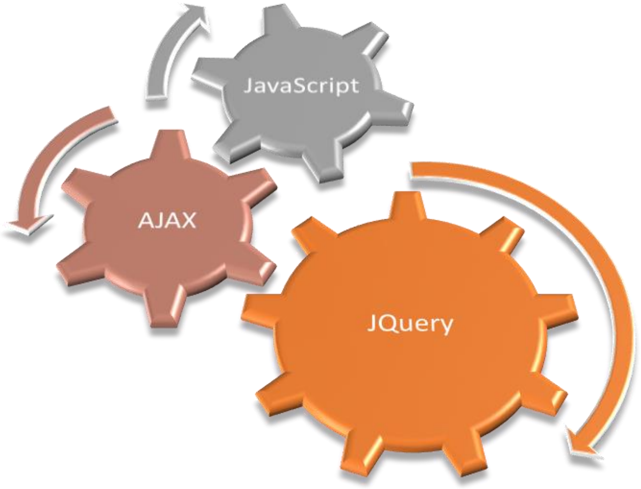
8/12/2018

JQuery & AJAX

From Zero to Hero

Mr. Naveen Saggam

JQuery & AJAX



JQuery

JQuery is a very light weight JavaScript library and the main purpose of JQuery is, it makes much easier to use JavaScript in our webpages.

The biggest feature and main motto of JQuery is “Write less and do more”.

In order to provide a rich dynamic nature for any website we require JavaScript. May be by using JavaScript the code may goes complicated and that can be more simplified using JQuery.

Most of the common tasks of JavaScript may require more number of line of code which are wrapped into more methods in JQuery and they can be called by using single line of code of JQuery.

JQuery simplifies the most complicated things can be done by JavaScript, such as DOM Manipulation and AJAX calls.

Features of JQuery

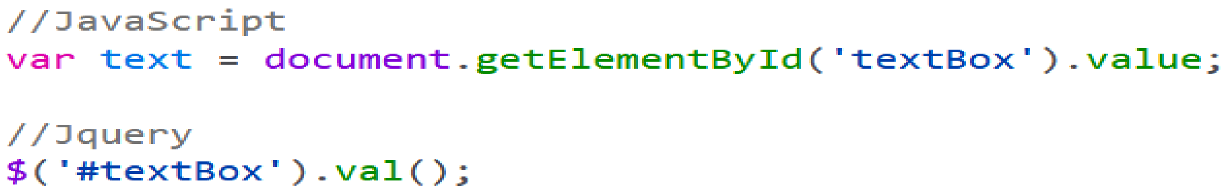
Using JQuery we can simplify the more complicated things can be done by using JavaScript. The main usage and features of JQuery are as follows,

1. HTML DOM Manipulation
2. CSS Manipulation
3. JavaScript event Handling methods.
4. Effects and Animations.
5. JQuery plugins for complicate and rich UI Components
6. Cross browser compatibility.

HTML DOM Manipulation

JQuery make easy of DOM manipulation. Using we can add or edit or delete DOM Nodes from the DOM Tree and provides a dynamic nature for the content.

Let’s understand the JavaScript way and JQuery way of DOM Manipulation.

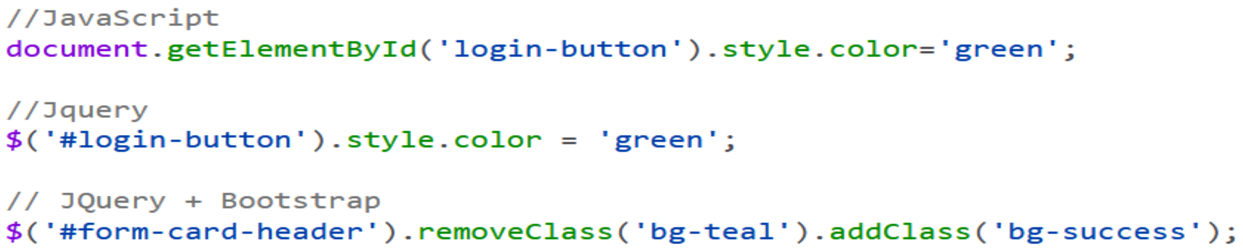


```
//JavaScript
var text = document.getElementById('textBox').value;

//Jquery
$('#textBox').val();
```

CSS Manipulation

By using the concept called DOM manipulation we can access any HTML and change its css properties using JavaScript dynamically. This even made easy using JQuery.

Let’s understand CSS Manipulation using JQuery and JavaScript.

