UNIT – I
jQuery

Syllabus:

Introduction, selectors, events, effects, manipulating HTML and CSS using jQuery

1.1. INTRODUCTION TO JQUERY:

- jQuery is a Client-side javascript library Created by John Resig in the year 2006.
- **Definition:**
  - jQuery is a lightweight, "write less, do more", javascript library.
  - Designed to simplify - HTML DOM traversal & manipulation, Event handling, CSS animation and AJAX.
- It is free, open source software.
- JQuery is a scripting language. Unlike traditional programming languages, it is interpreted, not executed.
- The purpose of jQuery is to make it much easier to use JavaScript on your website.
- jQuery's syntax is designed to make it easier to navigate a document, select DOM elements, create animations, handle events, and develop Ajax applications.
- It also provides capabilities for developers to create plug-ins on top of the JavaScript library.
- **Advantages of jQuery:**
  1. Simple and easy to use:
     - jQuery library is built using simpler and shorter codes.
     - It consists of a large number of predefined methods which can be directly used in our applications.
     - With its open coding standards and simple syntax, web designers can shorten the time that it takes to deploy a site or application.
  2. Compact and light weight library about 19KB in size.
  3. Open source library:
     - jQuery is an open source library that is free and supported well across different applications.
     - This means that anyone can use this language in their applications without worrying about any licensing or compatibility issues.

4. Separates JavaScript and HTML:
   - Instead of using HTML attributes to call JavaScript functions for event handling, jQuery can be used to handle events purely in JavaScript. Thus, the HTML tags and JavaScript can be completely separated.
5. Cross-browser compatibility:
   - JavaScript engines of different browsers differ slightly so JavaScript code that works for one browser may not work for another.
   - jQuery handles all these cross browser inconsistencies and provides a consistent interface that works across different browsers.
6. AJAX support:
   - Enables a web page to make AJAX requests to a web server to add the data, without reloading the page.
7. Event handling:
   - jQuery is tailor-made to respond to events in an HTML page. In jQuery, most DOM events have an equivalent jQuery method to handle them.
8. Custom animations and effects:
   - jQuery provides a lot of built in methods to add effects like fading and sliding of elements.
   - It also allows developer to add custom animations to web pages.
9. HTML/DOM manipulation:
   - The DOM is a tree structure representation of all the elements of a webpage.
   - The jQuery made it easy to select DOM elements, traverse them and modifying their content.
   - jQuery methods like html(), text(), val() and attr() can be used for this purpose.
10. Extensibility:
    - jQuery makes extending the framework very simple. New events, elements and methods can be easily added and then reused as plug-in.
11. Brevity and clarity: jQuery promotes brevity and clarity with features like chainable functions and shorthand function name.

- **Adding jQuery to Web pages:**

  There are two ways to use jQuery:
  - Local Installation:
      - Download the jQuery library from jQuery.com.

        - Save the downloaded library file in the same directory as the pages where you wish to use it.
        - *Syntax to include downloaded jquery in a HTML page:*

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

    - CDN Based Version:
        - jQuery library can be included in your HTML page directly from Content Delivery Network (CDN), like Google.
        - Google provided online version of the jQuery library which results in faster loading of pages.
        - *Syntax to include jquery online using Google CDN in a HTML page:*

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

- **jQuery Syntax:**
    - The jQuery syntax is used for selecting HTML elements and performing some action on the element(s).
    - Basic syntax is:
        1. ***$(selector).action()***
        2. ***$(selector).action(function(){***
           });
            - 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:
        - `$("p").hide()` - hides all <p> elements.
        - `$(this).hide()` - hides the current element.
    - **The Document Ready Event:**
        - To prevent any jQuery code from running before the document is finished loading all jQuery methods are written inside a a document ready event:

- Syntax:

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

1.2. SELECTORS:

- jQuery selectors allow you to select and manipulate HTML element(s).
- There are mainly 3 types of selectors:
  1. Element Selector
  2. Id Selector
  3. Class Selector
- All selectors in jQuery start with the dollar sign and parentheses: **`$()`**.

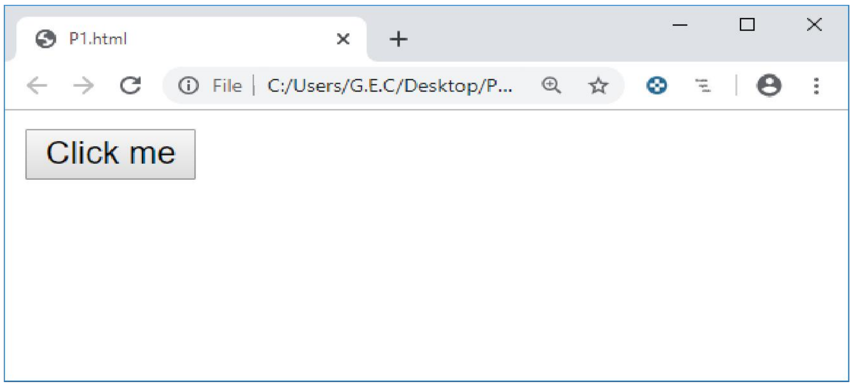
1.2.1. The Element Selector:

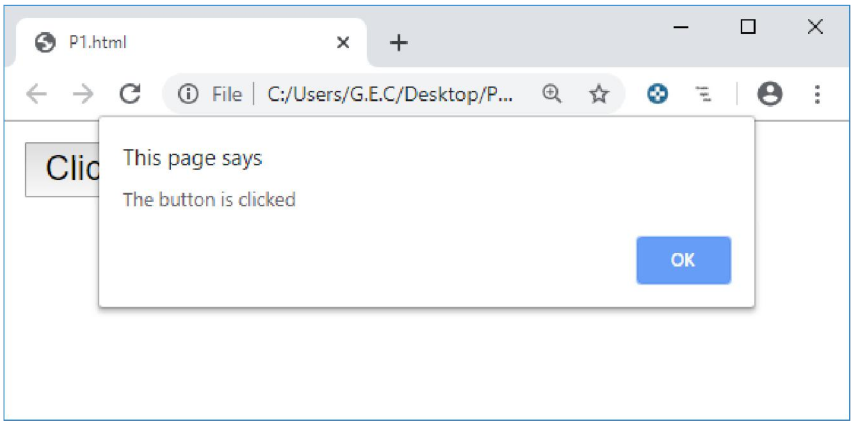
- The jQuery *element* selector selects elements based on the element name.
- Syntax:
  - `$("element").action()`

    element - specifies the element/tag name
- **Example:**

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
      alert("The button is clicked");
  });
});
</script>
</head>
<body>
<button>Click me</button>
</body>
</html>
```

Output:





This page says

The button is clicked

1.2.2. Id Selector:

- The jQuery *#id* selector uses the *id attribute of a 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:
- **Syntax:**

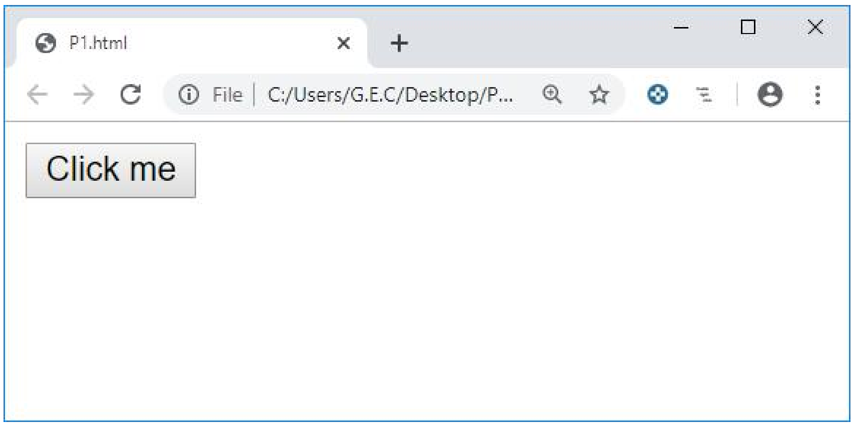
```
$("#id")
```

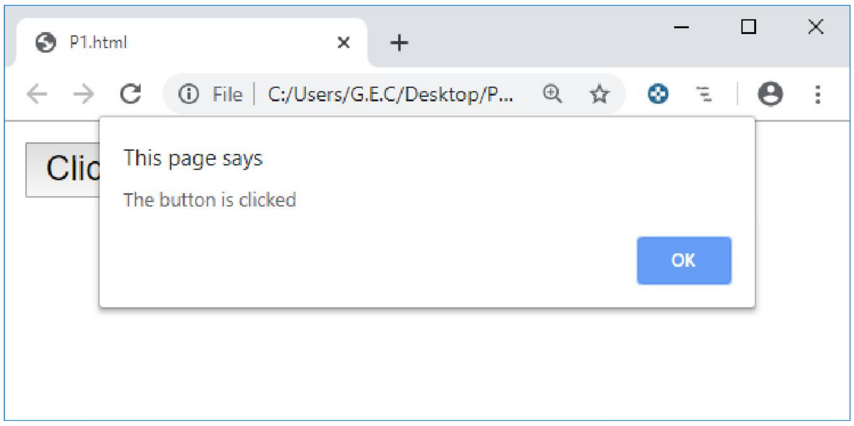
- **Example:**

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1
/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#b1").click(function(){
      alert("The button is clicked");
  });
});
```

```
</script>
</head>
<body>
<button id="b1">Click me</button>
</body>
</html>
```

