TOPIC:

- What is Jqury?, Use of Jquery in Web Designing, Adding Jquery in Your page.
- Jquery Syntax, Events in Jquery
- JQuery Functions: hide(), show(), toggle(),fadeIn(), fadeOut(), fadeToggle(), fadeTo().
- JQuery Sliding Method: slideDown(), slideUp(), slideToggle(),animate(), Stop().
- Add Element, Remove Element, Add Class and Remove Class.

What is Jqury? Use of Jquery in Web Designing, Adding Jquery in Your page.

- ✓ jQuery is a lightweight, "write less, do more", JavaScript library.
- ✓ The purpose of jQuery is to make it much easier to use JavaScript on your website.
- ✓ jQuery takes a lot of common tasks that require many lines of JavaScript code to accomplish, and wraps them into methods that you can call with a single line of code.
- ✓ jQuery also simplifies a lot of the complicated things from JavaScript, like AJAX calls and DOM manipulation.

The jQuery library contains the following features:

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

Tip: In addition, jQuery has plugins for almost any task out there.

Why we use jQuery in our web application

- ✓ jQuery is a fast, feature-rich, and lightweight JavaScript library. The main purpose of using jQuery is to make it much easier to use JavaScript on your modern and smart website. It is highly recommended to have a basic knowledge of HTML, CSS, and JavaScript.
- ✓ jQuery was developed to save the time of developers by reducing the code. It takes loads of common duties that require many lines of JavaScript code to perform and wrap them into strategies that you may name with a single line of code.
- ✓ Reasons to use jQuery in your application:
- ✓ Easy to understand: It has simpler code than JavaScript. So you just have to write few lines of code to do the same thing. In addition, builders ought not to be professionals in programming or web layout to create incredible patterns for their sites. Any developer who has spent hours coding and trying out CSS documents will recognize the easy implementation that jQuery brings to the table. There's additionally a set of strong jQuery UI additives that builders can plug into their websites.
- ✓ You might be able to understand by referring to the below example.

JavaScript Code Snippet:

```
function changeColor(color) {

   document.body.style.background = color;

}

Onload = changeColor('green');
```

jQuery Syntax:

```
$('body').css('background', 'green');
```

- ✓ From the above example, you can notice that the JavaScript code is lengthier and complicated than the jQuery code. Both the code are performing the same work of changing the background color but jQuery takes less code. You can work around other examples, which indicates that jQuery minimizes the code and is less complicated to use.
- ✓ Easily Integrated with other IDE: Most .Net developers use Visual Studio and are acquainted with NuGet. This is a part of the purpose why jQuery's recognition keeps developing with .Net developers. With the addition of the jQuery cell difficulty for Windows, you currently have all the development benefits of the jQuery library for the Windows Phone platform.
- ✓ Animation becomes easy: jQuery makes use of CSS, HTML, JavaScript, and AJAX. In this method, you could practice an optimization approach on your website online while not having to make unique changes for technologies like Flash. You can attain great-searching outcomes as a way to maintain your audience engaged.
- ✓ Faster: Many search engines are considering page load time as one of the main factors because it affects SEO. For this reason, every developer in today's world wants to make codes as concise as possible. The best way to make your website faster is by writing less code and it is possible by using the simplest JavaScript library called jQuery.
- ✓ SEO friendly: SEO stand for search engine optimization and it is the process of improving the quality and quantity of website traffic to a website or a web page from search engines. So many popular search engines like google, bing and yahoo use SEO. jQuery may be optimized for search engines, and there are a lot of plug-ins for developers.
- ✓ Run in all major browsers: The team behind the jQuery library knows what are the main issues that normally occurs in all major browsers. So they have developed this library to ease the developer's work.

Adding Jquery in Your page

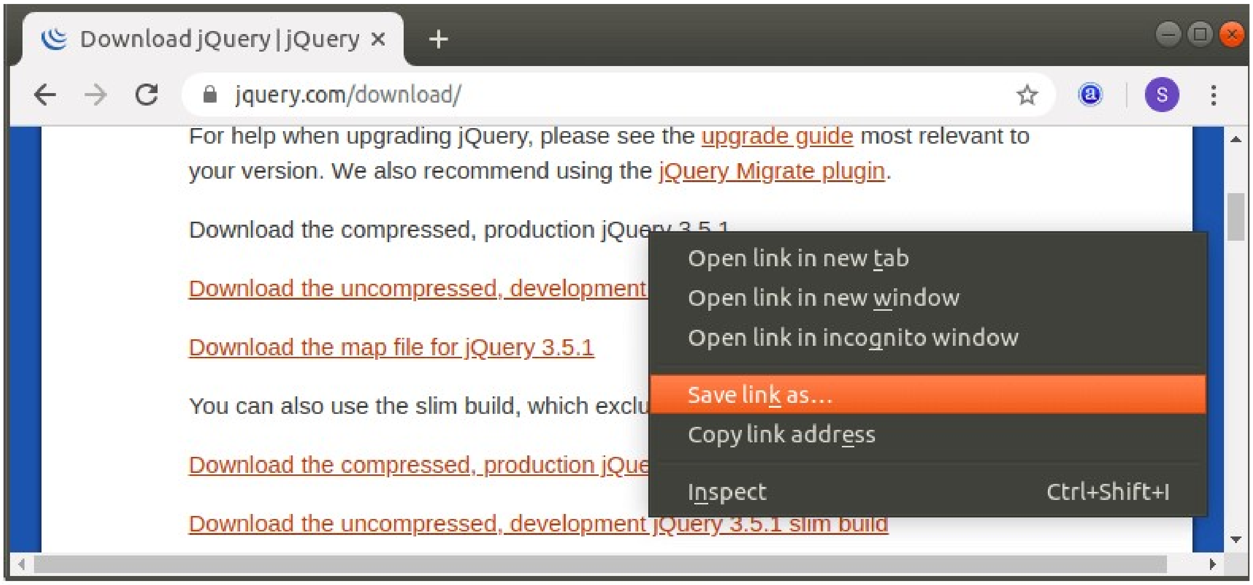
Following are the two different ways for adding the jQuery to Html page:

1. Download and Include jQuery file
2. Include the jQuery by CDN.

Download and Include jQuery File

If we want to add the jQuery to Html page by downloading the jQuery file, then we have to follow the steps which are given below. Using these steps, any user can easily add the jQuery.

