>');

    // Wrap paragraph's content on button click

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

        $("p").contents().wrap("<em><b></b></em>");

    });

});

</script>

<style>

.wrapper{

    padding: 20px;

    background: #f0e68c;

    margin: 10px 0;

}

.container{

    padding: 15px;

    background: #fff;

    font-size: 24px;

}
```

```
</style>
</head>
<body>
  <button type="button">Wrap Demo Text</button>
  <div class="container">
    <p>This is demo text.</p>
  </div>
</body>
</html>
```

---

## jQuery Remove Elements or Contents

jQuery provides handful of methods, such as `empty()`, `remove()`, `unwrap()` etc. to remove existing HTML elements or contents from the document.

### jQuery `empty()` Method

The jQuery `empty()` method removes all child elements as well as other descendant elements and the text content within the selected elements from the DOM.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Removing the Contents of the Elements in jQuery</title>
<style>
.container{
  padding: 10px;
  background: #f0e68C;
  border: 1px solid #bead18;
}
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
```

```

$(document).ready(function(){
    // Empty container div on button click
    $("button").click(function(){
        $(".container").empty();
    });
});
</script>
</head>
<body>
    <div class="container">
        <h1>Hello World!</h1>
        <p class="hint"><strong>Note:</strong> If you click the following button it will remove all the contents of the
        container div including the button.</p>
        <button type="button">Empty Container</button>
    </div>
</body>
</html>

```

---

## jQuery remove() Method

The jQuery `remove()` method removes the selected elements from the DOM as well as everything inside it. In addition to the elements themselves, all bound events and jQuery data associated with the elements are removed.

```

<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Removing the Elements from DOM in jQuery</title>
<style>
.container{
    padding: 10px;
    background: #f0e68C;

```

```
border: 1px solid #bead18;
}
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    // Removes paragraphs with class "hint" from DOM on button click
    $("button").click(function(){
        $("p.hint").remove();
    });
});
</script>
</head>
<body>
<div class="container">
    <h1>Hello World!</h1>
    <p class="hint"><strong>Note:</strong> If you click the following button it will remove this paragraph.</p>
    <button type="button">Remove Hint Paragraph</button>
</div>
</body>
</html>
```

---

## jQuery `unwrap()` Method

The jQuery `unwrap()` method removes the parent elements of the selected elements from the DOM. This is typically the inverse of the `wrap()` method.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Removing the Parents of the Elements from DOM in jQuery</title>
<style>
```

```
.container{
  padding: 10px;
  background: #f0e68c;
  border: 1px solid #bead18;
}

</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  // Removes the paragraph's parent element on button click
  $("button").click(function(){
    $("p").unwrap();
  });
});
</script>
</head>
<body>
  <div class="container">
    <h1>Hello World!</h1>
    <p class="hint"><strong>Note:</strong> If you click the following button it will remove the parent element of this
    paragraph.</p>
    <button type="button">Remove Paragraph's Parent</button>
  </div>
</body>
</html>
```

---

## jQuery `removeAttr()` Method

The jQuery `removeAttr()` method removes an [attribute](#) from the selected elements.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="utf-8">

<title>Removing an Attribute from the Elements in jQuery</title>

<style>

    a{

        font-size: 18px;

        margin-right: 20px;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    // Removes the hyperlink's href attribute on button click

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

        $("a").removeAttr("href");

    });

});

</script>

</head>

<body>

    <div class="container">

        <p>

            <a href="https://www.tutorialrepublic.com/">Home</a>

            <a href="https://www.tutorialrepublic.com/about-us.php">About</a>

            <a href="https://www.tutorialrepublic.com/contact-us.php">Contact</a>

        </p>

        <button type="button">Remove Attribute</button>

    </div>

</body>

</html>
```

---

# jQuery CSS Classes Manipulation

jQuery provides several methods, such as `addClass()`, `removeClass()`, `toggleClass()`, etc. to manipulate the CSS classes assigned to HTML elements.