Output:





This page says
The button is clicked

1.2.3. Class Selector:

- The .class selector selects all elements with the specific class.
- The class refers to the class attribute of an HTML element.
- The class attribute is used to set a particular style for several HTML elements.
- To find elements with a specific class, write a period character, followed by the name of the class.
- **Syntax:**

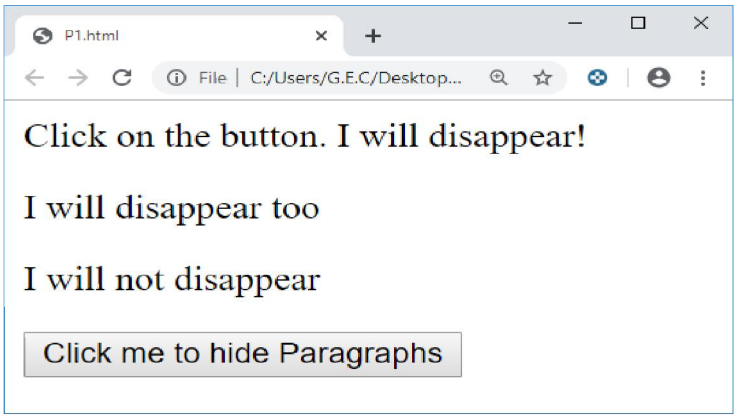
  ```
  $(“.class”)
  ```

- **Example:**

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

```
<script>
$(document).ready(function(){
  $("button").click(function(){
      $(".test").hide();
  });
});
</script>
</head>
<body>
<p class="test">Click on the button. I will
disappear!</p>
<p class="test">I will disappear too </p>
<p>I will not disappear</p>
<button>Click me to hide Paragraphs</button>
</body>
</html>
```

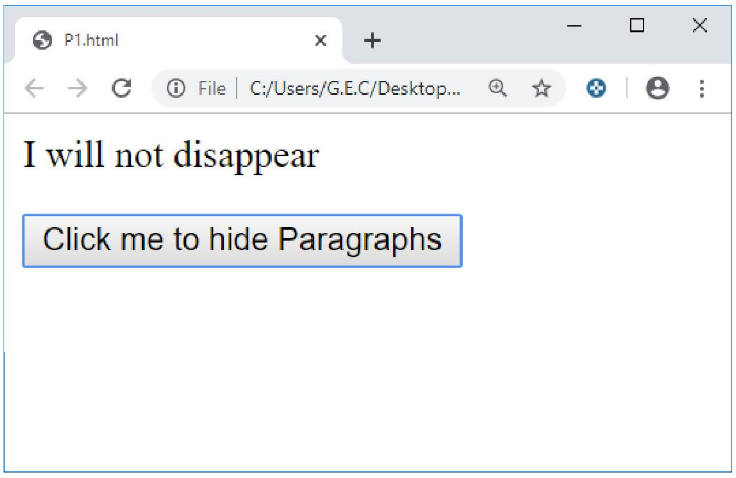
- **Output:**



Click on the button. I will disappear!

I will disappear too

I will not disappear



I will not disappear

1.3. EVENTS:

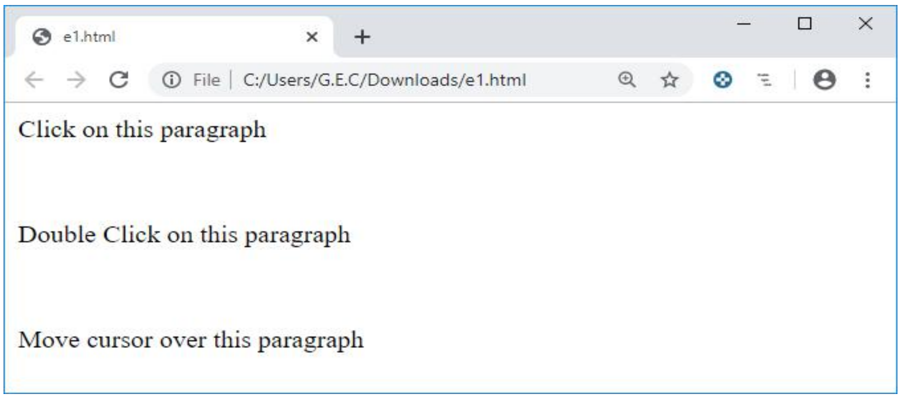
- jQuery is Event-driven – “respond to events in an HTML page”.
- “An event represents the precise moment when something happens”.
- A program contains necessary block of code known as “event handler”, to handle an event.
- **Example:**
  - Clicking of a mouse
  - Loading of a web page
  - Pressing a key on a keyboard
  - Submitting a form

Event	Description
blur	Occurs when an element loses focus
change	Occurs when the value of an element has been changed
click	Occurs when an element is clicked
dblclick	Occurs when an element is double-clicked.
focus	Occurs when an element gets focus
hover	When the mouse pointer hovers over the selected elements.
keydown, keypress	Occurs when a keyboard key is pressed down.
keyup	Occurs when a keyboard key is released
load	Occurs when a specified element has been loaded
mousedown	Occurs when the left mouse button is pressed down over the selected element.
mouseenter	Mouse pointer enters the selected element.
mouseleave, mouseout	Mouse pointer leaves the selected element.
mousemove	Mouse pointer moves within the selected element.
mouseover	Mouse pointer is over the selected element.
mouseup	Left mouse button is released over the selected element.
ready	Occurs when the dom (document object model) has been loaded.
resize	Occurs when the browser window changes size.
scroll	Occurs when the user scrolls in the specified element.
select	Occurs when a text is selected in a text area or a text field.
submit	Occurs when a form is submitted.
unload	Occurs when the user navigates away from the page