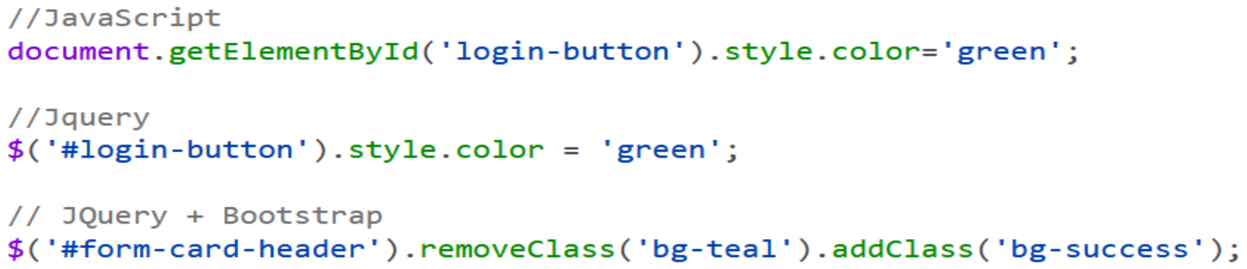


```
//JavaScript
document.getElementById('login-button').style.color='green';

//Jquery
$('#login-button').style.color = 'green';

// JQuery + Bootstrap
$('#form-card-header').removeClass('bg-teal').addClass('bg-success');
```

JavaScript Event Handling methods

JQuery offers an elegant way to capture a wide verity of events, user click on a button or click on a link or any kinds of events. By using JavaScript we may hook up the events and call the JavaScript functions for processing of an event. JQuery provides an easy way of event handing.

Let’s understand the event Handling using JavaScript and JQuery is as follows,

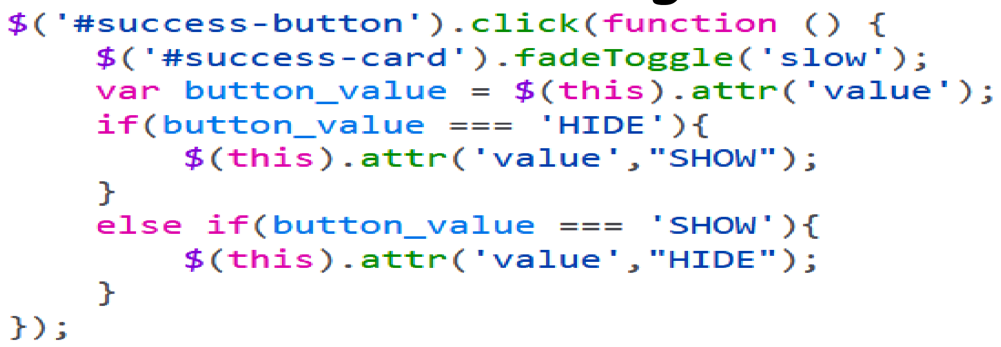


```
//JavaScript
document.getElementById('login-button').style.color='green';

//Jquery
$('#login-button').style.color = 'green';

// JQuery + Bootstrap
$('#form-card-header').removeClass('bg-teal').addClass('bg-success');
```

Effects and Animation

JQuery comes with plenty of effects and animations which we can use for our websites.

JQuery effects like hide/show, fade, and slide are more useful to build any rich UI Components. JQuery also comes along with lot of plugins for ready to use features in our websites.



```
$('#success-button').click(function () {
    $('#success-card').fadeToggle('slow');
    var button_value = $(this).attr('value');
    if(button_value === 'HIDE'){
        $(this).attr('value',"SHOW");
    }
    else if(button_value === 'SHOW'){
        $(this).attr('value',"HIDE");
    }
});
```

The above example code we use it for toggle effect for any buttons.

JQuery plugins

A plug-in is a piece of JavaScript code written in a standard JavaScript file. These files provides useful JQuery methods which can be used along with JQuery library methods.

These are ready to use plugins and in order to make use of those plugins, first we needs to add those plugin related JavaScript and CSS files to our HTML file and call the methods specified in the plugin in JQuery way.

There are lot of handpicked plugins available in JQuery website and which allows the user to start using them instead of re-inventing the wheel.

Examples of JQuery plugins are as follows,

JQuery plugins for Form validation, animations, gallery, light box, Menus and Nav, Social Media plugins, Accordions, layouts, sliders and zoom effects.

Cross Browser Compatibility

In the process of UI Development each browser behaves differently for each UI Components. We needs to do some extra work in order to make any website work perfectly in all the browsers.