## jQuery `addClass()` Method

The jQuery `addClass()` method adds one or more classes to the selected elements.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Adding a Class to the Elements in jQuery</title>

<style>

    .page-header{

        color: red;

        text-transform: uppercase;

    }

    .highlight{

        background: yellow;

    }

    .hint{

        font-style: italic;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

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

        $("h1").addClass("page-header");

        $("p.hint").addClass("highlight");

    });

});
```

```
});  
});  
</script>  
</head>  
<body>  
  <h1>Demo Text</h1>  
  <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>  
  <p class="hint"><strong>Tip:</strong> Lorem Ipsum is dummy text.</p>  
  <button type="button">Add Class</button>  
</body>  
</html>
```

---

## jQuery `removeClass()` Method

Similarly, you can remove the classes from the elements using the jQuery `removeClass()` method. The `removeClass()` method can remove a single class, multiple classes, or all classes at once from the selected elements.

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
  <meta charset="utf-8">  
  <title>Removing Classes from the Elements in jQuery</title>  
<style>  
  .page-header{  
    color: red;  
    text-transform: uppercase;  
  }  
  .highlight{  
    background: yellow;  
  }  
  .hint{  
    font-style: italic;
```



```
}
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("h1").removeClass("page-header");
        $("p").removeClass("hint highlight");
    });
});
</script>
</head>
<body>
    <h1 class="page-header">Demo Text</h1>
    <p>Lorem ipsum dolor sit amet, consectetur adipiscing elit...</p>
    <p class="hint highlight"><strong>Tip:</strong> Lorem Ipsum is dummy text.</p>
    <button type="button">Remove Class</button>
</body>
</html>
```

---

## jQuery `toggleClass()` Method

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Toggle the Classes of the Elements in jQuery</title>
<style>
p{
    padding: 10px;
    cursor: pointer;
    font: bold 16px sans-serif;
```

```
}  
.highlight{  
  background: yellow;  
}  
</style>  
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>  
<script>  
$(document).ready(function(){  
  $("p").click(function(){  
    $(this).toggleClass("highlight");  
  });  
});  
</script>  
</head>  
<body>  
  <p>Click on me to toggle highlighting.</p>  
  <p class="highlight">Click on me to toggle highlighting.</p>  
  <p>Click on me to toggle highlighting.</p>  
</body>  
</html>
```

---

# jQuery Get and Set CSS Properties

In this tutorial you will learn how to get or set style properties using jQuery.

## jQuery `css()` Method

The jQuery `css()` method is used to get the [computed value](#) of a CSS property or set one or more CSS properties for the selected elements.

This method provides a quick way to apply the styles directly to the HTML elements (i.e. [inline styles](#)) that haven't been or can't easily be defined in a stylesheet.

# Get a CSS Property Value

You can get the computed value of an element's CSS property by simply passing the property name as a parameter to the `css()` method. Here's the basic syntax:

```
$(selector).css("propertyName");
```

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>jQuery Get the Value of a CSS Property</title>
```

```
<style>
```

```
  div{
```

```
    width: 100px;
```

```
    height: 100px;
```

```
    margin: 10px;
```

```
    cursor: pointer;
```

```
    display: inline-block;
```

```
  }
```

```
</style>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<script>
```

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

```
  $("div").click(function(){
```

```
    var color = $(this).css("background-color");
```

```
    $("#result").html(color);
```

```
  });
```

```
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
  <div style="background-color:orange;"></div>
```

```
<div style="background-color:#ee82ee;"></div>
<div style="background-color:rgb(139,205,50);"></div>
<div style="background-color:#f00;"></div>
<p>The computed background-color property value of this DIV element is: <b id="result"></b></p>
</body>
</html>
```

---

## Set a Single CSS Property and Value

The `css()` method can take a property name and value as separate parameters for setting a single CSS property for the elements. The basic syntax can be given with:

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Set the Value of a CSS Property</title>
<style>
.box{
  width: 100px;
  height: 100px;
  margin: 10px;
  cursor: pointer;
  border: 1px solid #cdcdcd;
  display: inline-block;
}
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  $(".box").click(function(){
    $(this).css("background-color", "blue");
  });
});
```

```
});  
</script>  
</head>  
<body>  
  <div class="box"></div>  
  <div class="box"></div>  
  <div class="box"></div>  
  <div class="box"></div>  
  <p><strong>Note:</strong> Click inside the empty box to fill the background color.</p>  
</body>  
</html>
```

---

## Set Multiple CSS Properties and Values

You can also set multiple CSS properties with the `css()` method. The basic syntax for setting the more than one property for the elements can be given with:

```
$(selector).css({"propertyName":"value", "propertyName":"value", ...});
```

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
<meta charset="utf-8">  
<title>jQuery Set the Values of Multiple CSS Properties</title>  
<style>  
  p{  
    font-size: 18px;  
    font-family: Arial, sans-serif;  
  }  
</style>  
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>  
<script>  
$(document).ready(function(){
```

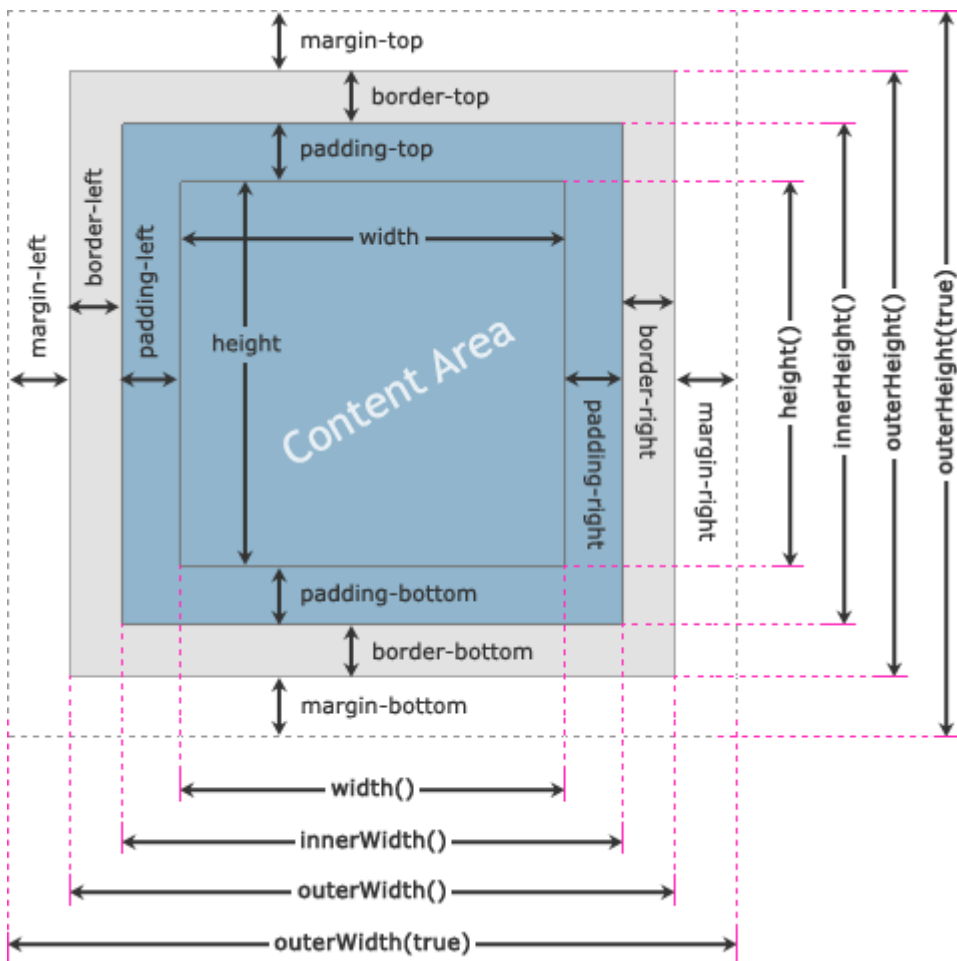
```
$("#button").click(function(){
    $("#p").css({"background-color": "yellow", "padding": "20px"});
});
});
</script>
</head>
<body>
    <h1>This is a heading</h1>
    <p style="background-color:orange;">This a paragraph.</p>
    <p style="background-color:#ee82ee;">This is another paragraph.</p>
    <p style="background-color:rgb(139,205,50);">This is none more paragraph.</p>
    <p>This is one last paragraph.</p>
    <button type="button">Add CSS Styles</button>
</body>
</html>
```

---

## Understanding the jQuery Dimensions

jQuery provides several methods, such

as `height()`, `innerHeight()`, `outerHeight()`, `width()`, `innerWidth()` and `outerWidth()` to get and set the CSS dimensions for the elements. Check out the following illustration to understand how these methods are calculating the dimensions of an element's box.