Step 1: Firstly, we have to download the jquery js file from the following official site of jQuery. https://jquery.com/download/



For help when upgrading jQuery, please see the upgrade guide most relevant to your version. We also recommend using the jQuery Migrate plugin.

Download the compressed, production jQuery 3.5.1

Download the uncompressed, development

Download the map file for jQuery 3.5.1

You can also use the slim build, which exclu

Download the compressed, production jQue

Download the uncompressed, development jQuery 3.5.1 slim build

Step 2: When we have downloaded the file, then we have to open that Html file in which we want to add the jquery.

Step 3: After then, we have to place the cursor between the head tag just before the title tag. And, then we have to use the <script> tag, which specify the src attribute for adding the jQuery file.

```
<!Doctype Html>
<Html>
<Head>
<script type="text/javascript" src="jquery-3.5.1.min.js">
</script>
<Title>
Add the jQuery file into Html by downloading and Including file
</Title>
</Head>
<Body>
Hello User!... <br> <center>
<img src="https://www.javatpoint.com/images/logo/jtp_logo.png" width="100" height="100"
> </center>
</Body>
</Html>
```

Step 4: And, at last, save the Html file and the jQuery file is successfully added into our Html page.

Include the jQuery by CDN

If we want to add the jQuery to Html page by the help of CDN, then we have to follow the steps which are given below. Using these steps, any user can easily add the jQuery.

Step 1: Firstly, we have to open that Html file in which we want to add the jQuery using CDN.

Step 2: After then, we have to place the cursor between the head tag just before the title tag. And, then we have to use the <script> tag, which specify the src attribute for adding.

Step 3: And then, we have to type the following path in the src attribute.

http://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js

Step 4: And, at last, save the Html file and the jQuery file is successfully added into our Html page.

```
<!Doctype Html>
<Html>
<Head>
<script type="text/javascript" src="http://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js" >
</script>
<Title>
Add the jQuery file into Html by including CDN
</Title>
</Head>
<Body>
Hello User!... <br> <center>
<img src="https://www.javatpoint.com/images/logo/jtp_logo.png" width="100" height="100" > </center>
</Body>
</Html>
```

Jquery Syntax, Events in Jquery

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

Basic syntax is: **$(*selector*).*action*()**

- A $ sign to define/access jQuery
- A (*selector*) to "query (or find)" HTML elements
- A jQuery *action*() to be performed on the element(s)

Examples:

$(this).hide() - hides the current element.

$("p").hide() - hides all <p> elements.

$(".test").hide() - hides all elements with class="test".

$("#test").hide() - hides the element with id="test".

The Document Ready Event

You might have noticed that all jQuery methods in our examples, are inside a document ready event:

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

});
```

- ✓ This is to prevent any jQuery code from running before the document is finished loading (is ready).
- ✓ It is good practice to wait for the document to be fully loaded and ready before working with it. This also allows you to have your JavaScript code before the body of your document, in the head section.
- ✓ Here are some examples of actions that can fail if methods are run before the document is fully loaded:

- Trying to hide an element that is not created yet
- Trying to get the size of an image that is not loaded yet

Tip: The jQuery team has also created an even shorter method for the document ready event:

```
$(function(){

  // jQuery methods go here...

});
```

Use the syntax you prefer. We think that the document ready event is easier to understand when reading the code.

jQuery Selectors

- ✓ jQuery selectors allow you to select and manipulate HTML element(s).
- ✓ jQuery selectors are used to "find" (or select) HTML elements based on their name, id, classes, types, attributes, values of attributes and much more. It's based on the existing CSS Selectors, and in addition, it has some own custom selectors.
- ✓ All selectors in jQuery start with the dollar sign and parentheses: $().

The element Selector

- ✓ The jQuery element selector selects elements based on the element name.
- ✓ You can select all <p> elements on a page like this:
- ✓ $("p")

Example

When a user clicks on a button, all <p> elements will be hidden:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script>
</head>
<body>
```

```
      <h2>This is a heading</h2>

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

      <button>Click me to hide paragraphs</button>

      </body>
</html>
```

The #id Selector

- ✓ The jQuery *#id* selector uses the id attribute of an HTML tag to find the specific element.
- ✓ An id should be unique within a page, so you should use the #id selector when you want to find a single, unique element.
- ✓ To find an element with a specific id, write a hash character, followed by the id of the HTML element:
- ✓ $("#test")

Example

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#test").hide();
  });
});
</script>
</head>
<body>

<h2>This is a heading</h2>

<p>This is a paragraph.</p>
<p id="test">This is another paragraph.</p>

<button>Click me</button>

</body>
</html>
```

This is a heading

This is a paragraph.
This is another paragraph.
Click me

The .class Selector

The jQuery *.class* selector finds elements with a specific class.
To find elements with a specific class, write a period character, followed by the name of the class:

```
$(".test")
```

Example

When a user clicks on a button, the elements with class="test" will be hidden:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $(".test").hide();
  });
});
</script>
</head>
<body>

<h2 class="test">This is a heading</h2>

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

<button>Click me</button>

</body>
</html>
```

✓ Events in Jquery

All the different visitors' actions that a web page can respond to are called events.
An event represents the precise moment when something happens.
Examples:

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

The term **"fires/fired"** is often used with events. Example: "The keypress event is fired, the moment you press a key".
Here are some common DOM events:

Mouse Events	**Keyboard Events**	**Form Events**	**Document/Window Events**
click	keypress	submit	load

dblclick	keydown	change	resize
mouseenter	keyup	focus	scroll
mouseleave		blur	unload

✓ **jQuery Syntax For Event Methods**

- ✓ In jQuery, most DOM events have an equivalent jQuery method.
- ✓ To assign a click event to all paragraphs on a page, you can do this:
- ✓ $("p").click();
- ✓ The next step is to define what should happen when the event fires. You must pass a function to the event:

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

✓ **Commonly Used jQuery Event Methods**

- ✓ **$(document).ready()**
- ✓ The $(document).ready() method allows us to execute a function when the document is fully loaded. This event is already explained in the jQuery Syntax chapter.