The jQuery team knows all about cross-browser issues, and they have written this knowledge into the jQuery library. JQuery will run exactly the same in all major browsers.

Installation of JQuery

We do not require any separate installation for JQuery. It needs to add a jquery.js to our application to starts working with it.

We can use the JQuery in two ways such as online version and offline version.

In Offline version we have to download the JQuery.js from JQuery official website and add it to our webpage. Or in Online version we can link JQuery to any CDN like Google.

In this course we will use an Offline version, here we have to download the JQuery.js from official website jquery.com.

There are two versions of JQuery such as Production version and Development version.

Production version

This is a compressed version of JQuery and not human readable.

This version is comparatively less in size than development version. We normally use this for live website or in production environments.

Development Version

This is an un-compressed version and it is human readable.

This version is comparatively more in size than production version. We normally use this for development environments and Testing environments.

We can easily identify the production version and development version with the name of the JavaScript file such as,

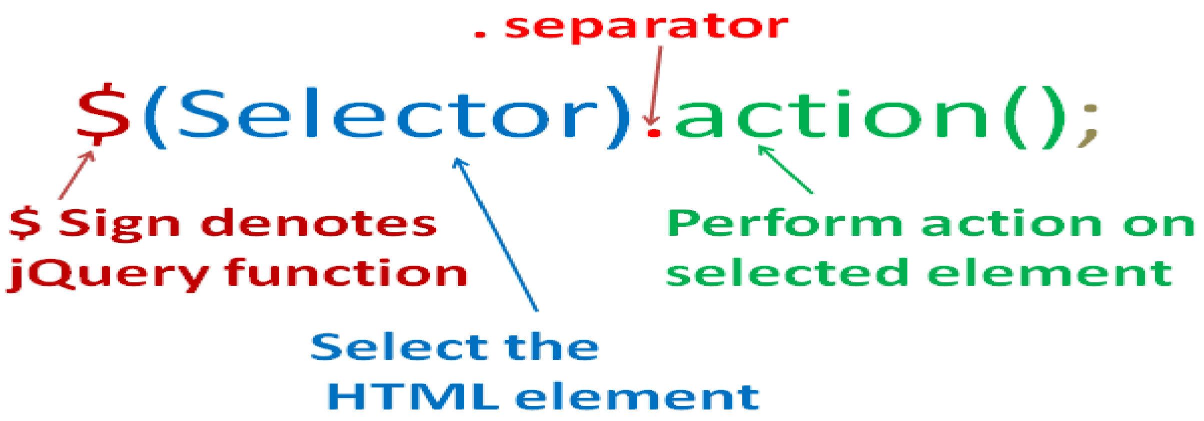
```
Development Version
-------------------
jquery-3.3.1.js

Production Version
------------------
jquery-3.3.1.min.js
```

JQuery Syntax

JQuery is used to select the HTML elements using various selectors and perform some actions on it.

We can identify the JQuery expression using the **$** symbol.

The syntax of JQuery is as follows,


. separator
$(Selector).action();
$ Sign denotes jQuery function
Perform action on selected element
Select the HTML element

The above diagram explains the complete syntax of JQuery.

Let’s see some of the examples of JQuery selectors and their actions as follows,

```
$(this).hide(); // Hides the current element
$('p').hide(); // Hides all the 'p' elements
$('#myCard').toggle('slow'); // Toggles the element
$('.container').hide(); // Hides all the elements with the class 'container'
$('#text').hide(); // Hides the element with id 'text'
```

JQuery & AJAX

JQuery Selectors

For JQuery Selectors we can use the same selectors used for CSS.

The type of CSS and JQuery selectors are as follows,

```
Element Selector => $(‘p’)
ID selector => $(‘#main-nav’)
Class selector => $(‘.container’)
Attribute Selector => $(“input[type=‘text’]”)
Pseudo classes selector => $(‘tr:even’)
```

JQuery Events

Any kind of actions that user can perform on the webpage is called an event. The moment a user is performing an action on the browser is called as an event.

Example:

Selecting a radio button

Focus on an input field

Hover on a button, clicking a button is an event.

Mouse Events	Keyboard Events	Form Events	Document Events
click	keypress	Submit	Load
dbclick	keydown	Change	Resize
mouseenter	keyup	Focus	Scroll
mouseleave			unload

JQuery Basics

We can select the specific HTML element using the JQuery selector. We can apply various actions on that selected HTML Element.

