

JQuery part 1

Introduction to jQuery



jQuery is a JavaScript Library created by John Resig in 2006.

JQuery is a lightweight, open-source JavaScript library that simplifies interaction between HTML and JavaScript.

jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development.

Why jQuery?



Cross Browser Support

Extensibility through plug-ins

DOM manipulation

DOM Manipulation



Event Handling.

AJAX Processing



Creating effects / animations

Using jQuery Libraries



There are two ways to use jQuery:

Local Installation - Can download jQuery library on the local machine and include it into the HTML code

CDN Based Version - Can include jQuery library in the HTML code directly from Content Delivery Network (CDN)

Local Installation



- ❖ Download the latest version of Jquery script file from the following URL

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

- ❖ Version used - **Jquery 3.1.1**

- ❖ Copy the **JS file** into the needed location and refer that file within the **html code** of a web page like:

```
<script type="text/javascript" src="jquery/jquery-3.1.1.js"></script>
```

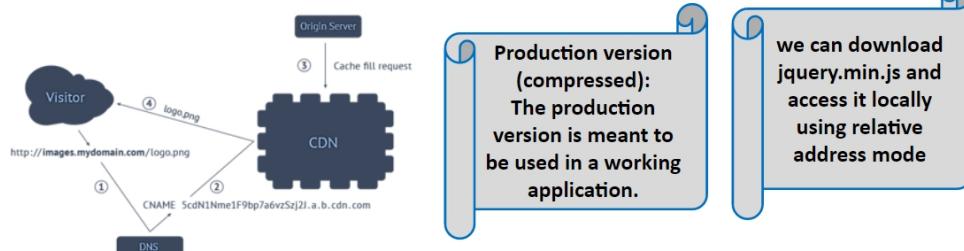
.js file is present inside the
folder jquery (relative path)

CDN Installation



- ❖ You can refer the jQuery library within your HTML code directly from Content Delivery Network (CDN)
- ❖ Both Google and Microsoft host jQuery files

```
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
```



Working with jQuery



All the JavaScript works that need to be performed, happens after the document is ready.

Example - adding events

```
$(document).ready(function()  
{  
    //JavaScript code  
});
```

Everything inside it will load as soon as the DOM is loaded, and before the page contents are loaded.

Example

```
<html>
<head>
<script type="text/javascript" src="jquery/jquery-3.1.1.js"></script>
<script type="text/javascript" >
$(document).ready(function()
{
    document.write("Welcome to JQuery");
});
</script>
</head>
<body>
</body>
</html>
```

Here, 'document' refers to the HTML element and ready() is the action on the selected element.



Output



JQUERY SELECTORS



Selectors are used to select one or more HTML elements using jQuery.

jQuery selectors start with the dollar sign and parentheses – \$()

Selectors can be :

- Tag name - \$('name of the tag') - \$('p')

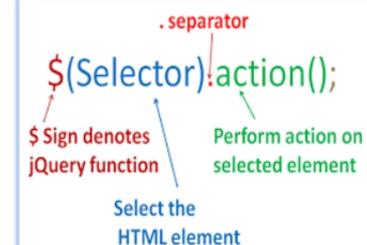
Selects all elements which match with the given element Name.

- Id - \$('#p_id')

Selects a single element which matches with the given ID.

- Class - \$('.p_class')

Selects all elements which match with the given Class.



The factory function \$() is a synonym of jQuery(). You can use function jQuery() instead of \$().

USING TAG NAME SELECTOR



```
<body>
<html>
<head>
<div class = "big" id = "div1">
<div>Inside First division</div>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
    $("div").css("background-color", "pink"); });
</script><div>"small" id = "div3">
</div>Inside Third division</p>
</div>
</body>
</html>
```

This would select all the div tags in the html file





USING TAG NAME SELECTOR

```
<body>
  <div class = "big" id = "div1">
    <p>Inside First Division</p>
  </div>
  <div class = "medium" id = "div2">
    <p>Inside Second Division</p>
  </div>
  <div class = "small" id = "div3">
    <p>Inside Third division</p>
  </div>
</body>
</html>
```



USING #ID SELECTOR

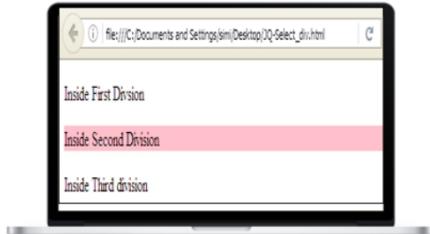
```
<html>
<head>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
  $("#div2").css("background-color", "pink"); });
</script>
</head>
```