1. **click()**

- ✓ The click() method attaches an event handler function to an HTML element.
- ✓ The function is executed when the user clicks on the HTML element.
- ✓ The following example says: When a click event fires on a <p> element; hide the current <p> element:

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").click(function(){
    $(this).hide();
  });
});
</script>
</head>
<body>
<p>If you click on me, I will disappear.</p>
<p>Click me away!</p>
<p>Click me too!</p>
</body>
</html>
```

2. dblclick()

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

✓ The function is executed when the user double-clicks on the HTML element:

Example:

```
    <!DOCTYPE html>
    <html>
    <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script>
    $(document).ready(function(){
      $("p").dblclick(function(){
        $(this).hide();
      });
    });
    </script>
    </head>
    <body>

    <p>If you double-click on me, I will disappear.</p>
    <p>Click me away!</p>
    <p>Click me too!</p>

    </body>
</html>
```

3. mouseenter()

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

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

Example:

```
    <!DOCTYPE html>
    <html>
    <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script>
    $(document).ready(function(){
      $("#p1").mouseenter(function(){
        alert("You entered p1!");
      });
    });
    </script>
    </head>
    <body>
    <p id="p1">Enter this paragraph.</p>
    </body>
</html>
```

4. mouseleave()

- ✓ The mouseleave() method attaches an event handler function to an HTML element.
- ✓ The function is executed when the mouse pointer leaves the HTML element:

Example:

```
    <!DOCTYPE html>
    <html>
    <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script>
    $(document).ready(function(){
      $("#p1").mouseleave(function(){
        alert("Bye! You now leave p1!");
      });
    });
    </script>
    </head>
    <body>

    <p id="p1">This is a paragraph.</p>

    </body>
</html>
```

5. mousedown()

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

Example:

```
    <!DOCTYPE html>
    <html>
    <head>
    <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
    <script>
    $(document).ready(function(){
      $("#p1").mousedown(function(){
        alert("Mouse down over p1!");
      });
    });
    </script>
    </head>
    <body>

    <p id="p1">This is a paragraph.</p>

    </body>
</html>
```

6. mouseup()

- ✓ The mouseup() method attaches an event handler function to an HTML element.
- ✓ The function is executed, when the left, middle or right mouse button is released, while the mouse is over the HTML element:

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").mouseup(function(){
    alert("Mouse up over p1!");
  });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

7. hover()

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").hover(function(){
    alert("You entered p1!");
  }, function(){
    alert("Bye! You now leave p1!");
  });
});
</script>
</head>
<body>
<p id="p1">This is a paragraph.</p>
</body>
</html>
```

8. focus()

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

Example:

```
        <!DOCTYPE html>
        <html>
        <head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script>
        $(document).ready(function(){
          $("input").focus(function(){
            $(this).css("background-color", "yellow");
          });
          $("input").blur(function(){
            $(this).css("background-color", "green");
          });
        });
        </script>
        </head>
        <body>

        Name: <input type="text" name="fullname"><br>
        Email: <input type="text" name="email">

        </body>
</html>
```

9. blur()

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("input").focus(function(){
    $(this).css("background-color", "yellow");
  });
  $("input").blur(function(){
    $(this).css("background-color", "green");
  });
});
</script>
</head>
```

```
        <body>

        Name: <input type="text" name="fullname"><br>
        Email: <input type="text" name="email">

        </body>
</html>
```

10. The on() Method

- ✓ The on() method attaches one or more event handlers for the selected elements.
- ✓ Attach a click event to a <p> element:

Example:

```
        <!DOCTYPE html>
        <html>
        <head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script>
        $(document).ready(function(){
          $("p").on("click", function(){
            $(this).hide();
          });
        });
        </script>
        </head>
        <body>

        <p>If you click on me, I will disappear.</p>
        <p>Click me away!</p>
        <p>Click me too!</p>

        </body>
</html>
```

Attach multiple event handlers to a <p> element:

```
        <!DOCTYPE html>
        <html>
        <head>
        <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
        <script>
        $(document).ready(function(){
          $("p").on({
            mouseenter: function(){
              $(this).css("background-color", "lightgray");
            },
            mouseleave: function(){
              $(this).css("background-color", "lightblue");
            },
            click: function(){
              $(this).css("background-color", "yellow");
            }
          });
        });
        </script>
        </head>
        <body>
        <p>Click or move the mouse pointer over this paragraph.</p>
        </body>
        </html>
```

❖ JQuery Functions

1. hide():

The **hide()** method in jQuery is used to hide the selected elements by animating their opacity and dimensions. When called, the hide() method animates the opacity and dimensions of the selected elements to zero over a specified duration, making them visually disappear. The elements are still present in the DOM but are they'll not visible to the user.
This method works similar to the CSS property "display:none".

*To display the hidden elements on the DOM, we need to use the **show()** method.*

Syntax

Following is the syntax of hide() method in jQuery

```
$(selector).hide(speed,easing,callback)
```

Parameters

This method accepts the following optional parameters −

- **speed (optional):** A string or number determining how long the animation will run. Default is "400" (milliseconds).
- **easing (optional):** A string specifying the easing function to use for the transition. Default is "swing".
- **callback (optional):** A function to call once the animation is complete.

Example:

```
<html>
<head>
  <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
  <script>
    $(document).ready(function(){
        $("#hideButton").click(function(){
            $("#content").hide();
        });
    });
  </script>
</head>
<body>
```

```
   <button id="hideButton">Hide Content</button>
   <div id="content">
      <p>Click the above button to hide this content.</p>
   </div>
</body>
</html>
```

When the button is clicked, the hide() method hides the <div> element.