Ex:

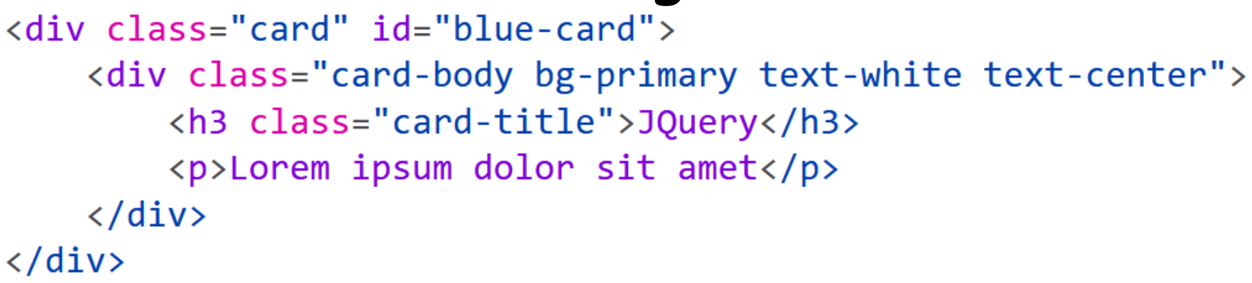
```
<p>Lorem ipsum dolor sit amet</p>
```

To hide the above paragraph while performing the click event is as follows,

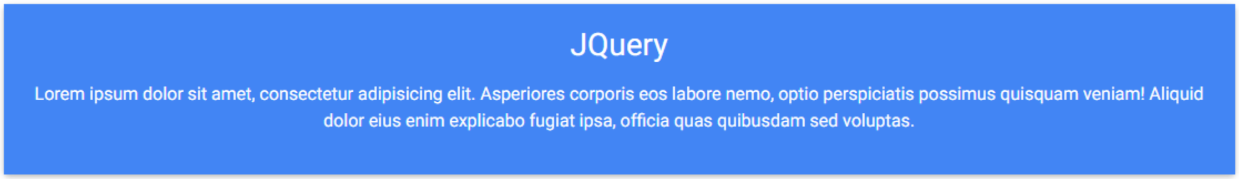
```
<p onclick="$(this).hide()">Lorem ipsum dolor sit amet</p>
```

We can even select the specific element from outside of the HTML and we can apply the actions on them.

JQuery & AJAX



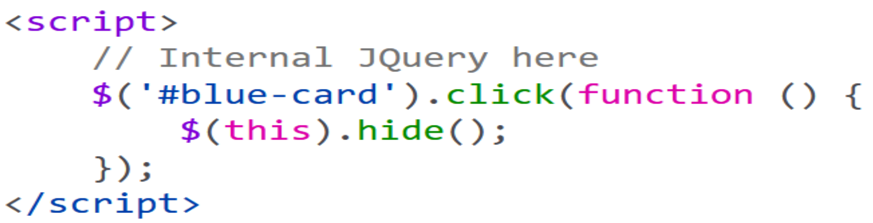
```html
<div class="card" id="blue-card">
    <div class="card-body bg-primary text-white text-center">
        <h3 class="card-title">JQuery</h3>
        <p>Lorem ipsum dolor sit amet</p>
    </div>
</div>
```