- **Example:**

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.
min.js">
</script>
<script>
   $(document).ready(function()
   {
         $("#p1").click(function()
         {
               alert("You clicked on the paragraph");
         });
         $("#p2").dblclick(function()
         {
               alert("You double clicked on the paragraph");
         });
         $("#p3").hover(function()
         {
               alert("mouse moved over the paragraph");
         });
   });
</script>
</head>
<body>
<p id="p1">Click on this paragraph</p>
<br>
<p id="p2">Double Click on this paragraph</p>
<br>
<p id="p3">Move cursor over this paragraph</p>
</body>
</html>
```

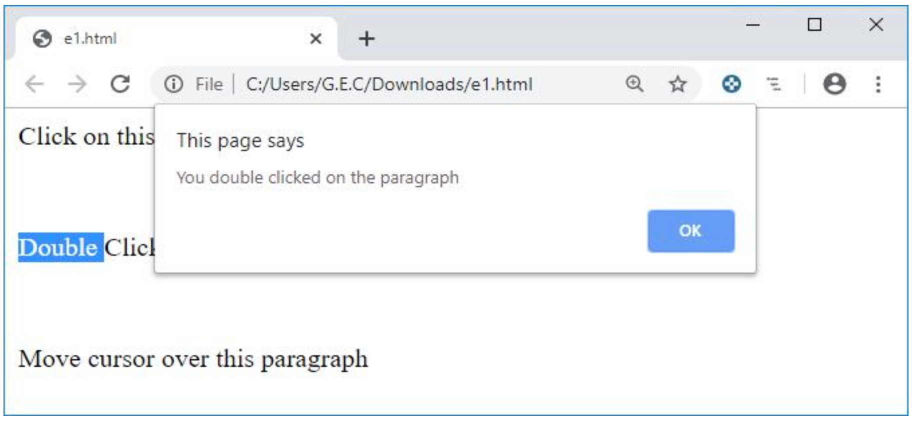
- **Output:**



Click on this paragraph

Double Click on this paragraph

Move cursor over this paragraph


e1.html

C:/Users/G.E.C/Downloads/e1.html

Click on this

Double Click

Move cursor over this paragraph

This page says

You double clicked on the paragraph

1.3.1. Keyboard events:

List of keyboard events:

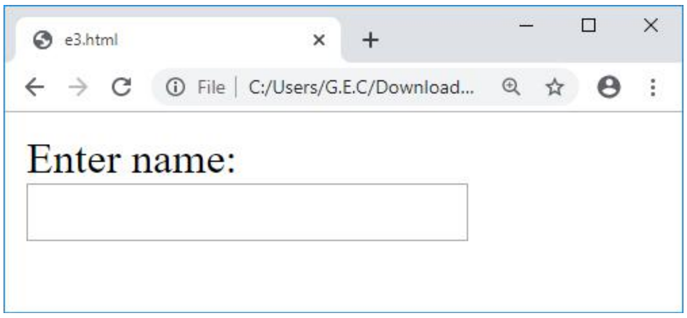
1. keydown
2. keypress
3. keyup

- **Example:**

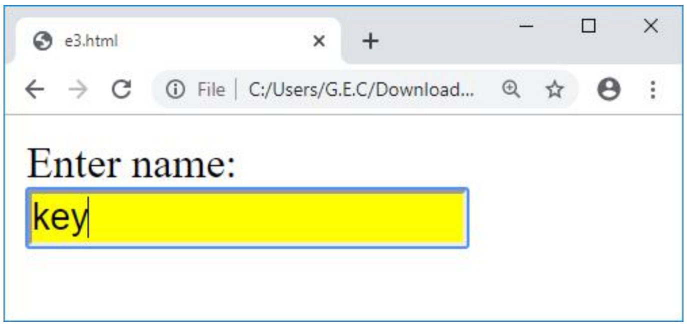
```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1
/jquery.min.js">
</script>
<script>
 $(document).ready(function()
 {
        $("input").keydown(function()
        {
               $(this).css("background-color","yellow");
        });
        $("input").keyup(function()
        {
               $(this).css("background-color","pink");
        });

 });
</script>
</head>
<body>
<form method="post">
Enter name:<input type="text" name="t1"><br>
</form>
</body>
</html>
```

- **Output:**



Enter name:



Enter name:

key

1.3.2. Mouse events:

List of Mouse events:

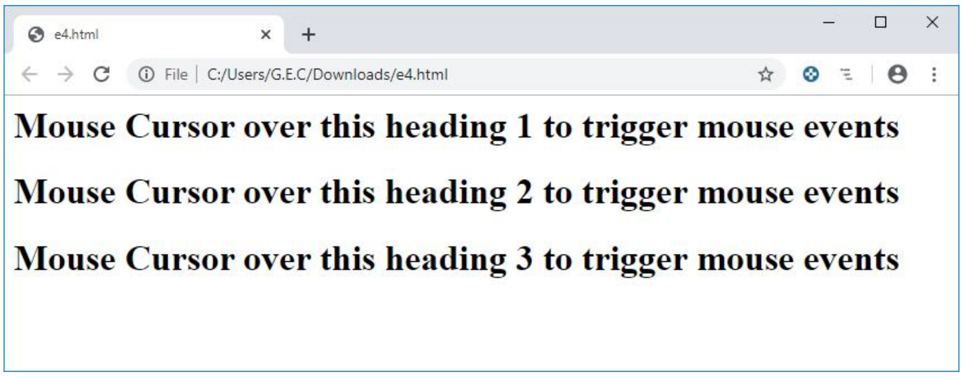
1. mousedown
2. mouseenter
3. mouseleave
4. mousemove
5. mouseout
6. mouseover
7. mouseup

Example:

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.
min.js">
</script>
<script>
    $(document).ready(function()
    {
          $("#i1").mouseenter(function()
          {
                $("#i1").css("background-color","yellow");
          });
          $("#i1").mouseout(function()
          {
                $("#i1").css("background-color","pink");
          });
          $("#i2").mousedown(function()
          {
                $("#i2").css("background-color","blue");
        });
        $("#i2").mouseup(function()
        {
                $("#i2").css("background-color","green");
        });
        $("#i3").mouseover(function()
        {
                alert("cursor is over this heading");
        });

    });
</script>
</head>
<body>
<h1 id="i1">Mouse Cursor over this heading 1 to trigger mouse
events</h1>
<h1 id="i2">Mouse Cursor over this heading 2 to trigger mouse
events</h1>
<h1 id="i3">Mouse Cursor over this heading 3 to trigger mouse
events</h1>
</body>
</html>
```

Output:



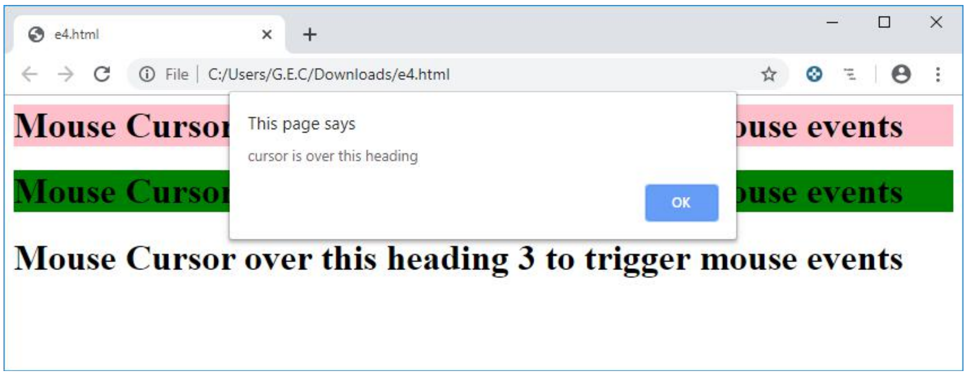
e4.html

C:/Users/G.E.C/Downloads/e4.html

Mouse Cursor over this heading 1 to trigger mouse events

Mouse Cursor over this heading 2 to trigger mouse events

Mouse Cursor over this heading 3 to trigger mouse events



e4.html

C:/Users/G.E.C/Downloads/e4.html

Mouse Cursor ... ouse events

Mouse Cursor ... ouse events

This page says

cursor is over this heading

OK

Mouse Cursor over this heading 3 to trigger mouse events

1.3.3. Form events:

List of form events:

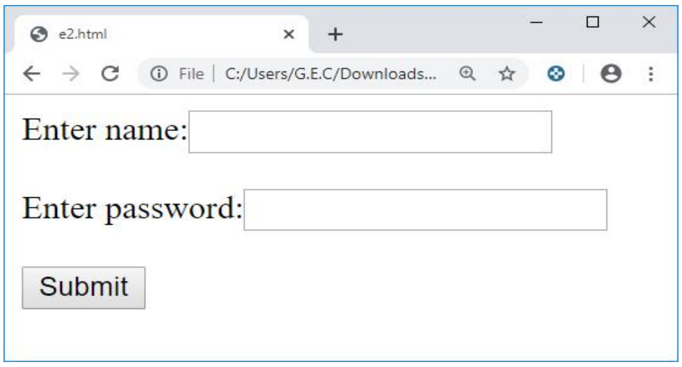
1. Submit
2. Change
3. Focus
4. Blur

Example:

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jq
uery.min.js">
</script>
<script>
   $(document).ready(function()
   {
        $("form").submit(function()
        {
             alert("Form is submitted");
        });
        $("input").focus(function()
        {
             $(this).css("background-color","yellow");
        });
        $("input").blur(function()
        {
             $(this).css("background-color","pink");
        });
        $("input").change(function()
        {
             alert("Text is changed");
        });
        $("input").select(function()
        {
             alert("Text is selected");
        });
   });
</script>
</head>
<body>
<form method="post">
Enter name:<input type="text" name="t1"><br>
Enter password:<input type="password" name="t2"><br>
<button type="submit">Submit</button>
```

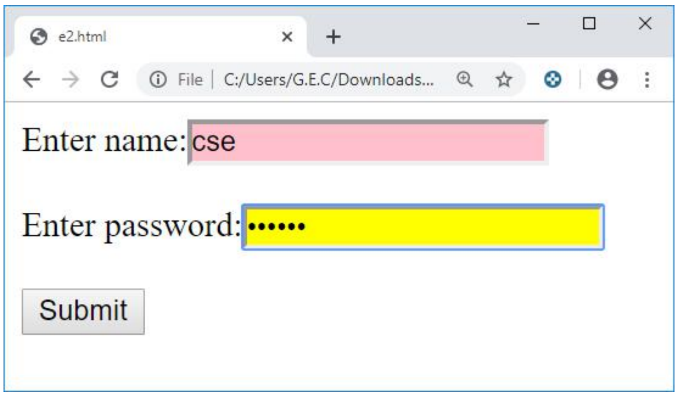
```
</form>
</body>
</html>
```

Output:



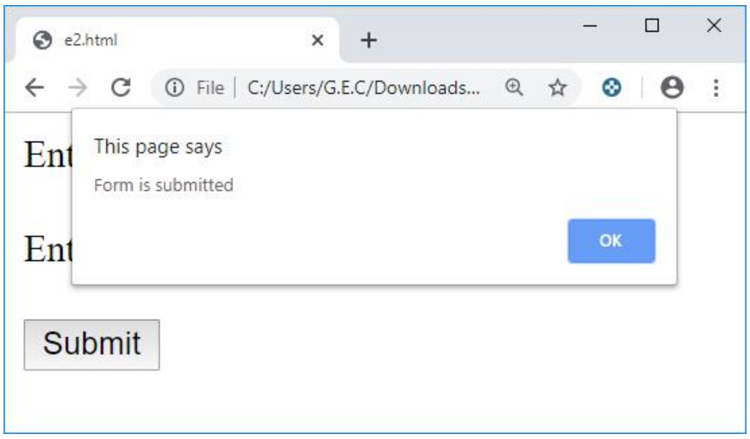
Enter name:

Enter password:



Enter name: cse

Enter password: ••••••



This page says

Form is submitted

1.3.4. Document/Window Events:

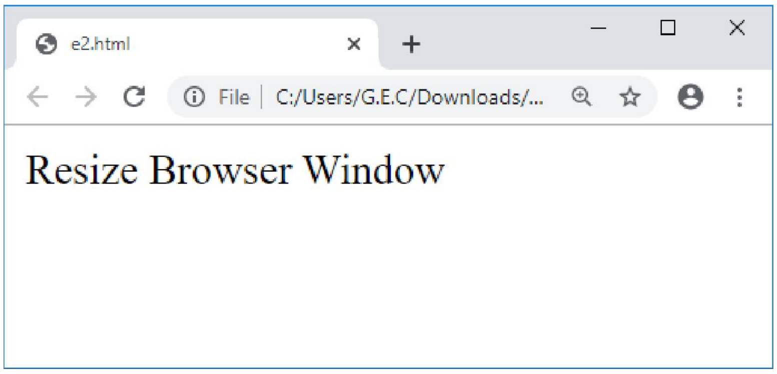
List of Document/Window Events:

1. Load
2. Resize
3. Scroll
4. Unload

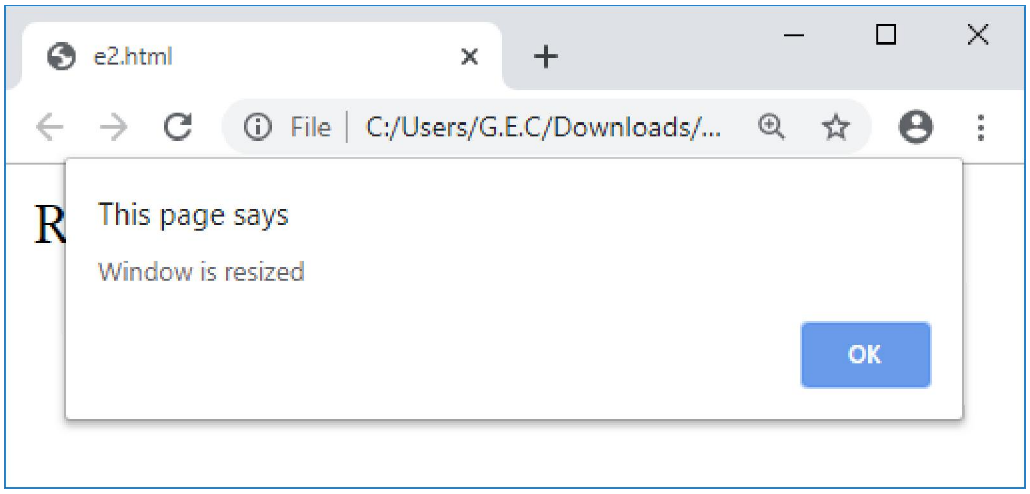
Example:

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

```
<script>
$(document).ready(function(){
  $(window).resize(function(){
    alert("Window is resized");
  });
});
</script>
</head>
<body>
<p>Resize Window</p>
</body>
</html>
```
Output:



Resize Browser Window



This page says

Window is resized

1.4. EFFECTS:

- The jQuery library provides several techniques for adding animation to a web page.
- It contains various methods to apply simple, standard animations that are frequently used, and also sophisticated custom effects.

Effects	**Method**	**Description**
Hide/Show	hide()	Hides the selected elements
	show()	Shows the selected elements
	toggle()	Toggles between the hide() and show() methods
Fading	fadeIn()	Fades in the selected elements
	fadeOut()	Fades out the selected elements
	fadeTo()	Fades in/out the selected elements to a given opacity
	fadeToggle()	Toggles between the fadeIn() and fadeOut() methods
Sliding	slideUp()	Slides-up (hides) the selected elements
	slideDown()	Slides-down (shows) the selected elements
	slideToggle()	Toggles between the slideUp() and slideDown() methods
Animation	animate()	Runs a custom animation on the selected elements
Stop	stop()	Stops the currently running animation for the selected elements
	delay()	Sets a delay for all queued functions on the selected elements. $(selector).delay(speed)

1.4.1. Showing and Hiding of elements:

- **hide()** - hide() method hides the selected elements.
  - Syntax:

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

    - speed - Specifies the speed of the hide/show effect. Possible values: milliseconds, “slow”, “fast”.

        - callback - A function to be executed after the method is completed.

- **show()** - shows the hidden, selected elements.
    - Syntax:

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

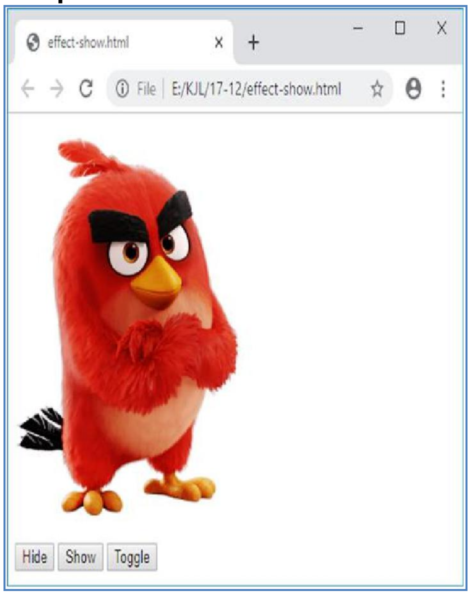
- **toggle()** - toggles between **hide()** and **show()** for the selected elements.
    - show() is run if an element is hidden.
    - hide() is run if an element is visible
    - Syntax:

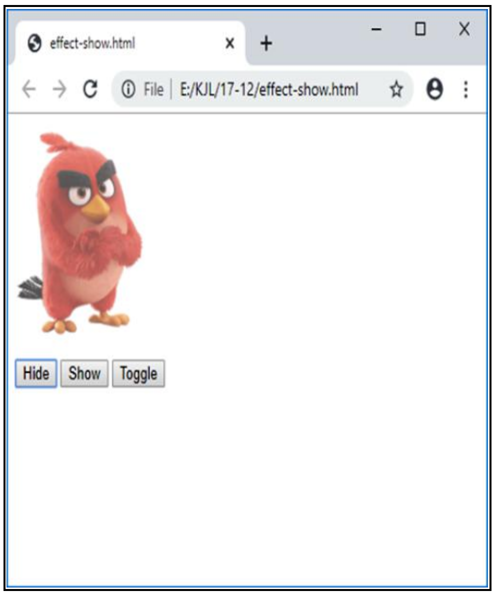
      `$(selector).toggle(speed,callback);`

Example:

```
<html>
<head>
  <script
  src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.
  min.js">
  </script>
  <script>
  $(document).ready(function(){
    $("#b1").click(function(){
        $("img").hide(1000);
    });
    $("#b2").click(function(){
        $("img").show("slow");
    });
    $("#b3").click(function(){
        $("img").toggle("fast");
    });
  });
  </script>
  </head>
  <body>
  <img src="6.png">
  <button id="b1"> Hide </button>
  <button id="b2"> Show </button>
  <button id="b3"> Toggle </button>
  </body>
  </html>
```

Output:





1.4.2. Fading effects:

With jQuery you can fade an element in and out of visibility. jQuery has the following fade methods:

- **fadeIn() -** used to fade in a hidden element.
  - Syntax:

    `$(selector).fadeIn(speed,callback);`
- **fadeOut() -** used to fade out a visible element.
  - Syntax:

    `$(selector).fadeOut(speed,callback);`
- **fadeToggle() –** toggles between the fadeIn() and fadeOut() methods.
  - If the elements are faded out, fadeToggle() will fade them in.
  - If the elements are faded in, fadeToggle() will fade them out.
  - Syntax:

    `$(selector).fadeToggle(speed,callback);`
- **fadeTo() -** allows fading to a given opacity (value between 0 and 1).
  - Syntax:

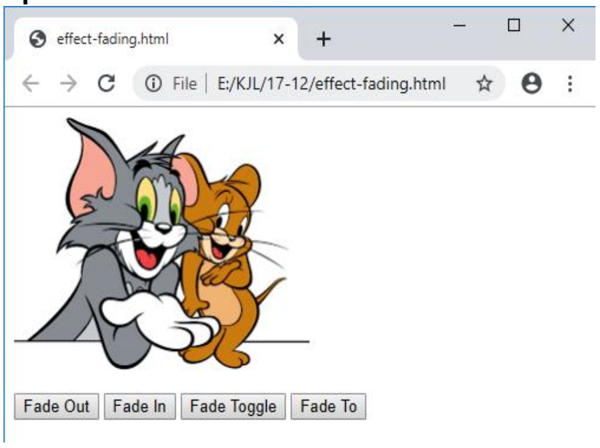
    `$(selector).fadeTo(speed,opacity,callback);`
  - Opacity: Specifies the opacity to fade to. Must be a number between 0.00 and 1.00.

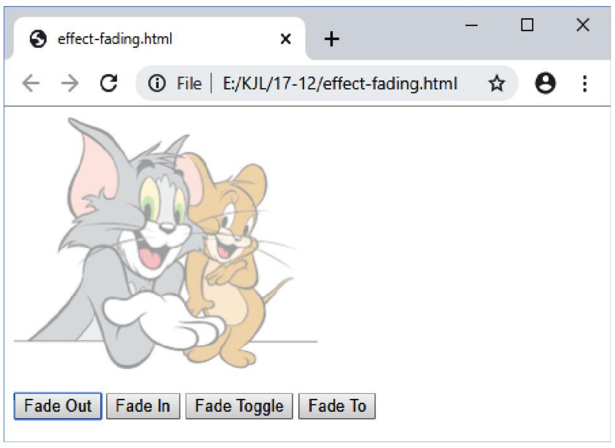
Example:

```
<html>
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min
.js">
</script>
<script>
$(document).ready(function(){
    $("#b1").click(function(){
        $("img").fadeOut(1000);
    });
    $("#b2").click(function(){
        $("img").fadeIn("slow");
    });
    $("#b3").click(function(){
        $("img").fadeToggle("fast");
    });
    $("#b4").click(function(){
        $("img").fadeTo("slow",0.3);
    });

});
</script>
</head>
<body>
<img src="d1.png"><br><br>
<button id="b1"> Fade Out </button>
<button id="b2"> Fade In </button>
<button id="b3"> Fade Toggle </button>
<button id="b4"> Fade To </button>
</body>
</html>
```

Output:





1.4.3. Sliding effects:

With jQuery you can create a sliding effect on elements. jQuery slide methods slide elements up and down.

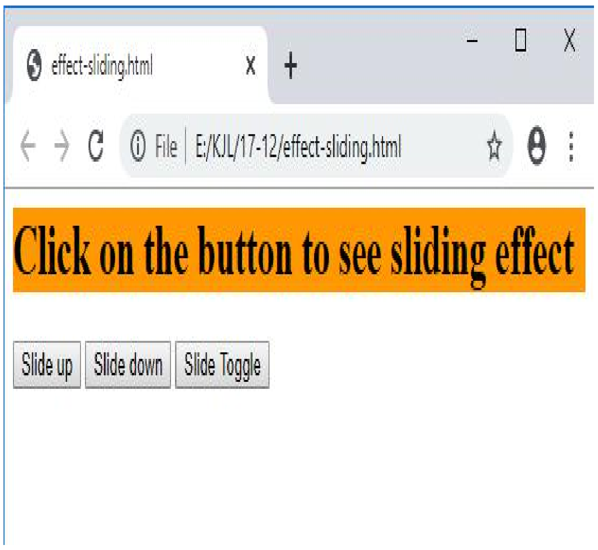
- **slideUp()** – used to slide up an element.
  - Syntax:
    ```
    $(selector).slideUp(speed,callback);
    ```
- **slideDown()** - used to slide down an element.
  - Syntax:
    ```
    $(selector).slideDown(speed,callback);
    ```
- **slideToggle()** – toggles between the slideDown() and slideUp() methods.
  - If the elements have been slide down, slideToggle() will slide them up.
  - If the elements have been slide up, slideToggle() will slide them down.
  - Syntax:
    ```
    $(selector).slideToggle(speed,callback);
    ```

Program for sliding effects:

```
<html>
<head>
<script
src="http://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.
js">
</script>
<script>
$(document).ready(function(){
       $("#b1").click(function(){
          $("h1").slideUp(1000);
       });
       $("#b2").click(function(){
          $("h1").slideDown("slow");
       });
       $("#b3").click(function(){
          $("h1").slideToggle("fast");
       });
});
</script>
</head>
<body>
```