```
<html>
<head>
   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
   <script>
      $(document).ready(function(){
         $("#hideButton").click(function(){
            $("#content").hide(1000);
         });
      });
   </script>
</head>
<body>
   <button id="hideButton">Hide Content</button>
   <div id="content">
      <p>Click the above button to hide this content.</p>
   </div>
</body>
</html>
```

Output:

Hide Content

Click the above button to hide this content

Example 3

The following example uses the jQuery's hide() method with a callback function to execute after the content is hidden

```
<html>
<head>
   <script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
   <script>
      $(document).ready(function(){
         $("#hideButton").click(function(){
            $("#content").hide("slow", function(){
               alert("Content is now hidden.");
            });
         });
```

```
        });
    </script>
</head>
<body>
    <button id="hideButton">Hide Content</button>
    <div id="content">
        <p>This is the content to be hidden with callback function.</p>
    </div>
</body>
</html>
```

2. show():

The show() method in jQuery is used to display hidden elements by animating them to become visible. It is used to display elements that have been hidden using jQuery's hide() method or CSS properties.

If an element is already visible, the show() method has no effect on it. The animation can be customized using options such as duration, easing, and callback functions.

Syntax

Following is the syntax of show() method in jQuery

$(selector).hide(speed,easing,callback)

Parameters

This method accepts the following optional parameters −

- speed (optional): A string or number determining how long the animation will run. Default value is 400 milliseconds. Possible values are: milliseconds, slow, fast.
- easing (optional): A string indicating which easing function to use for the transition. Default value is "swing". Possible values are: swing, linear.
- callback (optional): A function to call once the animation is complete.

Example 1

- In the following example, we are using the jQuery's show() method to display the hidden element <div>

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function(){
    $("#showBtn").click(function(){
        $("#hiddenElement").show();
    });
});
</script>
</head>
<body>
<button id="showBtn">Show Element</button>
<div id="hiddenElement" style="display:none;">
    <h3>This is a hidden element. It will be shown when the button is clicked.</h3>
</div>
</body>
</html>
```

When we click on the button, the show() method displays the hidden element <div>.

Example 2

The following example displays the two hidden elements when the page is loaded

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function(){
  $(".hidden").show();
});
</script>
</head>
<body>
<div class="hidden" style="display:none;">
  <h3>Hidden element 1</h3>
</div>
<div class="hidden" style="display:none;">
  <h3>Hidden element 2</h3>
</div>
</body>
</html>
```

After executing the above program, it shows the hidden elements <div> with id = hidden on to the DOM

Example 3

The following example shows the hidden elements with animation

```
. <html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function(){
  $("#showBtn").click(function(){
    $("#hiddenElement").show("slow");
  });
});
</script>
</head>
<body>
<button id="showBtn">Show Element with Animation</button>
<div id="hiddenElement" style="display:none;">
  <h3>This is a hidden element. It will be shown with animation when the button is clicked.</h3>
</div>
</body>
</html>
```

After clicking the 'Show Element with Animation' button, it triggers a smooth animation that displays the hidden elements.

3. **toggle():**

The jQuery event toggle() method is used to toggle the visibility of selected elements and automatically switch between the .hide() and .show() methods within a second (default).
This method accepts an optional parameter using which you can handle the toggle speed (hide or show effect), just pass the value of the 'speed' parameter according to that it handles the toggle speed.

Syntax

Following is the syntax of the jQuery event toggle() method

```
$(selector).toggle(speed, easing, callback);
```

Parameters

This method accepts three parameters 'speed', 'easing', and 'callback , which are described below

- **speed** − It Specifies the speed of the hide or show effect (values: milliseconds, 'slow', 'fast').
- **easing** − It Specifies the easing function for the transition (default is 'swing').
- **callback** − A function to run after the toggle effect finished.

Return Value

- This method does not have any return value.

Example 1

- The following is the basic example of the jQuery event **toggle()** method

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js"></script>
</head>
<body>
   <div class="">Toggle event occured</div>
   <script>
      $('div').toggle(function(){
         alert("Toggle event triggered");
      });
   </script>
</body>
</html>
```

Output

The above program displayed a message, and automatically hide when the toggle event triggered and a pop-up alert appear on the browser screen

Example 2

Using an optional parameter named 'speed' to handle the speed of toggle effect

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js"></script>
<style>
   div{
      width: 200px;
      padding: 10px;
      background-color: green;
      color: white;
```

```
        border-radius: 10px;
    }
</style>
</head>
<body>
    <div class="">Toggle event occured</div>
    <span></span>
    <script>
        $('div').toggle(3000, function(){
            $('span').text("Hidden");
        });
    </script>
</body>
</html>
```

Output

After executing the above program, a message with a green background will be displayed. It will then automatically begin to hide, and within 3 seconds, it will be completely hidden

Example 3

Let use the "slow" and "fast" value with speed so see the toggle effect

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/1.7/jquery.min.js"></script>
<style>
    div{
        width: 200px;
        padding: 10px;
        background-color: green;
        color: white;
        border-radius: 10px;
    }
    .fst{
        width: 200px;
        padding: 10px;
        background-color: red;
        color: white;
        border-radius: 10px;
    }
</style>
</head>
<body>
    <p>Slow value example</p>
    <div class="slw">Toggle event occured(with slow value)</div>
    <p>Fast value example</p>
    <div class="fst">Toggle event occured(with slow value)</div>
    <span></span>
    <script>
```

```
        $('.slw').toggle((5000, "slow"), function(){
           $('span').text("Hidden");
        });
        $('.fst').toggle((5000, "fast"), function(){
           $('span').text("Hidden");
        });
    </script>
</body>
</html>
```

Output

Once the above program is executed, two div elements with green and red backgrounds will be displayed. The first div will be hidden within 5 seconds with a slow effect, and the second div will be hidden within 5 seconds with a fast effect

4. **fadeIn():**

The fade-in effect is a visual transition where an element gradually becomes more visible from being initially hidden or transparent. It smoothly increases the opacity of the element over a specified duration, creating a smooth transition.