JQuery

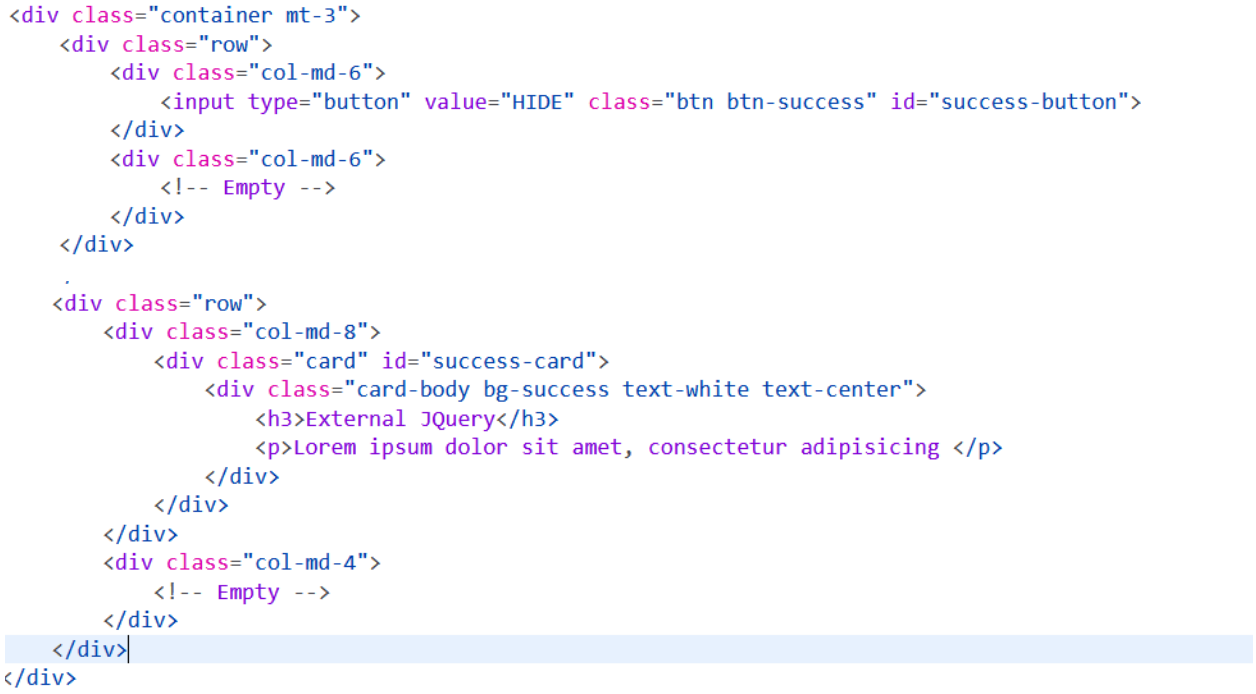
Lorem ipsum dolor sit amet, consectetur adipisicing elit. Asperiores corporis eos labore nemo, optio perspiciatis possimus quisquam veniam! Aliquid dolor eius enim explicabo fugiat ipsa, officia quas quibusdam sed voluptas.

We can hide the above card by clicking on it using JQuery is as follows,

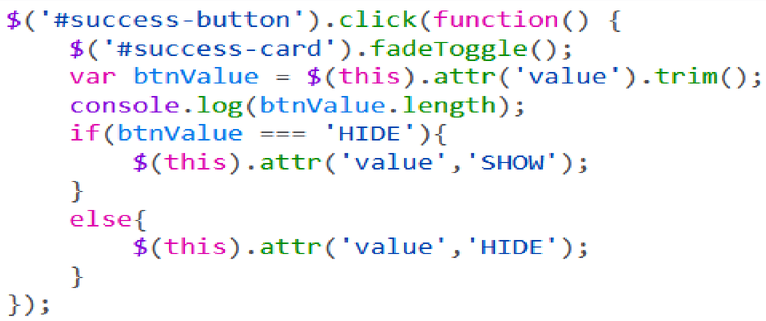


```html
<script>
    // Internal JQuery here
    $('#blue-card').click(function () {
        $(this).hide();
    });
</script>
```

We can even apply the toggle effect for the selected HTML Element.



```html
<div class="container mt-3">
    <div class="row">
        <div class="col-md-6">
            <input type="button" value="HIDE" class="btn btn-success" id="success-button">
        </div>
        <div class="col-md-6">
            <!-- Empty -->
        </div>
    </div>

    <div class="row">
        <div class="col-md-8">
            <div class="card" id="success-card">
                <div class="card-body bg-success text-white text-center">
                    <h3>External JQuery</h3>
                    <p>Lorem ipsum dolor sit amet, consectetur adipisicing </p>
                </div>
            </div>
        </div>
        <div class="col-md-4">
            <!-- Empty -->
        </div>
    </div>
</div>
```

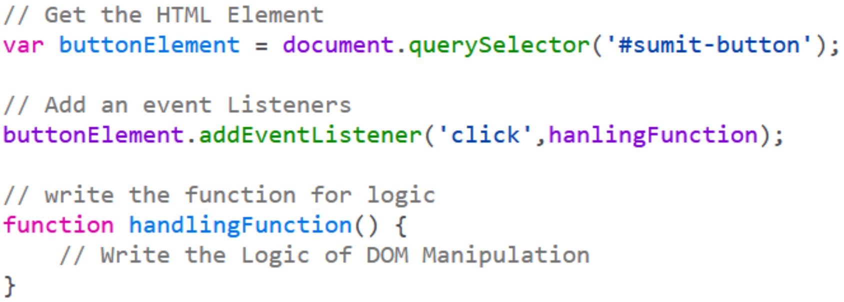


```
$('#success-button').click(function() {
    $('#success-card').fadeToggle();
    var btnValue = $(this).attr('value').trim();
    console.log(btnValue.length);
    if(btnValue === 'HIDE'){
        $(this).attr('value','SHOW');
    }
    else{
        $(this).attr('value','HIDE');
    }
});
```