This would select the div tag which has the id div2



USING #ID SELECTOR

```
<body>
  <div class = "big" id = "div1">
    <p>Inside First Division</p>
  </div>
  <div class = "medium" id = "div2">
    <p>Inside Second Division</p>
  </div>
  <div class = "small" id = "div3">
    <p>Inside Third division</p>
  </div>
</body>
</html>
```





USING .CLASS SELECTOR

```
<html>
<head>
<script type = "text/javascript" src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script type = "text/javascript" language = "javascript">
$(document).ready(function() {
    $(".big").css("background-color", "pink"); });
</script>
</head>
</html>
```

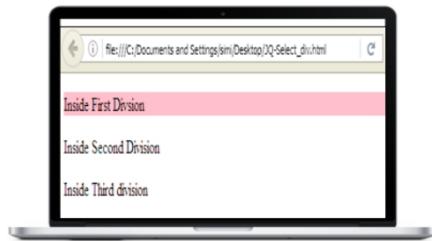
.css() is an action which sets the css attributes to the specified elements selected by the selector

This would select only the division which belongs to the class 'big'



USING .CLASS SELECTOR

```
<body>
<div class = "big" id = "div1">
    <p>Inside First Division</p>
</div>
<div class = "medium" id = "div2">
    <p>Inside Second Division</p>
</div>
<div class = "small" id = "div3">
    <p>Inside Third division</p>
</div>
</body>
</html>
```



USING ATTRIBUTE SELECTOR

The attribute selector is used to select an element by one of its HTML attributes, such as a link's target attribute or an input's type attribute, etc.

It selects elements that have the specified attribute with a value exactly equal to a certain value

Example: The following jQuery code will select and highlight all the text inputs :

```
$('.input[type="text"])'
```

<input> elements with the type="text", when the document is ready.

USING ATTRIBUTE SELECTOR

```
<html> <head>
<<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"
>/>
```

```
<script type="text/javascript">
$(document).ready(function(){
    // Highlight paragraph elements
    $('input[type="text"]').css("background", "yellow");
});
</script> </head>
<body> <form>
    <label>User Name: <input type="text"></label>
    <label>Password: <input type="password"></label>
    <input type="submit" value="Sign In">
</form> </body> </html>
```



`.css()` is an action which sets the css attributes to the specified elements selected by the selector

FORM ELEMENT SELECTORS

01 INPUT

This selector basically selects all form controls
Selects all input, textarea, select and button elements

02 CHECKBOX

Selects all elements of type checkbox

03 RADIO SELECTOR

Selects all elements of type radio

04 SUBMIT SELECTOR

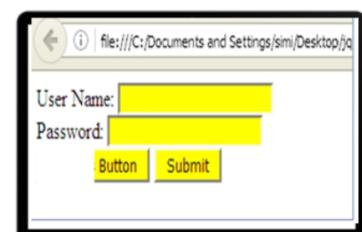
Selects all elements of type submit

04 TEXT SELECTOR

Selects all input elements of type text.

USING ELEMENT SELECTOR

```
<html><head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
    $(":input").css("background-color", "yellow"); });
</script> </head>
<body> <form action="">
    User Name: <input type="text" name="user"><br>
    Password: <input type="password" name="password"><br>
    <button type="button">Button</button>
    <input type="submit" value="Submit"><br>
</form> </body> </html>
```



OTHER JQUERY SELECTORS



Selector	Description
<code>\$('*')</code>	This selector selects all elements in the document
<code>\$('p > *')</code>	This selector selects all elements that are children of a paragraph element
<code>\$('li:not(.myclass)')</code>	Selects all elements matched by <code></code> that do not have class="myclass"
<code>\$('p a.specialClass')</code>	This selector matches links with a class of specialClass declared within <code><p></code> elements
<code>\$('ul li:first')</code>	This selector gets only the first <code></code> element of the <code></code> .
<code>\$(':empty')</code>	Selects all elements that have no children
<code>\$('p:empty')</code>	Selects all elements matched by <code><p></code> that have no children
<code>\$('div[p]')</code>	Selects all elements matched by <code><div></code> that contain an element matched by <code><p></code>

OTHER JQUERY SELECTORS



Selector	Description
<code>\$("li:even")</code>	Selects all elements matched by <code></code> that have an even index value.
<code>\$("tr:odd")</code>	Selects all elements matched by <code><tr></code> that have an odd index value.
<code>\$("li:first")</code>	Selects the first <code></code> element.
<code>\$("li:last")</code>	Selects the last <code></code> element.
<code>\$("li:visible")</code>	Selects all elements matched by <code></code> that are visible.
<code>\$("li:hidden")</code>	Selects all elements matched by <code></code> that are hidden.
<code>\$(":radio")</code>	Selects all radio buttons in the form.
<code>\$(":checked")</code>	Selects all checked boxes in the form.