## jQuery `width()` and `height()` Methods

The jQuery `width()` and `height()` methods get or set the `width` and the `height` of the element respectively. This width and height doesn't include `padding`, `border` and `margin` on the element.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>jQuery Get Width and Height of an Element</title>
<style>
  #box{
    width: 300px;
```

```
height: 200px;

padding: 25px;

text-align: justify;

border: 10px solid #c6b51a;

background: #f0e68c;

margin: 15px;

}

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

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

        var divWidth = $("#box").width();

        var divHeight = $("#box").height();

        $("#result").html("Width: " + divWidth + ", " + "Height: " + divHeight);

    });

});

</script>

</head>

<body>

    <div id="box">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus
    dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum
    scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum
    vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla non purus nisl Dapibus nec
    turpis vel, semper malesuada ant.</div>

    <button type="button">Get Width and Height</button>

    <p id="result"></p>

</body>

</html>

<!DOCTYPE html>

<html lang="en">

<head>
```



```
<meta charset="utf-8">

<title>jQuery Set Width and Height of the Elements</title>

<style>

    #box{

        width: 300px;

        height: 200px;

        padding: 25px;

        text-align: justify;

        border: 10px solid #c6b51a;

        background: #f0e68c;

        margin: 15px;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

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

        $("#box").width(400).height(300);

    });

});

</script>

</head>

<body>

    <div id="box">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus
    dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum
    scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum
    vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla non purus nisl Dapibus nec
    turpis vel, semper malesuada ant.</div>

    <button type="button">Set Width and Height</button>

</body>

</html>
```

---

# jQuery `innerWidth()` and `innerHeight()` Methods

The jQuery `innerWidth()` and `innerHeight()` methods get or set the *inner width* and the *inner height* of the element respectively. This inner width and height includes the padding but excludes border and margin on the element. The following example will return the inner width and height of a `<div>` element on the click of a button.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Get Inner-Width and Inner-Height of an Element</title>

<style>

#box{

    width: 300px;

    height: 200px;

    padding: 25px;

    text-align: justify;

    border: 10px solid #c6b51a;

    background: #f0e68c;

    margin: 15px;

}

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

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

        var divWidth = $("#box").innerWidth();

        var divHeight = $("#box").innerHeight();

        $("#result").html("Inner Width: " + divWidth + ", " + "Inner Height: " + divHeight);

    });

});

</script>
```

```
</head>

<body>

  <div id="box">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus
  dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum
  scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum
  vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla non purus nisl Dapibus nec
  turpis vel, semper malesuada ant.</div>

  <button type="button">Get innerWidth and innerHeight</button>

  <p id="result"></p>

  <hr>

  <p><strong>Note:</strong> jQuery <b>innerWidth()</b> includes the CSS properties (<b>width</b> +
  <b>padding-left</b> + <b>padding-right</b>), whereas the <b>innerHeight()</b> includes (<b>height</b> +
  <b>padding-top</b> + <b>padding-bottom</b>).</p>

</body>

</html>
```

---

## jQuery `outerWidth()` and `outerHeight()` Methods

The jQuery `outerWidth()` and `outerHeight()` methods get or set the *outer width* and the *outer height* of the element respectively. This outer width and height includes padding and border but excludes the margin on the element. The following example will return the outer width and height of a `<div>` element on the click of a button.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>jQuery Get Outer-Width and Outer-Height of an Element</title>

<style>

  #box{

    width: 300px;

    height: 200px;

    padding: 25px;

    text-align: justify;

    border: 10px solid #c6b51a;
```

```
background: #f0e68c;

margin: 15px;

}

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

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

        var divWidth = $("#box").outerWidth();

        var divHeight = $("#box").outerHeight();

        $("#result").html("Outer Width: " + divWidth + ", " + "Outer Height: " + divHeight);

    });

});