JQuery Event Listeners

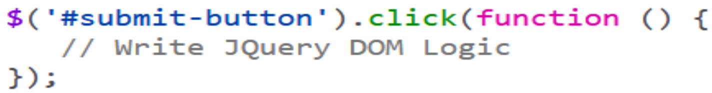
The JavaScript events handling is easier in the form of JQuery. Using JavaScript we have to get the element and add an event Listeners and attach a function for them.

JavaScript Way



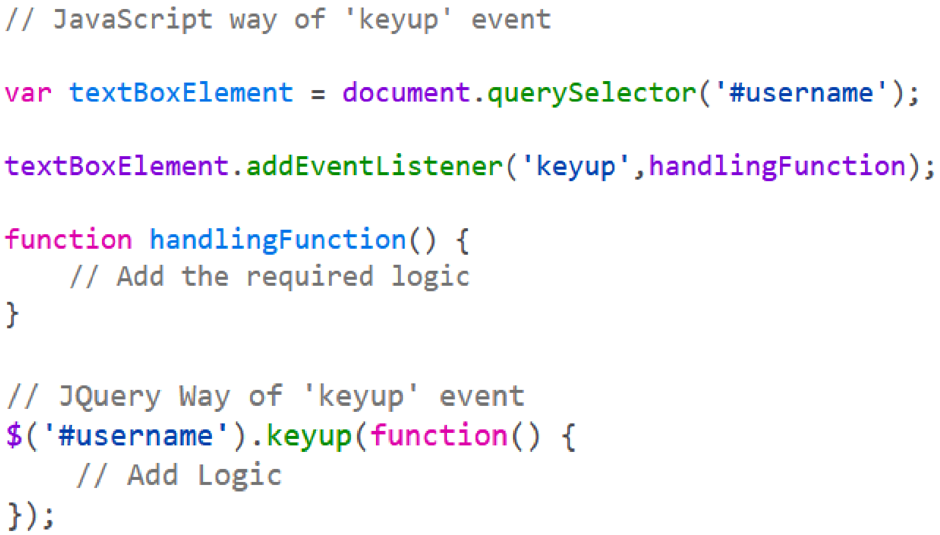
```
// Get the HTML Element
var buttonElement = document.querySelector('#sumit-button');

// Add an event Listeners
buttonElement.addEventListener('click',hanlingFunction);

// write the function for logic
function handlingFunction() {
    // Write the Logic of DOM Manipulation
}
```

JQuery Way



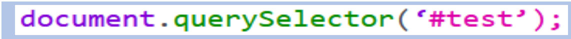
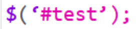
```
$('#submit-button').click(function () {
    // Write JQuery DOM Logic
});
```

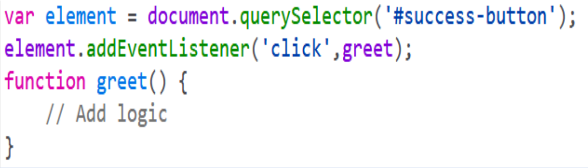
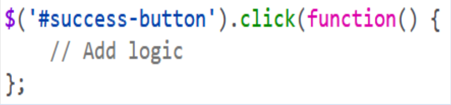
To apply the ‘keyup’ event for a text box using JavaScript is as follows,

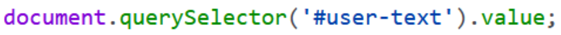
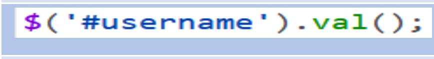
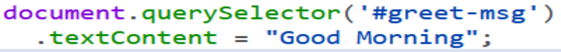
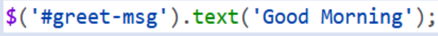
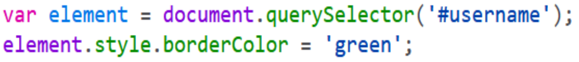
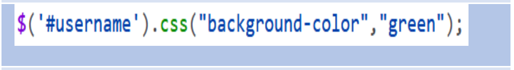


```
// JavaScript way of 'keyup' event

var textBoxElement = document.querySelector('#username');

textBoxElement.addEventListener('keyup',handlingFunction);

function handlingFunction() {
    // Add the required logic
}

// JQuery Way of 'keyup' event
$('#username').keyup(function() {
    // Add Logic
});
```