JQUERY SELECTOR:FILTER



Filter()

This method returns elements that match a certain criteria. Elements that do not match the criteria are removed from the selection, and those that match will be returned.

```
<html><head>
<script src=""
https://ajax.googleapis.com/ajax/libs/jquery/3.1.1
/jquery.min.js "></script>
<script>
$(document).ready(function(){
  $("li").filter( ":even" ).css(
    "background-color", "red" );
  $("p").filter(".intro").css("background-
color", "yellow");
});
```

This function filters even the elements in the list and paragraphs which has the id "intro" and changes the background color to red and yellow respectively

JQUERY SELECTOR:FILTER



```
<body>
<h1>Welcome to My Homepage</h1>
<p>Esther Saradha</p>
<p class="intro">Corporate Trainer</p>
<p>My Favourite Foods:</p>

<ul>
<li>Pizza</li>
<li>Club Sandwich</li>
<li>Orange juice Sandwich</li>
<li>Hamburger</li>
</ul>
</body>
<html>
```

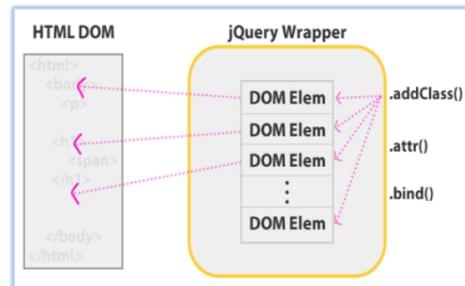


JQUERY DOM



jQuery contains methods for changing and manipulating DOM elements and attributes

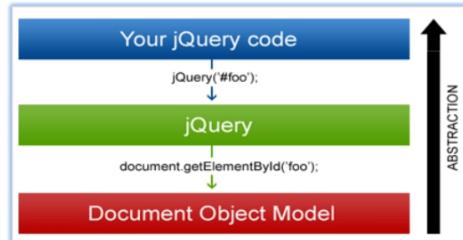
- `text()` - Sets or returns the text content of selected elements
- `html()` - Sets or returns the content of selected elements (including HTML markup)
- `val()` - Sets or returns the value of form fields



JQUERY HTML : VAL()



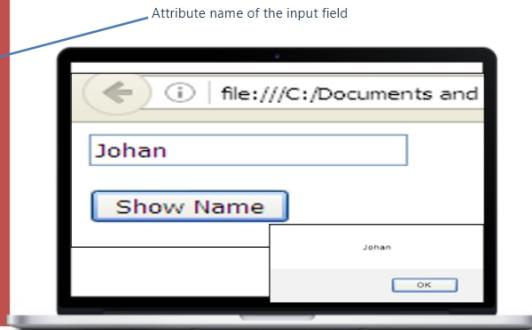
```
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert("Name Value: " + $("#name").val());
    });
});
</script>
</head>
<body>
    Name: <input type="text" id="name" value="Krishna">
    <button>Show Value</button>
</body>
```



JQUERY HTML: ATTR METHOD



```
<html>
<head>
<script>
$(document).ready(function(){
    $("button").click(function(){
        alert($("#name").attr("value"));
    });
});
</script>
</head>
<body>
<p><input type="text" id="name" value="Johan">
</p>
<button>Show Name</button>
</body>
</html>
```



JQUERY HTML: SET DATA



```
<script>
$(document).ready(function(){
    $("#b1").click(function(){
        $("#p1").text("Clicked the 'Set Text' Button");
    });
    $("#b2").click(function(){
        $("#p2").html("<font color='red'><b>Clicked the 'Set HTML' Button</b></font>");
    });
    $("#b3").click(function(){
        $("#name").val("Miní");
    });
});</script>

<body>
<p id="p1">This is a paragraph.</p>
<p id="p2">This is another paragraph.</p>
<p>Input field: <input type="text" id="name" value="Johan"></p>
<button id="b1">Set Text</button>
<button id="b2">Set HTML</button>
<button id="b3">Set Value</button>
</body>
```



JQUERY DOM INSERTIONS



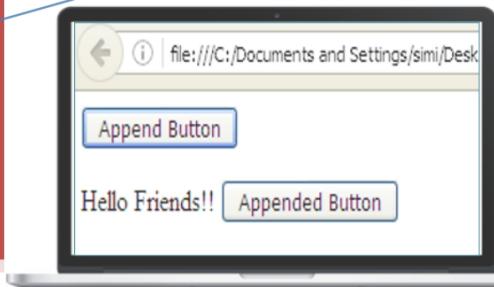
- 1 APPEND()
Inserts content at the end of the selected elements
- 2 PREPEND()
Inserts content at the beginning of the selected elements
- 3 AFTER()
Inserts content after the selected elements
- 4 BEFORE()
Inserts content before the selected elements