```
<h1 style="background-color:orange">Click on the button to see
sliding effect</h1>
<button id="b1"> Slide up </button>
<button id="b2"> Slide down </button>
<button id="b3"> Slide Toggle </button>
</body>
</html>
```

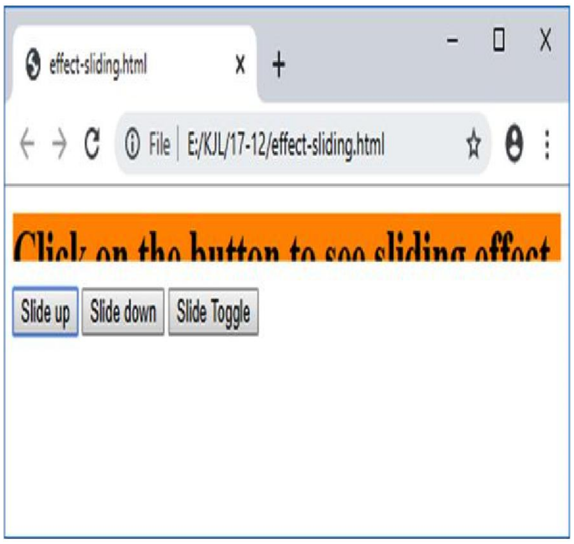
Output:



effect-sliding.html

File | E:/KJL/17-12/effect-sliding.html

Click on the button to see sliding effect



effect-sliding.html

File | E:/KJL/17-12/effect-sliding.html

1.4.4. Animate() and stop():

- **animate()**
  - animate() method is used to create custom animations.
  - multiple (CSS) properties can be animated at the same time using animate()
  - changes an element from one state to another with CSS styles.
  - Syntax:

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

    - **params** - required parameter defines the CSS properties to be animated.
- **Stop() –**
  - The jQuery stop() method is used to stop an animation or effect before it is finished.
  - works for all jQuery effect functions, including sliding, fading and custom animations.
  - Syntax:

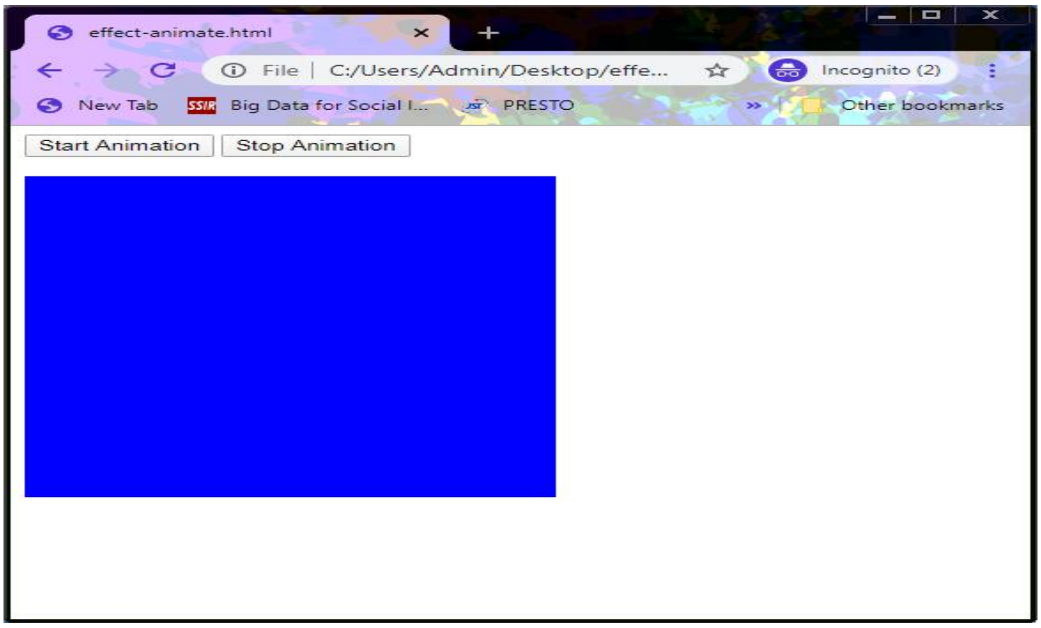
    `$(selector).stop();`

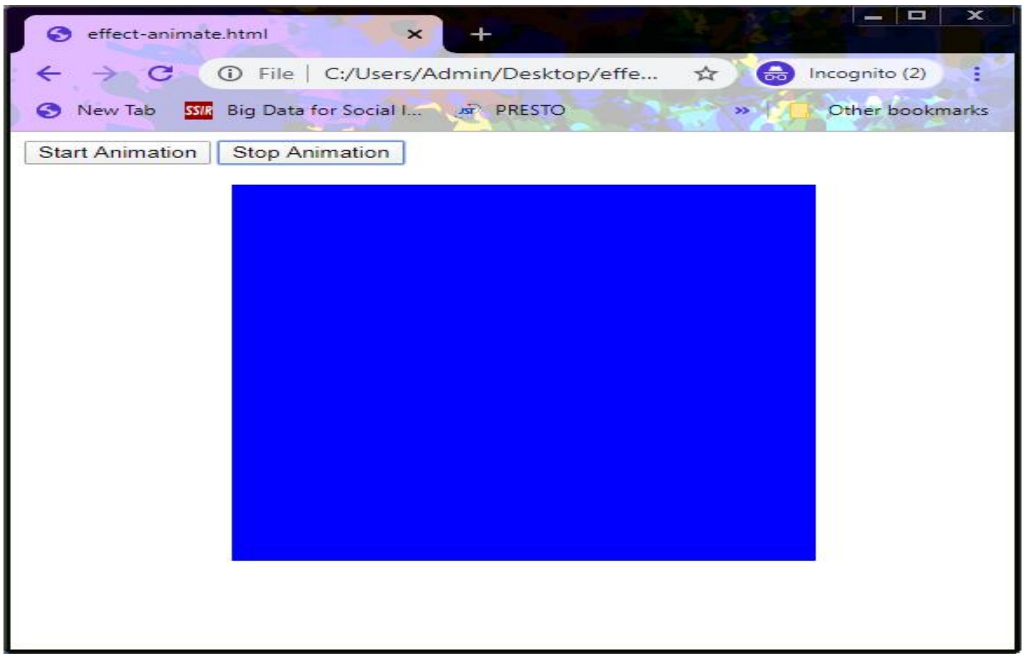
    - kills the current animation being performed on the selected element.

- **Program for animate and stop:**

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery
.min.js"></script>
<script>
$(document).ready(function(){
  $("#b1").click(function(){
    $("div").animate({left:'850px', height:'+=250',
width:'+=250px'},1000);
  });
  $("#b2").click(function(){
    $("div").stop();
  });
});
</script>
</head>
<body>
<button id="b1">Start Animation</button>
<button id="b2">Stop Animation</button>
<br><br>
<div style="background-
color:blue;width:300px;height:300px;position:absolute"></div>
</body>
</html>
```

Output:





1.4.5. Callback functions:

- A callback function is a function that is executed after the current effect is finished.
- It is passed as an argument to the effect methods and they typically appear as the last argument of the method.
- Syntax:

```
$(selector).method(speed,callback);
```

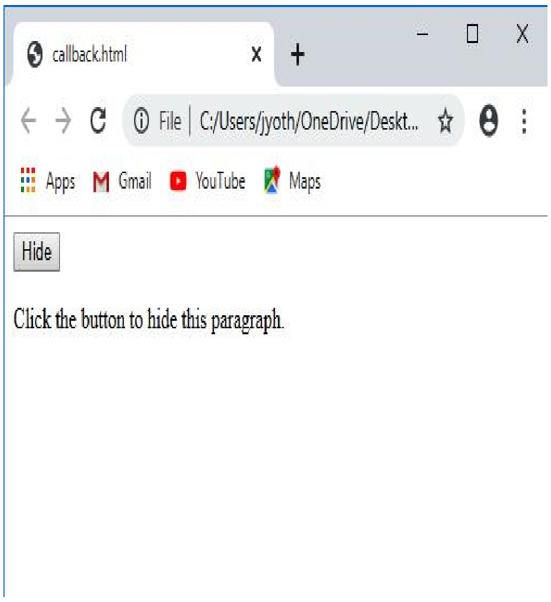
- Advantages:
  - JavaScript statements are executed line by line.
  - However, with effects, the next line of code can be run even though the effect is not finished. This can create errors.

Program for callback functions:

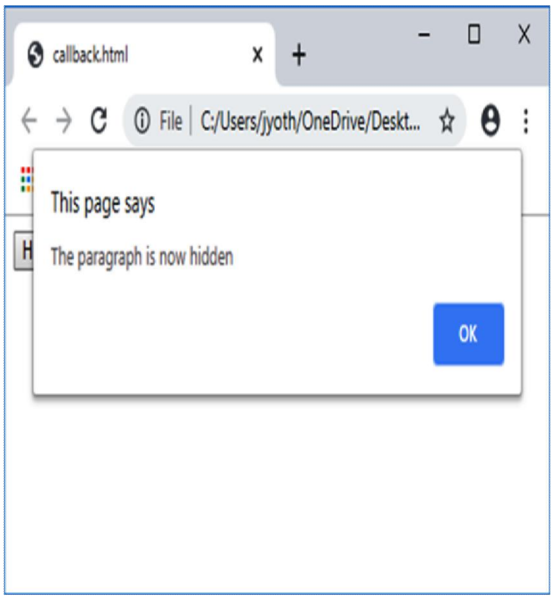
```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js
"> </script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide("slow", function(){
      alert("The paragraph is now hidden");
    });
  });
});
```

```
</script>
</head>
<body>
<button>Hide</button>
<p>Click the button to hide this paragraph.</p>
</body>
</html>
```

Output:



Click the button to hide this paragraph.



This page says

The paragraph is now hidden

1.4.6. Chaining:

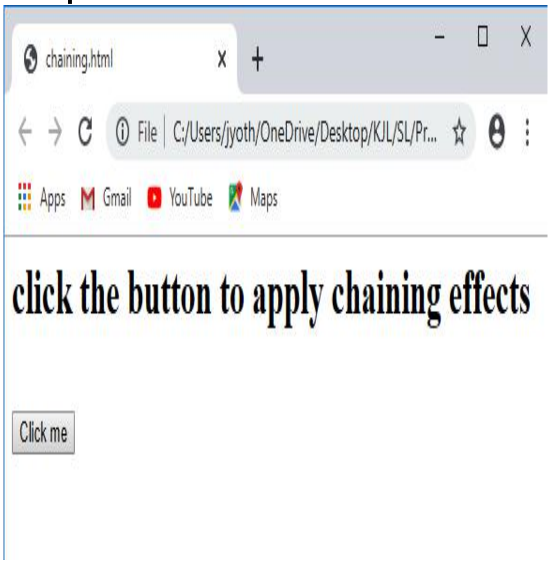
- Chaining allows us to run multiple jQuery commands, one after the other, on the same element(s) within a single statement
- To chain an action, simply append the action to the previous action.
- **Example:**
    - $(“#p1").slideUp(2000).slideDown(2000);
    - Element with id p1 first slides up, and then it slides down