JQuery & AJAX

Description	JavaScript Way	JQuery Way
Select a HTML Element	 `document.querySelector(‘#test’);`	 `$(‘#test’);`

Add an event Listener	 `var element = document.querySelector('#success-button');`
	`element.addEventListener('click',greet);`
	`function greet() {`
	`    // Add logic`
	`}`
		 `$('#success-button').click(function() {`
		`    // Add logic`
		`};`

Get input field value	 `document.querySelector('#user-text').value;`	 `$('#username').val();`
Set text from JS file	 `document.querySelector('#greet-msg')` `   .textContent = "Good Morning";`	 `$('#greet-msg').text('Good Morning');`
Change CSS Styles	 `var element = document.querySelector('#username');` `element.style.borderColor = 'green';`	 `$('#username').css("background-color","green");`

AJAX Introduction

Traditionally webpages required reloading to update their content.

For web based emails if users have to reload the page each time to check the new mails is the biggest drawback, as it is slow and requires user input.

For each reload server has to send the completed HTML, CSS, JavaScript code and also the data to the browser.

This approach is highly inefficient and ideally the server should only send the new messages only to the browser instead of the complete page.

By 2003 all the major browsers solved this issue by adopting a new concept called XMLHttpRequest Object.

XMLHttpRequest is a part of a Technology called AJAX.

Using AJAX the data could be passed between the browser and server using XMLHttpRequest API without complete reload of the entire webpage.

Example:

Google Maps uses AJAX to get new map tiles without reload of a webpage.

Gmail gets the new mails without reload the entire web page.

Crickinfo website displays the data without reloading the actual page.

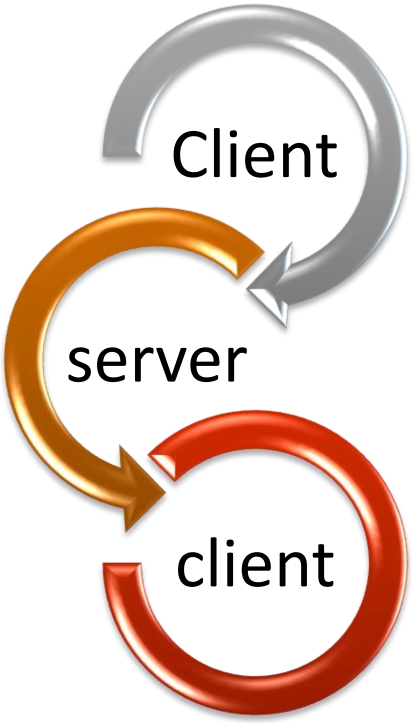
Unfortunately each browser implements the AJAX API differently. But JQuery supports AJAX which works in all the browsers uniformly.

- AJAX Stands for Asynchronous JavaScript and Xml
- AJAX is just a client side programming technique to updates the different parts of the webpage without reloading whole page.

- For This it uses the browsers built-in object called XMLHttpRequest Object.
- This object is responsible to request the data from the webserver for some specific intervals.
- Once we fetch the data from the webserver, we can display the data on the HTML using JavaScript DOM Manipulation or JQuery.
- Using AJAX we can asynchronously get the data from the server. We may get the data in ‘xml’ format.

Usually the same AJAX technique is used to transfer the data along with the ‘xml’ , we can even transfer the data in JSON and ‘plain Text’ format.

How AJAX Works



1) An event occurs in the client or the page loads We create an XMLHttpRequest and send to server

2) Server process the Request and prepares the response and send back to the browser

3) Client process the returned data from the server using JavaScript and updates the page contents using HTML DOM Manipulation.

XMLHttpRequest Object

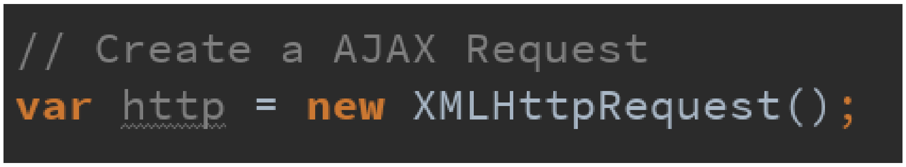
In the whole process of AJAX, this Object places the key role.