JQUERY HTML: APPEND



```
<html><head>
<head>
<script type="text/javascript" src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>(function(){
$(document).ready(function(){
    $("button").click(function(){
        $("p").append("<button>Appended
Button</button>");
    });
});
</script></head>
</head><body>
<button id="btn1">Append Button</button>
<button id="btn2">Hello Friends!!</p><button>
</body> </html>
```

This line will add the button after the <p> tag



JQUERY HTML: PREPEND



```
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#btn2").click(function(){
        $("ol").prepend("<li>ME</li>");
    });
});
</script> </head>
<body>
<ol>
<li> BE</li>
<li> BTech</li>
<li> MTech</li>
</ol>
<button id="btn2">Prepend list item</button>
</body> </html>
```

JQUERY HTML : PREPEND



```
html><head>
<style>
    p { background: yellow; }
</style>
<head> <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
    $("#b1").click(function(){
        $("p").before("<b>Hello</b>");
    });
});
</script></head>
<body>
<p>I would like to say: <br>
<button id="b1"> Insert 'Hello' Before
Paragraph</button>
</body> </html>
```

This function will add "Hello" at the beginning of the paragraph when you click the button



JQUERY HTML : PREPEND



```
<html><head>
<style>
p { background: yellow; }
</style>
<head> <script>
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1
.1/jquery.min.js"</script>
<script>
$(document).ready(function(){
    $("#b2").click(function(){
        $("p").after("<b>Hello</b>");
    });
});
</script></head>
<body> <p>I would like to say: </p>
<button id="b2"> Insert 'Hello' After
Paragraph</button>
</body></html>
```

This function will add "Hello" at the end of the paragraph when you click the button



JQUERY DOM REMOVAL



Remove Elements/Content

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

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $("#name").remove();
    });
});
</script>
</head>
<body>
<input type="text" id="name">
<button>Remove text box</button>
</body>
```

This will remove the text field when you click on the button

ATTACH EVENTS FROM DOM ELEMENTS



01 Using the jQuery Event Model, we can establish event handlers on DOM elements with the `on()` command

02 `.on()` method , attaches an event handler function for one or more events to the selected elements.

syntax:
`$([Selector/listener]).on(eventName [, selector] [, data])`

ATTACH EVENTS FROM DOM ELEMENTS



```
jQuery also provides a shortcut method for most  
common browser and Ajax events.  
jQuery( listener/selector).eventName(  
    handlerFunction)
```

Example: Attaching a click event, with and without
the shortcut:
// Using on()
jQuery('div').on('click',function(e){...});

// Using the shortcut
jQuery('div').click(function(e){...});

JQUERY HTML : PREPEND



```
<!DOCTYPE html>  
<html>  
<head>  
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.  
min.js"></script>  
<script>  
$(document).ready(function(){  
    $("p").on("click", function(){  
        alert("Welcome Everybody");  
    });  
});  
</script>  
</head>  
<body>  
<p>Click on the statement.</p>  
</body> </html>
```

When you click on the paragraph, it shows an
alert box
With the message " Welcome Everybody"

ATTACH EVENTS FROM DOM ELEMENTS



```
Attaching a Handler to Many Events:  
<!doctype html> <html lang="en"><head>  
<style>  
.test {  
color: #000;  
padding: .5em;  
border: 1px solid #444;  
}  
.active {  
color: #900;  
}  
.inside {  
background-color: aqua;  
} </style>
```

ATTACH EVENTS FROM DOM ELEMENTS



```
<script>
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/j
query.min.js"></script>
</head>
<body>
<div class="test">Hi Everyone!!!</div>
<script>
$( "div.test" ).on({
  click: function() {
    $( this ).toggleClass( "active" );
  }, mouseenter: function() {
    $( this ).addClass( "inside" );
  }, mouseleave: function() {
    $( this ).removeClass( "inside" );
  }
});</script> </body></html>
```



DETACH EVENTS FROM DOM ELEMENTS



off() : This method removes event handlers that were attached with .on()

Calling .off() with no arguments it removes all handlers attached to the elements

Syntax: .off(events [, selector] [, handler])

Events : One or more space-separated event types

Handler: A handler function previously attached for the event(s), or the special value false.

Selector : A selector which should match the one originally passed to .on() when attaching event handlers.

ATTACH EVENTS FROM DOM ELEMENTS