The **fadeIn()** method is used in jQuery to make a hidden element visible by fading them in over a specified duration. It animates the opacity of the selected elements from 0 to 1, making them visible.

Syntax

Following is the syntax of fadeIn() method in jQuery

```
$(selector).fadeIn(speed,easing,callback)
```

Parameters

This method accepts the following optional parameters −

- **speed (optional):** A string or number determining how long the animation will run. It can take values like "slow", "fast", or a specific duration in milliseconds (e.g., 1000 for 1 second).
- **easing (optional):** A string specifying the easing function to use for the animation (e.g., "swing" or "linear").
- **callback (optional):** A function to call once the animation is complete. It's executed for each selected element.

Example 1

- In the following example, we are using the JavaScript fadeIn() method to add a "slow" fade in effect on a <div> element

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
   $("#fadeButton").click(function() {
      $("#content").fadeIn("slow");
   });
});
</script>
```

```
</head>
<body>
<button id="fadeButton">Click me!</button>
<div id="content" style="display: none;">
   <h2>Welcome to Tutorialspoint</h2>
   <p>This text will fade in when the button is clicked.</p>
</div>
</body>
</html>
```

If we click on the button, the <div> element will fade in.

Example 2

In this example, the <div> element automatically fades in when the page loads

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
   $("#content").fadeIn("slow");
});
</script>
</head>
<body>

<div id="content" style="display: none;">
   <h2>Welcome to Tutorialspoint!</h2>
   <p>This content fades in automatically when the page loads.</p>
</div>
</body>
</html>
```

If we execute the above program, the content of <div> element will fade in automatically.

Example 3

The following example fades in multiple elements one after the other

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
   $("#fadeButton").click(function() {
      $(".content").each(function(index) {
         $(this).delay(500 * index).fadeIn("slow");
      });
   });
});
</script>
</head>
<body>
```

```
<button id="fadeButton">Click me!</button>
<div class="content" style="display: none;">
   <h2>First Element</h2>
</div>
<div class="content" style="display: none;">
   <h2>Second Element</h2>
</div>
<div class="content" style="display: none;">
   <h2>Third Element</h2>
</div>
</body>
</html>
```

If we click on the button, all the three <div> elements will fade in.

Example 4

In the below example, we are using both fadeIn() and fadeOut() methods together to create a simple toggle effect

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function() {
   $("#toggleButton").click(function() {
      $("#content").fadeOut("slow", function() {
         // Fade out complete callback
         $("#content").fadeIn("slow");
      });
   });
});
</script>
</head>
<body>
<button id="toggleButton">Click me!</button>
<div id="content">
   <h2>Welcome to Tutorialspoint!</h2>
</div>
</body>
</html>
```

The <div> element will fade out and then fade back in when the button is clicked.

5. fadeOut()

The fade-out effect is a visual transition where an element gradually becomes hidden or transparent from being visible. It smoothly decreases the opacity of the element over a specified duration, creating a smooth transition.

The **fadeOut()** method is used in jQuery to gradually reduce the opacity of selected elements, simply hiding them by fading them out. It provides a smooth visual transition effect, making elements disappear gradually.

Syntax

Following is the syntax of fadeOut() method in jQuery −

```
$(selector).fadeOut(speed,easing,callback)
```

Parameters

This method accepts the following optional parameters −

- **speed (optional):** Specifies the duration of the fade-out animation in milliseconds (default is 400).
- **easing (optional):** Specifies the easing function to use for the animation (default is 'swing').
- **callback (optional):** A callback function to be executed after the fade-out animation.

Example 1

In the following example, we are fading out a single <div> element with the id "content" using the fadeOut() method

```
<html>
<head>
<script src="https://code.jquery.com/jquery-3.6.0.min.js"></script>
<script>
$(document).ready(function(){
  $("#fadeBtn").click(function(){
    $("#content").fadeOut();
  });
});
</script>
</head>
<body>

<div id="content" style="background-color: lightblue; padding: 20px;">
  <h2>This content will fade out.</h2>
</div>
<button id="fadeBtn">Click me!</button>
</body>
</html>
```

6. fadeToggle()

The fadeToggle() method in jQuery is used to toggle between the fadeIn() and fadeOut() methods.

Syntax

The syntax is as follows −

```
$(selector).fadeToggle(speed,easing,callback)
```

Above, speed is the speed of the fading effect. The easing can be swing or linear for speed at different animation points. Callback is the function to be executed after the method gets finished.

Example

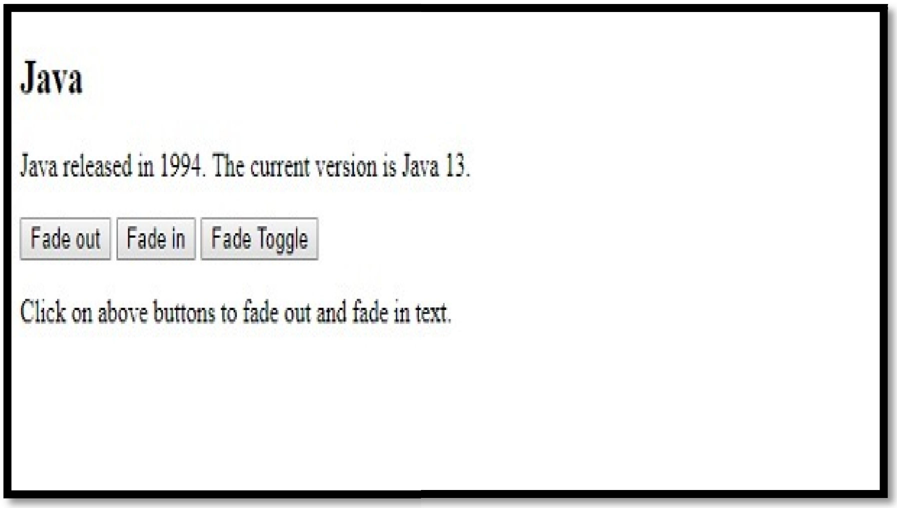
Let us now see an example to implement the jQuery fadeToggle() method −

```
<!DOCTYPE html>
<html>
```

```
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>
<script>
  $(document).ready(function(){
    $(".btnout").click(function(){
      $("div").fadeOut();
    });
    $(".btnin").click(function(){
      $("div").fadeIn();
    });
    $(".btnfade").click(function(){
      $("div").fadeToggle("slow");
    });
  });
</script>
</head>
<body>
<h2>Java</h2>
<div>
<p>Java released in 1994. The current version is Java 13.</p>
</div>
<button class="btnout">Fade out</button>
<button class="btnin">Fade in</button>
<button class="btnfade">Fade Toggle</button>
<p>Click on above buttons to fade out and fade in text.</p>
</body>
</html>
```