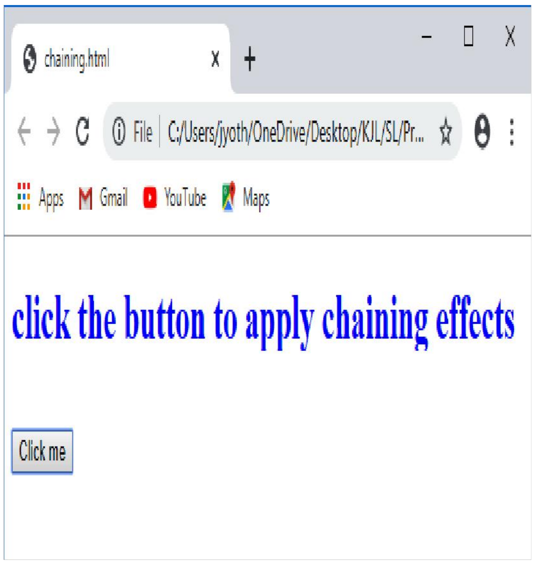
Example:

```
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min
.js"></script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("#p1").css("color", "blue").slideUp(2000).slideDown(2000);
  });
});
</script>
```

```
</head>
<body>
<h1 id="p1">click the button to apply chaining effects</h1><br>
<button>Click me</button>
</body>
</html>
```

Output:


chaining.html
File C:/Users/jyoth/OneDrive/Desktop/KJL/SL/Pr...
Apps
Gmail
YouTube
Maps
click the button to apply chaining effects
Click me


chaining.html
File C:/Users/jyoth/OneDrive/Desktop/KJL/SL/Pr...
Apps
Gmail
YouTube
Maps
click the button to apply chaining effects
Click me

1.5. MANIPULATING HTML USING JQUERY:

- jQuery contains powerful methods for changing and manipulating HTML elements and attributes.
- DOM = Document Object Model
  "The DOM defines a standard for accessing HTML and XML documents"
- jQuery comes with a bunch of DOM related methods that make it easy to access and manipulate elements and attributes.
- **html():**
  - The html() method sets or returns the content of the selected elements.
  - When this method is used to **return** content, it returns the content of the FIRST matched element.
  - When this method is used to **set** content, it overwrites the content of ALL matched elements.
  - **Syntax:**
    - Return content: `$(selector).html()`
    - Set content: `$(selector).html(`*`content`*`)`

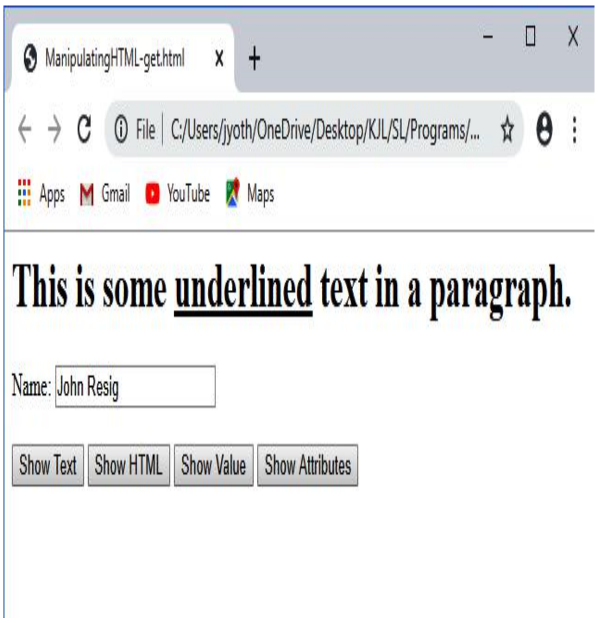
- **text():**
    - The text() method sets or returns the text content of the selected elements.
    - When this method is used to **return** content, it returns the text content of all matched elements (HTML markup will be removed).
    - When this method is used to **set** content, it overwrites the content of ALL matched elements.
- **Val():**
    - The val() method sets or returns the value attribute of the selected elements.
    - When this method is used to **return** value, it returns the value of the value attribute of the FIRST matched element.
    - When this method is used to **set** value, it returns the value of the value attribute for ALL matched elements.
    - **Syntax:**
        - Return attribute value: `$(`*`selector`*`).val()`
        - Set attribute value: `$(`*`selector`*`).val(`*`value`*`)`
- **Attr():**
    - The attr() method sets or returns the attributes and values of the selected elements.
    - When this method is used to **return** the attribute value of the FIRST matched element.
    - When this method is used to **set** one or more attribute/value pairs for the set of matched elements.
    - **Syntax:**
        - Returns value of an attribute: `$(“selector").attr(“attributename”);`
        - Sets value of an attribute:
        - `$(“selector").attr(“attributename”, “value”);`
        - `$(`*`selector`*`).attr({`*`attribute`*`:`*`value`*`, `*`attribute`*`:`*`value`*`,...}`

- **Manipulating HTML - get:**

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

```
<script>
$(document).ready(function(){
  $("#b1").click(function(){
    alert("Text: " + $("h1").text());
  });
  $("#b2").click(function(){
    alert("HTML: " + $("h1").html());
  });
  $("#b3").click(function(){
    alert("Value: " + $("input").val());
  });
  $("#b4").click(function(){
    alert("Attribute Name:"+$("input").attr("name"));
  });
});
</script>
</head>
<body>
<h1>This is some <u>underlined</u> text in a paragraph.</h1>
Name: <input type="text" name="t1" value="John Resig"><br><br>
<button id="b1">Show Text</button>
<button id="b2">Show HTML</button>
<button id="b3">Show Value</button>
<button id="b4">Show Attributes</button>
</body>
</html>
```

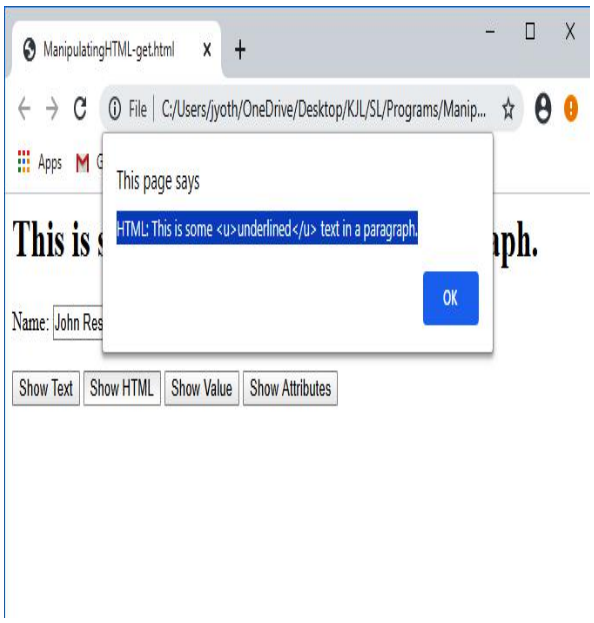
Output:


ManipulatingHTML-get.html
File | C:/Users/jyoth/OneDrive/Desktop/KJL/SL/Programs/...
Apps
Gmail
YouTube
Maps

This is some underlined text in a paragraph.

Name: John Resig

Show Text Show HTML Show Value Show Attributes


ManipulatingHTML-get.html
File | C:/Users/jyoth/OneDrive/Desktop/KJL/SL/Programs/Manip...
Apps

This page says

HTML: This is some <u>underlined</u> text in a paragraph.

OK

Name: John Res

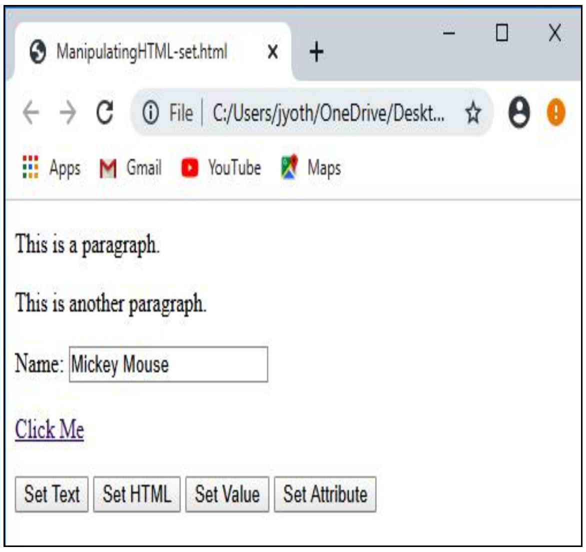
Show Text Show HTML Show Value Show Attributes

- **Manipulating HTML – Set:**

```
<html>
<head>
```

```
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.mi
n.js"></script>
<script>
$(document).ready(function(){
  $("#b1").click(function(){
    $("#t1").text("Hello world!");
  });
  $("#b2").click(function(){
    $("#t2").html("<b>Text was Updated</b>");
  });
  $("#b3").click(function(){
    $("#t3").val("John Resig");
  });
   $("#b4").click(function(){
    $("a").attr("href","https://www.google.com");
  });
});
</script>
</head>
<body>
<p id="t1">This is a paragraph.</p>
<p id="t2">This is another paragraph.</p>
Name: <input type="text" id="t3" value="Mickey Mouse"><br><br>
<a target="" href=""> Click Me </a><br><br>
<button id="b1">Set Text</button>
<button id="b2">Set HTML</button>
<button id="b3">Set Value</button>
<button id="b4">Set Attribute</button>
</body>
</html>
```