```
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jqu
ery.min.js">
</script>
<script>
function changeSize() {
  $(this).animate({fontSize: "+=10px"});
}
$(document).ready(function(){
  $("p").on("click", changeSize);
  $("button").click(function(){
    $("p").off("click", changeSize);
  });
});</script>
</head>
<body>
<p>Write Less, Do More</p>
<button>Remove the changeSize() event handler</button>
</body></html>
```

In this, when we click on the paragraph, its size increases and once you click the button, the changeSize() event handler is removed from the paragraph element.



jQuery part 2

JQUERY EVENTS



- Events are user's actions
- jQuery events are the actions that can be detected by your web application.
- The term "fires/fired" is often used with events.
- The click event is fired, the moment you click a button



JQUERY 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

EVENTS: SAMPLE PROGRAM

```
<!DOCTYPE html>
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/
jquery/3.1.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $("p").hide();
  });
});
</script></head>
<body>
<h2><font color="red"> Event
Header</font></h2>
<p>Your Only Limit is YOU.</p>
<p>Mistakes are proof that you are
TRYING.</p>
<button>Click me to hide
paragraphs</button></body></html>
```

Event Header

Your Only Limit is YOU.

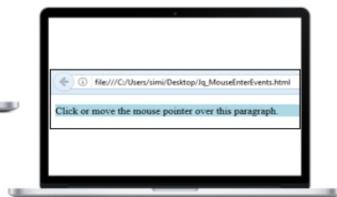
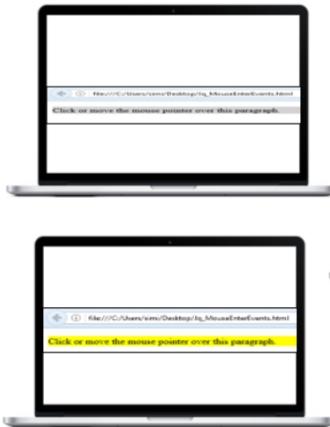
Mistakes are proof that you are TRYING.

Click me to hide paragraphs

EVENTS: SAMPLE PROGRAM



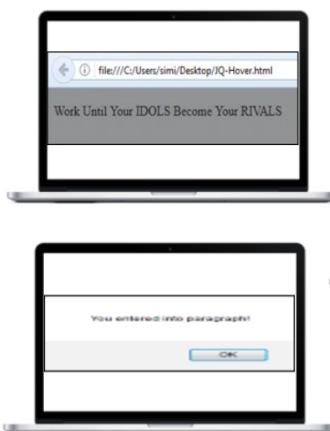
```
html><head>
<script>
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("p").on({
    mouseenter: function(){
      $(this).css("background-color",
"lightgray");
    },
    mouseleave: function(){
      $(this).css("background-color",
"lightblue");
    },
    click: function(){
      $(this).css("background-color",
"yellow");
    }
  });
</script></head> <body><p>Click or move the mouse pointer over this paragraph.</p></body></html>
```



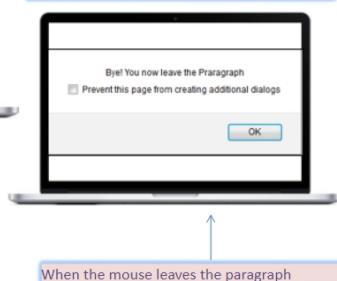
EVENTS: SAMPLE PROGRAM



```
<!DOCTYPE html>
<html><head>
<script>
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function(){
  $("#p1").hover(function(){
    alert("You entered into paragraph!");
  },
  function(){
    alert("Bye! You now leave the Paragraph");
  });
});</script></head>
<body>
<p id="p1">Work Until Your IDOLS Become Your RIVALS</p>
</body></html>
```



hover() method takes two functions and is a combination of the mouseenter() and mouseleave() methods.



When the mouse leaves the paragraph

JQUERY KEYBOARD EVENTS



1 KEYDOWN

The key is on its way down

2 KEYPRESS

The key is pressed down
The keypress() method triggers the keypress event, or attaches a function to run when a keypress event occurs
The keypress event is similar to the keydown event

3 KEYUP

The event occurs when a button is pressed down

The key is released

JQUERY KEYBOARD EVENTS



```
html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
</script>
<script>
i = 0;
$(document).ready(function(){
    $("input").keypress(function(){
        $("span").text(i += 1);
    });
});
</script>
</head><body>
Enter your name: <input type="text">
<p>Keypresses: <span>0</span></p>
</body></html>
```



JQUERY FORM EVENTS



```
<head>
<script type="text/javascript"
src=" https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js
"></script>
<script type="text/javascript" >
$(document).ready(function(){
    $('input').change(function(){
        var name=document.myForm.name.value;
        $('div').append("<h3>Welcome
"+name+"</h3>");
    });
});
</script> </head>