Output

This will produce the following output −

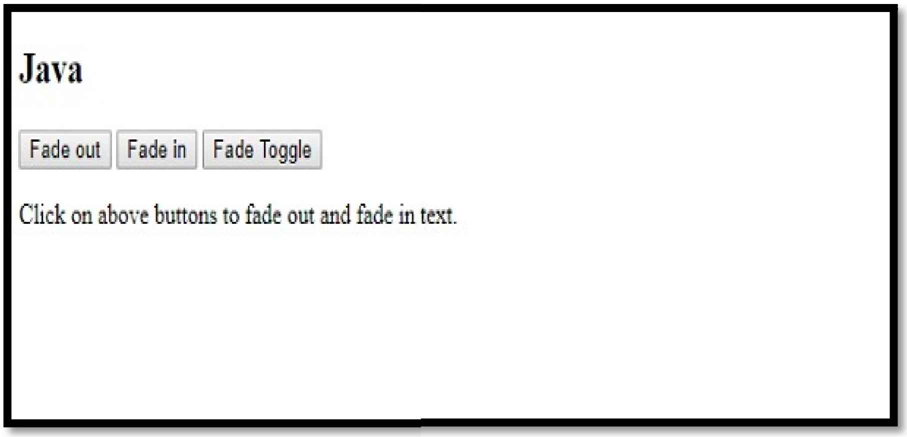


Java

Java released in 1994. The current version is Java 13.

Click on above buttons to fade out and fade in text.

Above, click on “Fade Toggle” to fade slowly −



Java

Click on above buttons to fade out and fade in text.

7. **fadeTo().**

jQuery fadeTo() method is used to fading to a given opacity.

Syntax:

1. $(selector).fadeTo(speed, opacity);
2. $(selector).fadeTo(speed, opacity, callback);
3. $(selector).fadeTo(speed, opacity, easing, callback);

speed: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

opacity:It specifies the opacity. The opacity value ranges between 0 and 1.

easing: It specifies the easing function to be used for transition.

callback: It is also an optional parameter. It specifies the function to be called after completion of fadeToggle() effect.

Let's take an example to demonstrate jQuery fadeTo() effect.

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("#div1").fadeTo("slow", 0.3);
      $("#div2").fadeTo("slow", 0.4);
      $("#div3").fadeTo("slow", 0.5);
   });
});
</script>
</head>
<body>
<p>See the fadeTo() method example with different parameters.</p>
<button>Click to fade boxes</button><br><br>
```

```
<div id="div1" style="width:80px;height:80px;background-color:red;"></div><br>
<div id="div2" style="width:80px;height:80px;background-color:green;"></div><br>
<div id="div3" style="width:80px;height:80px;background-color:blue;"></div>
</body>
</html>
```

❖ JQuery Sliding Method

1. slideDown():

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

Syntax:

1. $(selector).slideDown(speed);
2. $(selector).slideDown(speed, callback);
3. $(selector).slideDown(speed, easing, callback);

speed: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

easing: It specifies the easing function to be used for transition.

callback: It is also an optional parameter. It specifies the function to be called after completion of slideDown() effect.

Let's take an example to demonstrate jQuery slideDown() effect.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

<script>
$(document).ready(function(){
   $("#flip").click(function(){
      $("#panel").slideDown("slow");
   });
});
</script>
 <style>
#panel, #flip {
   padding: 5px;
   text-align: center;
   background-color: #00FFFF;
   border: solid 1px #c3c3c3;
}
#panel {
   padding: 50px;
   display: none;
}
</style>
</head>
<body>
<div id="flip">Click to slide down panel</div>
```

```
<div id="panel">Hello javatpoint.com!
It is the best tutorial website to learn jQuery and other languages.</div>
</body>
</html>
```

2. **slideUp()**

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

Syntax:

1. $(selector).slideUp(speed);
2. $(selector).slideUp(speed, callback);
3. $(selector).slideUp(speed, easing, callback);

speed: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

easing: It specifies the easing function to be used for transition.

callback: It is also an optional parameter. It specifies the function to be called after completion of slideUp() effect.

Let's take an example to demonstrate jQuery slideUp() effect.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#flip").click(function(){
      $("#panel").slideUp("slow");
   });
});
</script>
 <style>
#panel, #flip {
   padding: 5px;
   text-align: center;
   background-color: #00FFFF;
   border: solid 1px #c3c3c3;
}
#panel {
   padding: 50px;
}
</style>
</head>
<body>
<div id="flip">Click to slide up panel</div>
<div id="panel">Hello javatpoint.com!
It is the best tutorial website to learn jQuery and other languages.</div>
</body>
</html>
```

3. **slideToggle()**

jQuery slideToggle () method is used to toggle between slideUp() and slideDown() method. If the element is slide down, it will slide up the element and if it is slide up, it will slide down.

Syntax:

1. $(selector).slideToggle(speed);
2. $(selector).slideToggle(speed, callback);
3. $(selector).slideToggle(speed, easing, callback);

speed: It specifies the speed of the delay. Its possible vales are slow, fast and milliseconds.

easing: It specifies the easing function to be used for transition.

callback: It is also an optional parameter. It specifies the function to be called after completion of slideToggle() effect.

Let's take an example to demonstrate jQuery slideToggle() effect.

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("#flip").click(function(){
      $("#panel").slideToggle("slow");
   });
});
</script>
 <style>
#panel, #flip {
   padding: 5px;
   text-align: center;
   background-color: #00FFFF;
   border: solid 1px #c3c3c3;
}
#panel {
   padding: 50px;
   display:none;
}
</style>
</head>
<body>
<div id="flip">Click to slide toggle panel</div>
<div id="panel">Hello javatpoint.com!
It is the best tutorial website to learn jQuery and other languages.</div>
</body>
</html>
```

4. **animate()**

The jQuery animate() method provides you a way to create custom animations.