Output:

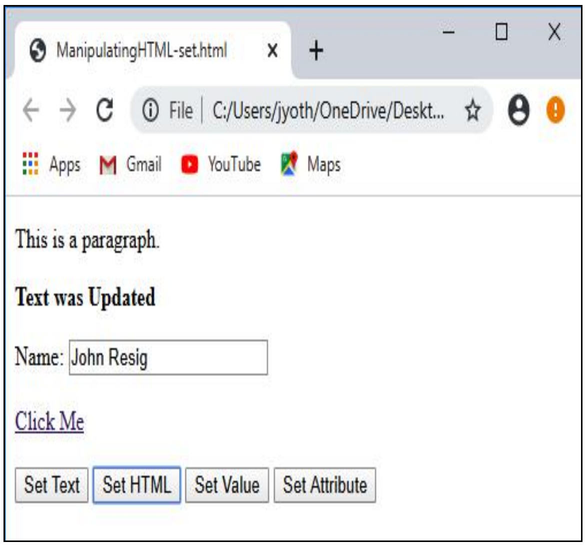

ManipulatingHTML-set.html
File | C:/Users/jyoth/OneDrive/Deskt...
Apps
Gmail
YouTube
Maps
Set Text
Set HTML
Set Value
Set Attribute

This is a paragraph.

This is another paragraph.

Name: Mickey Mouse

Click Me


ManipulatingHTML-set.html
File | C:/Users/jyoth/OneDrive/Deskt...
Apps
Gmail
YouTube
Maps
Set Text
Set HTML
Set Value
Set Attribute

This is a paragraph.

Text was Updated

Name: John Resig

Click Me

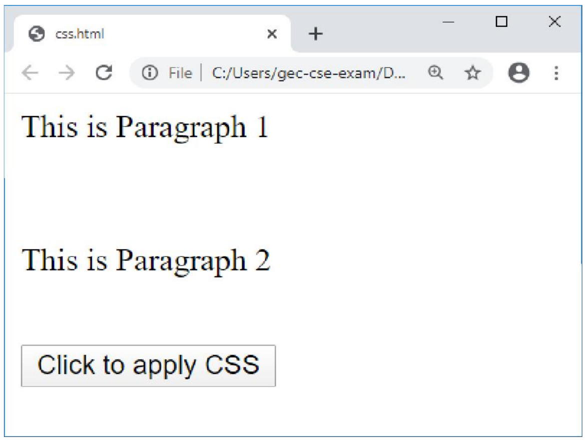
1.6. MANIPULATING CSS USING JQUERY:

- The css() method sets or returns one or more style properties for the selected elements.
- **Set a CSS Property:**
  - Syntax: `css("propertyname","value");`
  - Example: `$("p").css("background-color", "yellow");`
- **Set Multiple CSS Properties:**
  - Syntax:
    `css({"propertyname":"value","propertyname":"value",...});`
  - Example:
    `$("p").css({"background-color": "yellow", "font-size": "200%"});`
- **Return a CSS Property:** returns the value of the property associated with first occurrence of the matching element.
  - Syntax: `css("propertyname");`
  - Example: `$("p").css("background-color");`
- **Example:**

```
<html>
<head>
<script src="jquery-3.4.1.min.js">
</script>
<script>
   $(document).ready(function()
   {
         $("button").click(function()
         {
               $("#para1").css("background-color","pink");
               $("#para2").css({"background-color":"cyan","font-
         size":"30pt","text-decoration":"underline"});
               alert("Background color of
         Para1:"+$("p").css("background-color"));
         });
   });
</script>
</head>
<body>
<p id="para1">This is Paragraph 1</p>
<br>
<p id="para2">This is Paragraph 2</p>
<br>
```

```
<button>Click to apply CSS</button>
</body>
</html>
```

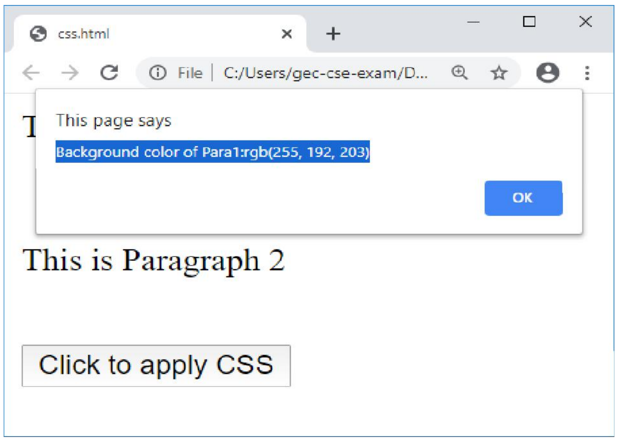
Output:



This is Paragraph 1

This is Paragraph 2

Click to apply CSS



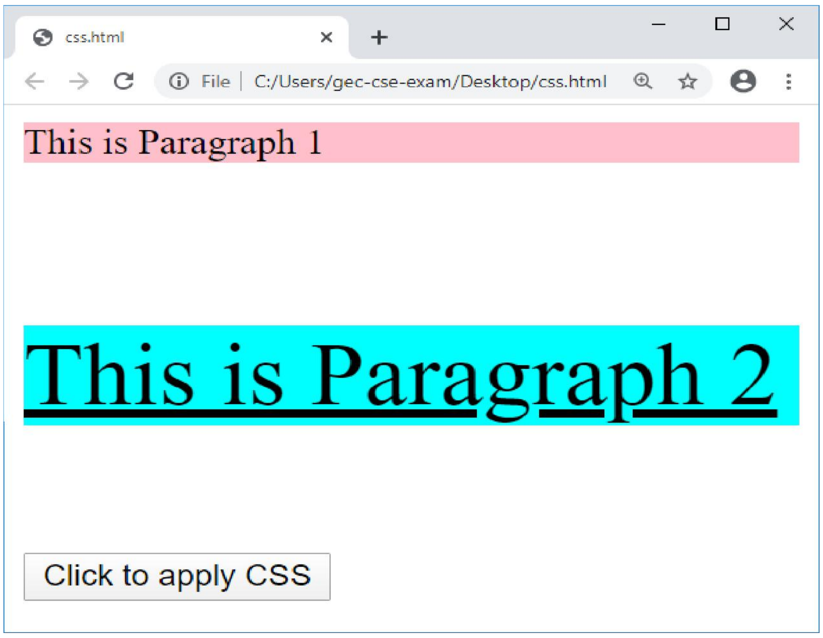
This page says

Background color of Para1:rgb(255, 192, 203)

OK

This is Paragraph 2

Click to apply CSS



This is Paragraph 1

This is Paragraph 2

Click to apply CSS

Differences between Javascript and JQuery:

JavaScript	jQuery
Client Side Scripting language	Client Side JavaScript library.
Definition: Scripting language used to create interactive Web pages.	*Definition:* jQuery is a lightweight, "write less, do more", JavaScript library.
Purpose: To convert static HTML pages to dynamic pages.	*Purpose:* To make it much easier to use JavaScript on your website.
Interpreted language – no need to compile JavaScript code.	Interpreted language – no need to compile jQuery code.
Suffers from browser compatibility issues.	Cross browser Compatible.
No need to add any additional plug-ins as all browsers supports JavaScript.	Need to include jQuery library URL in the head section of the page.
Need to write more lines of code.	Need to write fewer lines of code than JavaScript.
Cannot add effects and animations.	Can add effects and animations using jQuery library methods.

Syntax:
`<script>`
`    //JavaScript code`
`</script>`
	Syntax:
	`<script>`
	`    $(“selector”).action(function()`
	`    {`
	`        // jQuery code`
	`    });`
	`</script>`

Example:
`<html>`
`<head>`
`<script>`
`    alert("This is Javascipt");`
`</script>`
`</head>`
`<body>`
`</body>`
`</html>`
	Example:
	`<html>`
	`<head>`
	`<script src="jquery-3.4.1.min.js">`
	`</script>`
	`<script>`
	`$(document).ready(function()`
	`{`
	`    alert("This is jQuery");`
	`});`
	`</script>`
	`</head>`
	`<body>`
	`</body>`
	`</html>`