<body>
<form name="myForm">
    Name : <input type="text" name="name">
    <div></div>
</form>
</body>
```



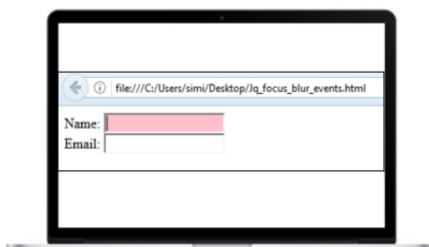
When you type /change the name in the input box, it displays the word Welcome' along with that name

JQUERY FORM EVENTS



```
<!DOCTYPE html>
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js">
</script>
$(document).ready(function(){
    $('input').focus(function(){
        $(this).css("background-color", "pink");
    });
    $('input').blur(function(){
        $(this).css("background-color", "#ffffff");
    });
});
</script></head>
<body>
Name: <input type="text" name="fullname"><br>
Email: <input type="text" name="email"></body></html>
```

Focus() is executed when the form field gets focus.
blur() is executed when the form field loses focus.



When the mouse pointer focus on the input field, it turns to pink, when exits, turns to default color



JQUERY SCROLL EVENTS

```
<html><head>
<script src=
"//ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js
"></script>
<script>
$(document).ready(function(){
$(txtscroll).scroll(function(){
alert ("Scroll event occurred!");
});
x = 0;
$("div").scroll(function(){
$(sp+"").text( x+= 1);
}); });
</script></head>
<body><p>font color="Green"> TEXT AREA </font></p>
<textarea cols="20" rows="10" id="txtscroll">
jQuery is a fast, small, and feature-rich JavaScript library.
<br><br>
```

The scroll event occurs when the user scrolls in the specified element
\$(selector).scroll(function)

JQUERY SCROLL EVENTS

It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

```
</textarea>
<p><font color="blue"> DIVISION AREA </font></p>
<div style="border:1px solid black; width:200px; height:100px;
overflow:scroll;">
jQuery is a fast, small, and feature-rich JavaScript library.
<br><br>
It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
</div>
<p><font color="red">Scrolled <span>0</span>
times.</font></p>
</body></html>
```



JQUERY RESIZE EVENTS



The resize event happens as browser window is resized.



This is useful for many situations like adjusting page elements depending on window size in order to avoid scroll bars ,etc.



The resize() method triggers the resize event, or attaches a function to run when a resize event occurs

Syntax:

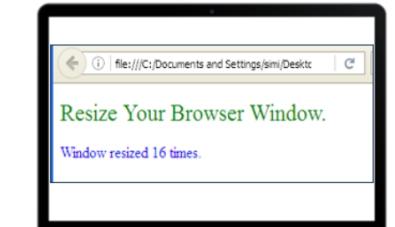
```
$(selector).resize()
$(selector).resize(handler function)
```



JQUERY RESIZE EVENTS



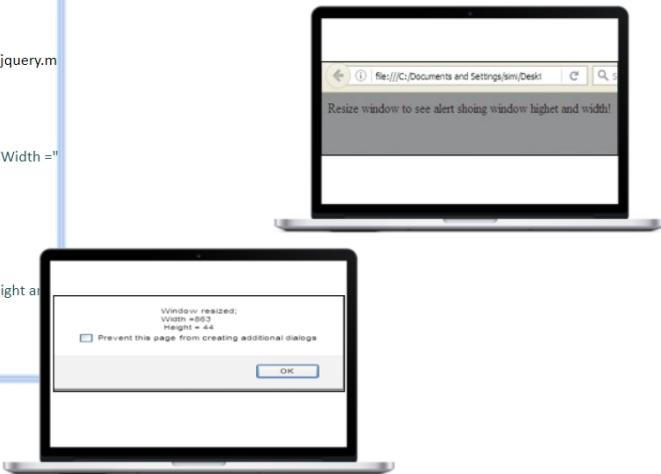
```
<!DOCTYPE html>
<html><head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
x = 0;
$(document).ready(function(){
    $(window).resize(function(){
        $("span").text(x += 1);
    });
});</script> </head>
<body><p><font color="green" size="5">Resize Your Browser Window.</font></p>
<p><font color="blue"> Window resized <span>0</span> times.</font></p></body> </html>
```



JQUERY RESIZE



```
<html><head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.m
in.js"></script>
<script>
$(document).ready(function(){
    $(window).resize(function(){
        alert ("Window resized;" + "\n" + "Width =" +
+ $(window).width() + "\n" + "Height =" +
$(window).height());
    });
});
</script> </head>
<body>
<p>Resize window to see alert showing window height and width!</p>
</body>
</html>
```



Mouse Events