</script>

</head>

<body>

    <div id="box">Lorem ipsum dolor sit amet, consectetur adipiscing elit. Nam eu sem tempor, varius quam at, luctus
    dui. Mauris magna metus, dapibus nec turpis vel, semper malesuada ante. Vestibulum id metus ac nisl bibendum
    scelerisque non non purus. Suspendisse varius nibh non aliquet sagittis. In tincidunt orci sit amet elementum
    vestibulum. Vivamus fermentum in arcu in aliquam. Quisque aliquam porta odio in fringilla non purus nisl Dapibus nec
    turpis vel, semper malesuada ant.</div>

    <button type="button">Get outerWidth and outerHeight</button>

    <p id="result"></p>

    <hr>

    <p><strong>Note:</strong> jQuery <b>outerWidth()</b> includes the CSS properties (<b>width</b> +
    <b>padding-left</b> + <b>padding-right</b> + <b>border-left</b> + <b>border-right</b>), whereas the
    <b>outerHeight()</b> includes (<b>height</b> + <b>padding-top</b> + <b>padding-bottom</b> + <b>border-top</b>
    + <b>border-bottom</b>).</p>

</body>

</html>
```

---

# What is Traversing

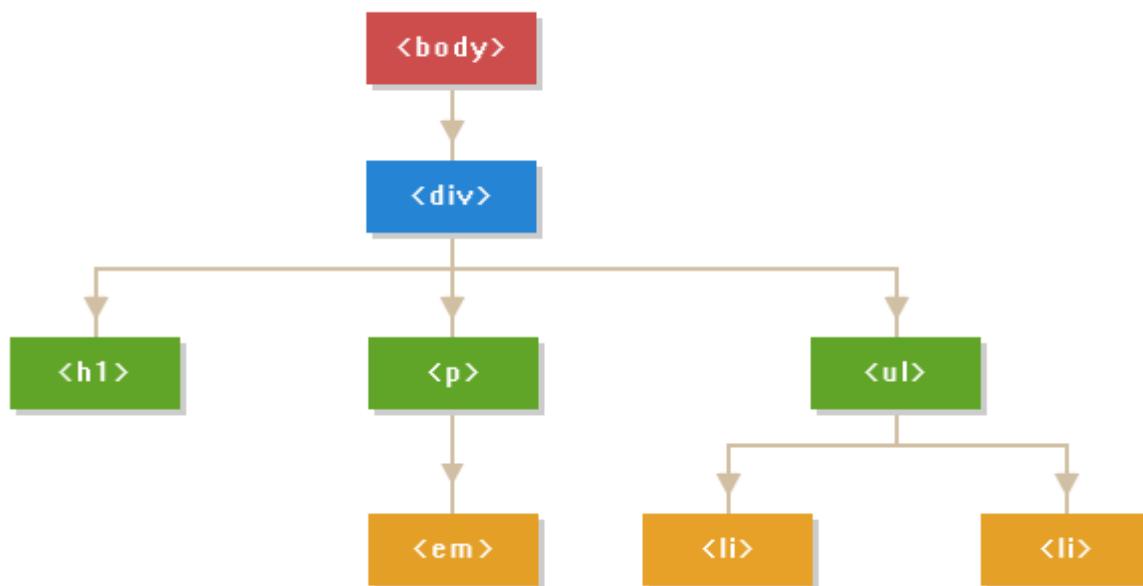
The jQuery selectors we've seen so far only allow us to select the elements down the DOM tree. But there are many occasions when you need to select a parent or ancestor element; that is where jQuery's DOM traversal methods come into play. With these traversal methods, we can go up, down, and all around the DOM tree very easily.

DOM traversing is one of the prominent features of the jQuery. To make the most it you need to understand the relationships between the elements in a DOM tree.

```
<!DOCTYPE html>
<html lang="en">
<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 example above can be represented by the following DOM 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**.
- 

## Traversing Up the DOM Tree

In logical relationships an ancestor is a parent, grandparent, great-grandparent, and so on.

jQuery provides the useful methods such as `parent()`, `parents()` and `parentsUntil()` that you can use to traverse up in the DOM tree either single or multiple levels to easily get the parent or other ancestors of an element in the hierarchy.