Syntax:

$(selector).animate({params}, speed, callback);

Here, **params** parameter defines the CSS properties to be animated.

The **speed** parameter is optional and specifies the duration of the effect. It can be set as "slow" , "fast" or milliseconds.

The **callback** parameter is also optional and it is a function which is executed after the animation completes.

Let's take a simple example to see the animation effect.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("div").animate({left: '450px'});
   });
});
</script>
</head>
<body>
<button>Start Animation</button>
<p>A simple animation example:</p>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

animate() method using multiple properties

You can use multiple properties to animate at the same time.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
   $("button").click(function(){
      $("div").animate({
         left: '250px',
         opacity: '0.5',
         height: '150px',
         width: '150px'
      });
   });
});
</script>
```

```
</head>
<body>
<button>Start Animation</button>
<div style="background:#125f21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>

<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            left: '250px',
            opacity: '0.5',
            height: '150px',
            width: '150px'
        });
    });
});
</script>
</head>
<body>
<button>Start Animation</button>
<div style="background:#125f21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

animate() method using relative values

You can also define relative values (it is relative to the element's current value) by putting += or -= in front of the value.

```
<!DOCTYPE html>
<html>
<head>
<script src="http://ajax.googleapis.com/ajax/libs/jquery/1.11.2/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("div").animate({
            left: '250px',
            height: '+=150px',
            width: '+=150px'
        });
    });
});
</script>
</head>
<body>
<button>Start Animation</button>
<div style="background:#98bf21;height:100px;width:100px;position:absolute;"></div>
</body>
</html>
```

Animation with Custom Speed

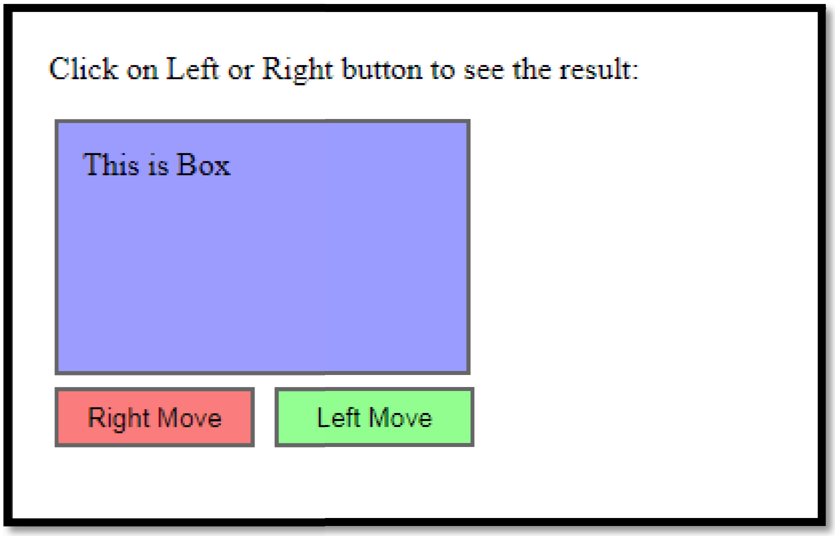
We can animate different CSS numerical properties (for example, width, height, or left) of a DOM element with different speed.

Example

Let's re-write above example, where we will animate <div>'s right movement with a speed parameter of 1000 milliseconds, and left movement with a speed parameter of 5000 milliseconds.

```
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src = "https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
$(document).ready(function() {
  $("#right").click(function(){
    $("div").animate({left: '250px'}, 1000);
  });
  $("#left").click(function(){
    $("div").animate({left: '0px'}, 5000);
  });
});
</script>
<style>
  #left, #right{margin:3px;border:2px solid #666; height:30px;
width:100px;cursor:pointer;}
  #box{position:relative;margin:3px;padding:12px;border:2px solid #666;  height:100px;
width:180px;}
</style>
</head>
<body>
  <p>Click on Left or Right button to see the result:</p>
  <div id="box"  style="background-color:#9c9cff;">This is Box</div>
  <button id="right" style="background-color:#fb7c7c;">Right Move</button>
  <button id="left" style="background-color:#93ff93;">Left Move</button>
</body>
</html>
```

Output



Click on Left or Right button to see the result:

This is Box

Animation with Pre-defined Values

We can use strings **'show'**, **'hide'**, and **'toggle'** as the value of CSS numeric properties.

Example

Following is an example where we are setting **left** property of an element to either **hide** or **show** with the help of two buttons.

```
<!doctype html>
<html>
<head>
<title>The jQuery Example</title>
<script src = "https://www.tutorialspoint.com/jquery/jquery-3.6.0.js"></script>
<script>
$(document).ready(function() {
  $("#right").click(function(){
    $("div").animate({left: 'hide'});
  });
  $("#left").click(function(){
    $("div").animate({left: 'show'});
  });
});
</script>
<style>
  #left, #right{margin:3px;border:2px solid #666; height:30px;
width:100px;cursor:pointer;}
  #box{position:relative;margin:3px;padding:12px;border:2px solid #666; height:100px;
width:180px;}
</style>
</head>
<body>
  <p>Click on Left or Right button to see the result:</p>
  <div id="box"  style="background-color:#9c9cff;">This is Box</div>
  <button id="right" style="background-color:#fb7c7c;">Right Move</button>
  <button id="left" style="background-color:#93ff93;">Left Move</button>
</body>
</html>
```

Stop():

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

Syntax:

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

- ✓ The optional stopAll parameter specifies whether also the animation queue should be cleared or not. Default is false, which means that only the active animation will be stopped, allowing any queued animations to be performed afterwards.
- ✓ The optional goToEnd parameter specifies whether or not to complete the current animation immediately. Default is false.
- ✓ So, by default, the stop() method kills the current animation being performed on the selected element.
- ✓ The following example demonstrates the stop() method, with no parameters:

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#flip").click(function(){
    $("#panel").slideDown(5000);
  });
  $("#stop").click(function(){
    $("#panel").stop();
  });
});
</script>
<style>
#panel, #flip {
  padding: 5px;
  font-size: 18px;
  text-align: center;
  background-color: #555;
  color: white;
  border: solid 1px #666;
  border-radius: 3px;
}
```

```
#panel {
  padding: 50px;
  display: none;
}
</style>
</head>
<body>
<button id="stop">Stop sliding</button>
<div id="flip">Click to slide down panel</div>
<div id="panel">Hello world!</div>
</body>
</html>
```

❖ Add Element, Remove Element, Add Class and Remove Class.

➢ Add New HTML Content

We will look at four jQuery methods that are used to add new content:

- append() - Inserts content at the end of the selected elements
- prepend() - Inserts content at the beginning of the selected elements
- after() - Inserts content after the selected elements
- before() - Inserts content before the selected elements

1. jQuery append() Method

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").append(" <b>Appended text</b>.");
  });
  $("#btn2").click(function(){
    $("ol").append("<li>Appended item</li>");
  });
});
</script>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<button id="btn1">Append text</button>
```

```
<button id="btn2">Append list items</button>
</body>
</html>
```

2. jQuery prepend() Method

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("p").prepend("<b>Prepended text</b>. ");
  });
  $("#btn2").click(function(){
    $("ol").prepend("<li>Prepended item</li>");
  });
});
</script>
</head>
<body>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<ol>
  <li>List item 1</li>
  <li>List item 2</li>
  <li>List item 3</li>
</ol>
<button id="btn1">Prepend text</button>
<button id="btn2">Prepend list item</button>
</body>
</html>
```

3. jQuery after() and before() Methods

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#btn1").click(function(){
    $("img").before("<b>Before</b>");
  });
```

```
 $("#btn2").click(function(){
  $("img").after("<i>After</i>");
 });
});
</script>
</head>
<body>

<img src="/images/w3jquery.gif" alt="jQuery" width="100" height="140"><br><br>

<button id="btn1">Insert before</button>
<button id="btn2">Insert after</button>

</body>
</html>
```

➢ jQuery - Remove Elements

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

Remove Elements/Content

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

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

1. jQuery remove() Method

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
 $("button").click(function(){
  $("#div1").remove();
 });
});
</script>
</head>
<body>
<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">
This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>
</div>
<br>
<button>Remove div element</button>
</body>
</html>
```

2. jQuery empty() Method

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").empty();
  });
});
</script>
</head>
<body>

<div id="div1" style="height:100px;width:300px;border:1px solid black;background-color:yellow;">

This is some text in the div.
<p>This is a paragraph in the div.</p>
<p>This is another paragraph in the div.</p>

</div>
<br>

<button>Empty the div element</button>

</body>
</html>
```

Filter the Elements to be Removed

- ✓ The jQuery remove() method also accepts one parameter, which allows you to filter the elements to be removed.
- ✓ The parameter can be any of the jQuery selector syntaxes.
- ✓ The following example removes all <p> elements with class="test":

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").remove(".test");
  });
});
</script>
<style>
.test {
  color: red;
  font-size: 20px;
}
</style>
</head>
<body>
<p>This is a paragraph.</p>
<p class="test">This is another paragraph.</p>
<p class="test">This is another paragraph.</p>
<button>Remove all p elements with class="test"</button>
</body>
</html>
```

This example removes all <p> elements with class="test" and class="demo":

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").remove(".test, .demo");
  });
});
</script>
<style>
.test {
  color: red;
  font-size: 20px;
}

.demo {
  color: green;
  font-size: 25px;
}
</style>
</head>
<body>

<p>This is a paragraph.</p>
<p class="test">This is p element with class="test".</p>
```

```
<p class="test">This is p element with class="test".</p>
<p class="demo">This is p element with class="demo".</p>

<button>Remove all p elements with class="test" and class="demo"</button>

</body>
</html>
```

❖ Add Class and Remove Class.

With jQuery, it is easy to manipulate the style of elements.

➢ jQuery Manipulating CSS

jQuery has several methods for CSS manipulation. We will look at the following methods:

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

jQuery addClass() Method

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

Example:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").addClass("blue");
    $("div").addClass("important");
  });
});
</script>
<style>
.important {
  font-weight: bold;
  font-size: xx-large;
}
.blue {
  color: blue;
}
</style>
</head>
<body>
<h1>Heading 1</h1>
```

```
<h2>Heading 2</h2>
<p>This is a paragraph.</p>
<p>This is another paragraph.</p>
<div>This is some important text!</div><br>
<button>Add classes to elements</button>
</body>
</html>
```

You can also specify multiple classes within the addClass() method:

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#div1").addClass("important blue");
  });
});
</script>
<style>
.important {
  font-weight: bold;
  font-size: xx-large;
}
.blue {
  color: blue;
}
</style>
</head>
<body>

<div id="div1">This is some text.</div>
<div id="div2">This is some text.</div>
<br>

<button>Add classes to first div element</button>

</body>
</html>
```

jQuery removeClass() Method

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
```

```
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").removeClass("blue");
  });
});
</script>
<style>
.blue {
  color: blue;
}
</style>
</head>
<body>

<h1 class="blue">Heading 1</h1>
<h2 class="blue">Heading 2</h2>

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

<button>Remove class from elements</button>

</body>
</html>
```

jQuery toggleClass() Method

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

```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("h1, h2, p").toggleClass("blue");
  });
});
</script>
<style>
.blue {
  color: blue;
}
</style>
</head>
```

```
<body>

<h1>Heading 1</h1>
<h2>Heading 2</h2>

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

<button>Toggle class</button>

</body>
</html>
```