```
<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>
<script>
$(document).ready(function()
{
    $("#p1").hover(function(){
        alert("You entered into paragraph!");
    },
    function(){
        alert("Bye! You now leave the Paragraph");
    });
});</script>
```

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

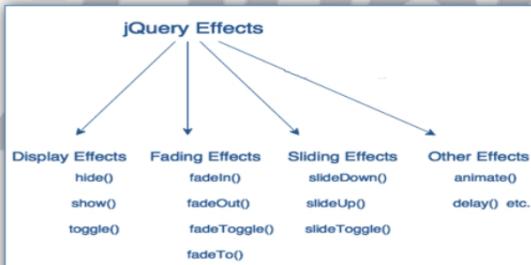
hover() method takes two functions and is a combination of the **mouseenter()** and **mouseleave()** methods.

jQuery Effects



jQuery enables us to add effects on a web page.

jQuery effects can be categorized into display, fading, sliding, hiding/showing and other animation effects.



Display Effects



```
<script type="text/javascript" >
$(document).ready(function()
{
    $('#hide').click(function(){
        $('img').hide();
    });
    $('#show').click(function(){
        $('img').show();
    });
});
</script>
```



Hide and show can take an optional time in milliseconds to have transition

```
<script type = "text/javascript" language = "javascript">
$(document).ready(function()
{
    $(".clickme").click(function(event){
        $(".target").toggle('slow', function(){
            });
        });
});
</script>
```



After clicking the hide button, image is hidden

toggles() method toggles between the hide() and show() methods.



Fading Effects



fadeIn() - Syntax

```
$(selector).fadeIn([speed, callback]);
```

Specifies the duration of the effect

fadeOut() - Syntax

```
$(selector).fadeOut([speed, callback]);
```

It is a function to be executed after fading completes

fadeToggle() - Syntax

```
$(selector).fadeToggle([speed, callback]);
```

fadeTo() - Syntax

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

Opacity value between 0 and 1

Fading Effects - Example



```
<script type="text/javascript" >
$(document).ready(function() {
    $('#fadeout').click(function(){
        $('h3').fadeOut(1000);
    });
    $('#fadein').click(function(){
        $('h3').fadeIn(1000);
    });
    $('#fadeto').click(function(){
        $('h3').fadeTo(1000,.5);
    });
});
</script>
```

Output



Sliding Effects



slideDown() - Syntax

```
$(selector).slideDown([speed, callback]);
```

slideUp() - Syntax

```
$(selector).slideUp([speed, callback]);
```

slideToggle() - Syntax

```
$(selector).slideToggle([speed, callback]);
```

Sliding Effects - Example



```
<script type="text/javascript" >
$(document).ready(function()
{
    $('#slidedown').click(function(){
        $('h3').slideDown("slow");
    });
    $('#slideup').click(function(){
        $('h3').slideUp("fast");
    });
    $('#slidetoggle').click(function(){
        $('h3').slideToggle(1000);
    });
});
</script>
```

Output



animate() - Example

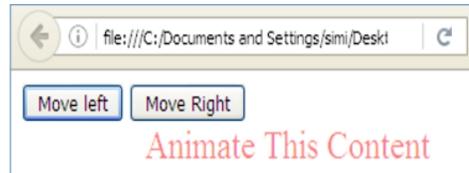


```
<!DOCTYPE html>
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
$(document).ready(function()
{
    $('#left').click(function()
    {
        $('div').animate( {left: "-=50px" , height:'toggle', opacity:0.5}, "slow");
    });
    $('#right').click(function(){
        $('div').animate( {left: "+=50px", width : 'toggle',opacity:1}, "slow");
    });
});
</script>
<head>
<body>
    <button id="left">Move left</button>
    <button id="right">Move Right</button>
    <div style="left: 100px ; position: absolute ">
        <font size=5 color="red">Animate This Content</font></div>
</body>
</html>
```

Syntax

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

Output



SlideShow - Example



```
<html>
<head>
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script>
function slideSwitch() {
    var $active = $('#slideshow IMG.active');
    if ( $active.length == 0 )
        $active = $('#slideshow IMG:last');
    var $next = $active.next().length ? $active.next() : $('#slideshow IMG:first');
    $active.addClass('last-active');
    $next.css({opacity: 0.0}).addClass('active') .animate({opacity: 1.0}, 1000,
    );
    $next.click( "slideSwitch()", 1000 );
}
</script>
<style type="text/css">
    .active{
        z-index:99;
    }
</style>
</head>
<body>
    <div id="slideshow">
        
        
        