JQuery & AJAX

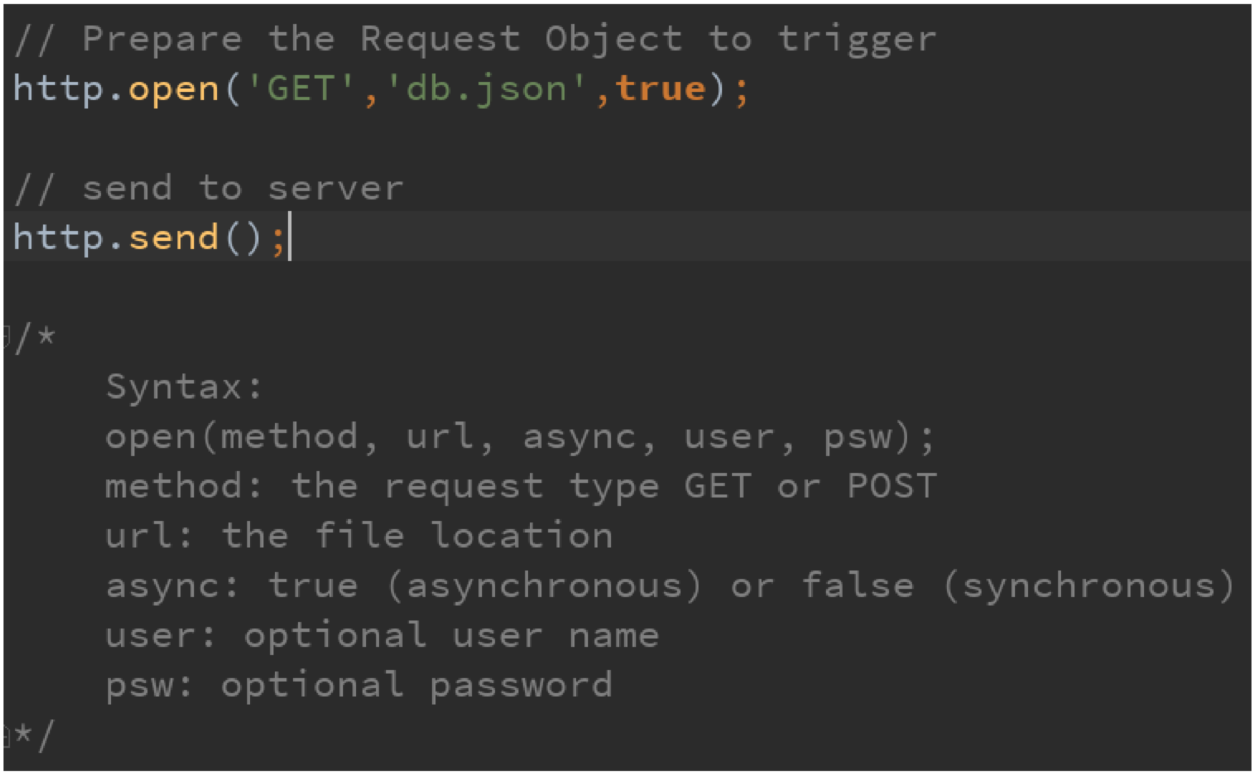
This object is responsible to exchange data with a webserver behind the scenes. It means that we can updates the parts of the webpage without reloading the entire webpage.

All the modern browsers have a built in XMLHttpRequest Object.

To create this object we use the below syntax,



```
// Create a AJAX Request
var http = new XMLHttpRequest();
```



```
// Prepare the Request Object to trigger
http.open('GET','db.json',true);

// send to server
http.send();

/*
    Syntax:
    open(method, url, async, user, psw);
    method: the request type GET or POST
    url: the file location
    async: true (asynchronous) or false (synchronous)
    user: optional user name
    psw: optional password
*/
```

The onreadystatechange Property

The readyState property holds the status of the XMLHttpRequest.
The onreadystatechange property defines a function to be executed when the readyState changes.
The status property and the statusText property holds the status of the XMLHttpRequest object.

Property	Description
Onreadystatechange	Defines a function to be called when the readyState property changes
readyState	Holds the status of the XMLHttpRequest. 0: request not initialized 1: server connection established 2: request received

	3: processing request 4: request finished and response is ready
Status	200: "OK" 403: "Forbidden" 404: "Page not found"
statusText	Returns the status-text (e.g. "OK" or "Not Found")

Note: The onreadystatechange function is called every time the readyState changes.
When readyState is 4 and status is 200, the response is ready.