# jQuery `parent()` Method

The jQuery `parent()` method is used to get the direct parent of the selected element.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting the Direct Parent Element in jQuery</title>

<style>

    .highlight{

        background: yellow;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $("li").parent().addClass("highlight");

});

</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>
```

---

# jQuery `parents()` Method

The jQuery `parents()` method is used to get the ancestors of the selected element.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting All the Ancestor Elements in jQuery</title>

<style>

  *{

    margin: 10px;

  }

  .frame{

    border: 2px solid green;

  }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

  $("li").parents().addClass("frame");

});

</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>
```

---

## jQuery `parentsUntil()` Method

The jQuery `parentsUntil()` method is used to get all the ancestors up to but not including the element matched by the selector. In simple words we can say it returns all ancestor elements between two given elements in a DOM hierarchy.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting All the Ancestors between Two Elements in jQuery</title>
<style>
  *{
    margin: 10px;
  }
  .frame{
    border: 2px solid green;
  }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  $("li").parentsUntil("html").addClass("frame");
});
</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>
```

---

## jQuery `children()` Method

The jQuery `children()` method is used to get the direct children of the selected element.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting the Direct Child Elements in jQuery</title>

<style>

  .highlight{

    background: yellow;

  }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

  $("ul").children().addClass("highlight");

});

</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>
```

---

## jQuery `find()` Method

The jQuery `find()` method is used to get the descendant elements of the selected element.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting the Specific Descendant Elements in jQuery</title>

<style>

  *{

    margin: 10px;

  }

  .frame{

    border: 2px solid green;

  }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

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

```
    $("div").find("li").addClass("frame");
  });
</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>
```

---

## jQuery `siblings()` Method

The jQuery `siblings()` method is used to get the sibling elements of the selected element.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting All the Sibling Elements in jQuery</title>
<style>
  .highlight{
    background: yellow;
  }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
```

```
$(document).ready(function(){
    $("p").siblings().addClass("highlight");
});
</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>
```

---

## jQuery next() Method

The jQuery `next()` method is used to get the immediately following sibling i.e. the next sibling element of the selected element.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting the Next Sibling Element in jQuery</title>
<style>
    .highlight{
        background: yellow;
    }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<script>
$(document).ready(function(){
    $("p").next().addClass("highlight");
});
</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>
```

---

## jQuery nextAll() Method

The jQuery `nextAll()` method is used to get all following siblings of the selected element.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting All the Following Sibling Elements in jQuery</title>
<style>
    .highlight{
        background: yellow;
    }
</style>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("p").nextAll().addClass("highlight");
});
</script>
</head>
<body>
    <div class="container">
        <h1>Hello World</h1>
        <p>This is a <em>simple paragraph</em>.</p>
        <p>This is another paragraph.</p>
        <ul>
            <li>Item One</li>
            <li>Item Two</li>
        </ul>
    </div>
</body>
</html>
```

---

## jQuery nextUntil() Method

The jQuery `nextUntil()` method is used to get all the following siblings up to but not including the element matched by the selector. In simple words we can say it returns all the next siblings elements between two given elements in a DOM hierarchy.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting All Following Siblings between Two Elements in jQuery</title>
<style>
    .highlight{
```

```
        background: yellow;
    }
</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>
$(document).ready(function(){
    $("h1").nextUntil("ul").addClass("highlight");
});
</script>
</head>
<body>
    <div class="container">
        <h1>Hello World</h1>
        <p>This is a <em>simple paragraph</em>.</p>
        <p>This is another paragraph.</p>
        <ul>
            <li>Item One</li>
            <li>Item Two</li>
        </ul>
    </div>
</body>
</html>
```

---

## jQuery prev() Method

The jQuery `prev()` method is used to get the immediately preceding sibling i.e. the previous sibling element of the selected element.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting the Previous Sibling Element in jQuery</title>
```



```
<style>
    .highlight{
        background: yellow;
    }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("ul").prev().addClass("highlight");
});
</script>
</head>
<body>
    <div class="container">
        <h1>Hello World</h1>
        <p>This is a <em>simple paragraph</em>.</p>
        <p>This is another paragraph.</p>
        <ul>
            <li>Item One</li>
            <li>Item Two</li>
        </ul>
    </div>
</body>
</html>
```

---

## jQuery prevAll() Method

The jQuery `prevAll()` method is used to get all preceding siblings of the selected element.