        
    </div>
</body>
</html>
```

Output



JQUERY WITH JSON



JSON

JSON stands for "JavaScript Object Notation".

- It is called so because storing data with JSON creates a JavaScript object.

It is a syntax for storing and exchanging data.

It is lightweight data-interchange format

"self-describing" and easy to understand

It is language independent

WHY USE JSON



STANDARD STRUCTURE

JSON objects are having a standard structure that makes developers job easy to read and write code.

LIGHT WEIGHT

JSON is light weighted, it becomes easier to get and load the requested data quickly

SCALABLE

JSON is language independent



JSON SYNTAX RULES



JSON syntax is derived from JavaScript object notation syntax:

- JSON Data is in name/value pairs
- Data is separated by commas
- Curly braces hold objects
- Square brackets hold arrays



A name/value pair consists of a field name (in double quotes), followed by a colon, followed by a value:

Example : "name":"John"

JSON DATA TYPES



STRING

Strings in JSON must be written in double quotes.

NUMBER

Numbers in JSON must be an integer or a floating point

DATA TYPES

NULL

Values in JSON can be null.

BOOLEAN

Values in JSON can be true/false.

```
* Numbers { "id": 5, "balance":2942.59 }
* Strings { "name": "Joe", "city": "Morrisville" }
* Booleans { "premium": true, "balance":2942.59 }
* Null { "last_address": null }
* Arrays { "hobbies": ["tennis", "skiing", "Judo"] }
* Objects { address: { "street": "1, Main street", "city": Morrisville, "state": "CA", "zip": "94024" }}
```

JSON OBJECTS



A JavaScript object is an unordered set of name/value pairs

- 01 The pairs are enclosed within braces { }
- 02 Keys must be strings, and values must be a valid JSON data type (string, number, object, array, Boolean or null).
- 01 There is a colon between the name and the value
- 02 Pairs are separated by commas

```
{  
  "no": "1",  
  "name": "Sam",  
  "gender": "male",  
  "phone": "555  
1234567"  
}
```

ACCESSING JSON OBJECTS



USING DOT(.) NOTATION

Example

```
myObj = { "name": "Sathish",  
          "age": 20, "car": null };  
x = myObj.name;
```

USING BRACKET ([]) NOTATION

Example

```
myObj = { "name": "Sathish",  
          "age": 20, "car": null };  
x = myObj["name"];
```

NESTED JSON OBJECTS



NESTED JSON OBJECTS :
VALUES IN A JSON OBJECT
CAN BE ANOTHER JSON
OBJECT.



```
myObj = {  
  "name": "Sathish",  
  "age": 20,  
  "cars": {  
    "car1": "Ford",  
    "car2": "BMW",  
    "car3": "Fiat"  
  }  
}
```

Example

```
x = myObj.cars.car2;  
//or:  
x = myObj.cars["car2"];
```

MODIFY/DELETE OBJECTS



Modify Object Values : You can use the dot notation to modify any value in a JSON object.

Example:

```
myObj.cars.car2 = "Mercedes";  
myObj.cars["car2"] = "Mercedes";
```

Delete Object Properties: Use the 'delete' keyword to delete properties from a JSON object.

Example:

```
delete myObj.cars.car2;
```

JSON ARRAYS



array is an ordered collection of values

- The values are enclosed within brackets []
- Values are separated by commas

Example:

```
[{  
    "name": "html",  
    "years": 5  
,  
    {  
        "name": "jquery",  
        "years": 6  
    } ]
```

USING JSON IN JQUERY



jQuery.getJSON(url, [data], [callback]) :

- jQuery helps to load JSON-encoded data from a remote source using \$.getJSON()
 - This jquery method loads JSON-encoded data from the server using a GET HTTP request
 - The method returns XMLHttpRequest object
- Syntax: \$.getJSON(url, [data], [callback])

The three parameters correspond to:

01

URL

This parameter is a string containing the URL to which the request is sent.

02

DATA

The optional data parameter is either an object or a string that is sent to the server with the request.

03

CALL BACK

This optional parameter represents a function to be executed whenever the data is loaded successfully.

USING JSON IN JQUERY

Example: for JSON file: bookData.json

```
{
  "books": [
    {
      "bookId": "B101",
      "bookName": "C++ Programming",
      "bookPrice":7890.80
    },
    {
      "bookId": "B102",
      "bookName": "Java Programming",
      "bookPrice":800
    },
    {
      "bookId": "B103",
      "bookName": "Python Programming",
      "bookPrice":600
    }
  ]
}
```

\$.each(array, callback):
A generic iterator function, which can be used to iterate over both objects and arrays.

USING TAG NAME SELECTOR

```
<html> <head>
<script type = "text/javascript"
src =
"https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>
<script type="text/javascript">
$(function(){
  $.getJSON('bookData.json',function(data){
    $.each(data.books,function(key,val){
      $('#div').append('<p>' +val.bookId+ " "+val.bookName+ " "+val.bookPrice+
'</p>');
    });
  });
</script> </head>
<body>
<div></div>
</body>
</html>
```