```
<!DOCTYPE html>
<html lang="en">
<head>
```

```
<meta charset="utf-8">
<title>Selecting All the Preceding Sibling Elements in jQuery</title>
<style>
    .highlight{
        background: yellow;
    }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("ul").prevAll().addClass("highlight");
});
</script>
</head>
<body>
    <div class="container">
        <h1>Hello World</h1>
        <p>This is a <em>simple paragraph</em>.</p>
        <p>This is another paragraph.</p>
        <ul>
            <li>Item One</li>
            <li>Item Two</li>
        </ul>
    </div>
</body>
</html>
```

---

## jQuery `prevUntil()` Method

The jQuery `prevUntil()` method is used to get all the preceding siblings up to but not including the element matched by the selector. In simple words we can say it returns all the previous siblings elements between two given elements in a DOM hierarchy.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting All Preceding Siblings between Two Elements in jQuery</title>

<style>

    .highlight{

        background: yellow;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $("ul").prevUntil("h1").addClass("highlight");

});

</script>

</head>

<body>

    <div class="container">

        <h1>Hello World</h1>

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

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

        <ul>

            <li>Item One</li>

            <li>Item Two</li>

        </ul>

    </div>

</body>

</html>
```

---

## Filtering the Elements Selection

jQuery provides several methods such as `filter()`, `first()`, `last()`, `eq()`, `slice()`, `has()`, `not()`, etc. that you can use to narrow down the search for elements in a DOM tree.

## jQuery `first()` Method

The jQuery `first()` method filters the set of matched elements and returns the first element from the set.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting the First Element in jQuery</title>

<style>

    .highlight{

        background: yellow;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $("ul li").first().addClass("highlight");

});

</script>

</head>

<body>

    <h2>Unordered List</h2>

    <ul>

        <li>First list item</li>

        <li>Second list item</li>

        <li>Third list item</li>

        <li>Fourth list item</li>

    </ul>
```

```
<hr>
<h2>Another Unordered List</h2>
<ul>
<li>First list item</li>
<li>Second list item</li>
<li>Third list item</li>
<li>Fourth list item</li>
</ul>
</body>
</html>
```

---

## jQuery `last()` Method

The jQuery `last()` method filters the set of matched elements and returns the last element from the set.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting the Last Element in jQuery</title>
<style>
.highlight{
  background: yellow;
}
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  $("ul li").last().addClass("highlight");
});
</script>
</head>
```

```
<body>

    <h2>Unordered List</h2>

    <ul>

        <li>First list item</li>

        <li>Second list item</li>

        <li>Third list item</li>

        <li>Fourth list item</li>

    </ul>

    <hr>

    <h2>Another Unordered List</h2>

    <ul>

        <li>First list item</li>

        <li>Second list item</li>

        <li>Third list item</li>

        <li>Fourth list item</li>

    </ul>

</body>
</html>
```

---

## jQuery eq() Method

The jQuery `eq()` method filters the set of matched elements and returns only one element with a specified index number.

```
<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting an Element by Index in jQuery</title>

<style>

    .highlight{

        background: yellow;

    }

</style>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
    $("ul li").eq(1).addClass("highlight");
});
</script>
</head>
<body>
    <h2>Unordered List</h2>
    <ul>
        <li>First list item</li>
        <li>Second list item</li>
        <li>Third list item</li>
        <li>Fourth list item</li>
    </ul>
    <hr>
    <h2>Another Unordered List</h2>
    <ul>
        <li>First list item</li>
        <li>Second list item</li>
        <li>Third list item</li>
        <li>Fourth list item</li>
    </ul>
</body>
</html>
```

---

## jQuery `filter()` Method

The jQuery `filter()` method can take the selector or a function as its argument to filters the set of matched elements based on a specific criteria.

```
<!DOCTYPE html>
<html lang="en">
```

```
<head>

<meta charset="utf-8">

<title>Filtering the Selection of Elements in jQuery via Selectors</title>

<style>

    .highlight{

        background: yellow;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $("ul li").filter(":even").addClass("highlight");

});

</script>

</head>

<body>

    <h2>Unordered List</h2>

    <ul>

        <li>First list item</li>

        <li>Second list item</li>

        <li>Third list item</li>

        <li>Fourth list item</li>

    </ul>

    <hr>

    <h2>Another Unordered List</h2>

    <ul>

        <li>First list item</li>

        <li>Second list item</li>

        <li>Third list item</li>

        <li>Fourth list item</li>

    </ul>

</body>
```



```
</body>
```

```
</html>
```

---

```
<!DOCTYPE html>
```

```
<html lang="en">
```

```
<head>
```

```
<meta charset="utf-8">
```

```
<title>Filtering the Selection of Elements in jQuery via Function</title>
```

```
<style>
```

```
  .highlight{  
    background: yellow;  
  }
```

```
</style>
```

```
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
```

```
<script>
```

```
$(document).ready(function(){  
  $("ul li").filter(function(index){  
    return index % 2 !== 0;  
  }).addClass("highlight");  
});
```

```
</script>
```

```
</head>
```

```
<body>
```

```
  <h2>Unordered List</h2>
```

```
  <ul>
```

```
    <li>First list item</li>
```

```
    <li>Second list item</li>
```

```
    <li>Third list item</li>
```

```
    <li>Fourth list item</li>
```

```
  </ul>
```

```
    <hr>
```

```
    <h2>Another Unordered List</h2>
```

```
<ul>

<li>First list item</li>

<li>Second list item</li>

<li>Third list item</li>

<li>Fourth list item</li>

</ul>

</body>

</html>
```

---

## jQuery has() Method

The jQuery `has()` method filters the set of matched elements and returns only those elements that has the specified descendant element.

```
<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<title>Selecting Elements that Contain Specific Child Elements in jQuery</title>

<style>

    .highlight{

        background: yellow;

    }

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

$(document).ready(function(){

    $("ul li").has("ul").addClass("highlight");

});

</script>

</head>

<body>

<ul>
```

```
<li>Section 1</li>
<li>Section 2</li>
<li>
  <ul>
    <li>Section 2.1</li>
    <li>Section 2.2</li>
    <li>Section 2.3</li>
  </ul>
</li>
<li>Section 4</li>
</ul>
</body>
</html>
```

---

## jQuery `not()` Method

The jQuery `not()` method filters the set of matched elements and returns all elements that does not met the specified conditions. It can take the selector or a function as its argument.

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>Selecting Elements that Doesn't Match a Condition in jQuery</title>
<style>
  .highlight{
    background: yellow;
  }
</style>
<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
<script>
$(document).ready(function(){
  $("ul li").not(":even").addClass("highlight");
```

```
});  
</script>  
</head>  
<body>  
    <h2>Unordered List</h2>  
  
    <ul>  
        <li>First list item</li>  
        <li>Second list item</li>  
        <li>Third list item</li>  
        <li>Fourth list item</li>  
    </ul>  
  
    <hr>  
    <h2>Another Unordered List</h2>  
  
    <ul>  
        <li>First list item</li>  
        <li>Second list item</li>  
        <li>Third list item</li>  
        <li>Fourth list item</li>  
    </ul>  
</body>  
</html>
```

---

## jQuery `slice()` Method

The jQuery `slice()` method filters the set of matched elements specified by a range of indices. This method accepts *start* and *end* (optional) index number as arguments, where the start index specifies the position at which the elements begin to be selected and the end index specify the position at which the elements stop being selected.

```
<!DOCTYPE html>  
  
<html lang="en">  
  
<head>  
  
<meta charset="utf-8">
```

<title>Selecting the Elements by Range of Indices in jQuery</title>

<style>

```
.highlight{  
    background: yellow;  
}
```

</style>

<script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>

<script>

```
$(document).ready(function(){  
    $("ul li").slice(0, 2).addClass("highlight");  
});
```

</script>

</head>

<body>

<h2>Unordered List</h2>

<ul>

```
<li>First list item</li>  
<li>Second list item</li>  
<li>Third list item</li>  
<li>Fourth list item</li>
```

</ul>

<hr>

<h2>Another Unordered List</h2>

<ul>

```
<li>First list item</li>  
<li>Second list item</li>  
<li>Third list item</li>  
<li>Fourth list item</li>
```

</ul>

</body>

</html>

---