

## **Study Unit 13: INTRODUCTION OF JAVASCRIPT**

### **Outline**

Student will be able to learn about Javascript basics and its libraries and most commonly and widely jQuery and Ajax technology.

- Introduction to JavaScript
- JavaScript libraries
- jQuery library and Ajax

### **Learning Outcomes of Study Unit 13**

Upon completion of this study unit, Student will be introduced to Javascript and its different libraries. The most popular jQuery library as well

#### **1.1: JavaScript**

- Why to Learn Javascript
- Applications of Javascript Programming
- Multidimensional Array

#### **1.2: jQuery Library**

- What is jQuery?
- How to use URL/CDN
- How to call jQuery functions?
- How to use Custom scripts?

#### **1.3: Ajax (Asynchronous JavaScript and XML)**

- Why to learn Ajax?
- Ajax Technologies
- Getting and passing data to server

# 1.1: JASCRPT

**JavaScript** is a lightweight, interpreted **programming** language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

## Why to Learn Javascript

**Javascript** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning Javascript:

- Javascript is the most popular **programming language** in the world and that makes it a programmer's great choice. Once you learnt Javascript, it helps you developing great front-end as well as back-end softwares using different Javascript based frameworks like jQuery, Node.JS etc.
- Javascript is everywhere, it comes installed on every modern web browser and so to learn Javascript you really do not need any special environment setup. For example Chrome, Mozilla Firefox , Safari and every browser you know as of today, supports Javascript.
- Javascript helps you create really beautiful and crazy fast websites. You can develop your website with a console like look and feel and give your users the best Graphical User Experience.
- JavaScript usage has now extended to mobile app development, desktop app development, and game development. This opens many opportunities for you as Javascript Programmer.
- Due to high demand, there is tons of job growth and high pay for those who know JavaScript. You can navigate over to different job sites to see what having JavaScript skills looks like in the job market.
- Great thing about Javascript is that you will find tons of frameworks and Libraries already developed which can be used directly in your software development to reduce your time to market.

There could be 1000s of good reasons to learn Javascript Programming. But one thing for sure, to learn any **programming language**, not only Javascript, you just need to code, and code and finally code until you become expert.

## Hello World using Javascript

Just to give you a little excitement about **Javascript programming**, I'm going to give you a small conventional Javascript Hello World program, You can try it using Demo link

```
<html>
  <body>
    <script language = "javascript" type = "text/javascript">
      <!--
        document.write("Hello World!")
      //-->
    </script>
  </body>
</html>
```

There are many useful **Javascript frameworks** and libraries available:

- Angular
- React
- jQuery
- Vue.js
- Ext.js
- Ember.js
- Meteor
- Mithril
- Node.js
- Polymer

- Aurelia
- Backbone.js

It is really impossible to give a complete list of all the available Javascript frameworks and libraries. The Javascript world is just too large and too much new is happening.

## Applications of Javascript Programming

As mentioned before, **Javascript** is one of the most widely used **programming languages** (Front-end as well as Back-end). It has its presence in almost every area of software development. I'm going to list few of them here:

- **Client side validation** - This is really important to verify any user input before submitting it to the server and Javascript plays an important role in validating those inputs at front-end itself.
- **Manipulating HTML Pages** - Javascript helps in manipulating HTML page on the fly. This helps in adding and deleting any HTML tag very easily using javascript and modify your HTML to change its look and feel based on different devices and requirements.
- **User Notifications** - You can use Javascript to raise dynamic pop-ups on the webpages to give different types of notifications to your website visitors.
- **Back-end Data Loading** - Javascript provides Ajax library which helps in loading back-end data while you are doing some other processing. This really gives an amazing experience to your website visitors.
- **Presentations** - JavaScript also provides the facility of creating presentations which gives website look and feel. JavaScript provides RevealJS and BespokeJS libraries to build a web-based slide presentations.
- **Server Applications** - Node JS is built on Chrome's Javascript runtime for building fast and scalable network applications. This is an event based library which helps in developing very sophisticated server applications including Web Servers.

This list goes on, there are various areas where millions of software developers are happily using Javascript to develop great websites and others softwares

## 1.2: What is jQuery?

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: **Write less, do more.** jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery –

- **DOM manipulation** – The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
- **Event handling** – The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- **AJAX Support** – The jQuery helps you a lot to develop a responsive and featurerich site using AJAX technology.
- **Animations** – The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- **Lightweight** – The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).
- **Cross Browser Support** – The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
- **Latest Technology** – The jQuery supports CSS3 selectors and basic XPath syntax.

There are two ways to use jQuery.

- **Local Installation** – You can download jQuery library on your local machine and include it in your HTML code.
- **CDN Based Version** – You can include jQuery library into your HTML code directly from Content Delivery Network (CDN).
- Go to the <https://jquery.com/download/> to download the latest version available.

- Now put downloaded **jquery-2.1.3.min.js** file in a directory of your website, e.g. /jquery.

## Example

Now you can include *jquery* library in your HTML file as follows –

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript" src = "/jquery/jquery-2.1.3.min.js">
    </script>

    <script type = "text/javascript">
      $(document).ready(function() {
        document.write("Hello, World!");
      });
    </script>
  </head>

  <body>
    <h1>Hello</h1>
  </body>
</html>
```

This will produce following result –

You can include jQuery library into your HTML code directly from Content Delivery Network (CDN). Google and Microsoft provides content deliver for the latest version.

We are using Google CDN version of the library throughout this tutorial.

## Example

Now let us rewrite above example using jQuery library from Google CDN.

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript"
      src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>

    <script type = "text/javascript">
      $(document).ready(function() {
        document.write("Hello, World!");
      });
    </script>
  </head>

  <body>
    <h1>Hello</h1>
  </body>
</html>
```

This will produce following result –

As almost everything, we do when using jQuery reads or manipulates the document object model (DOM), we need to make sure that we start adding events etc. as soon as the DOM is ready.

If you want an event to work on your page, you should call it inside the `$(document).ready()` function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

To do this, we register a ready event for the document as follows –

```
$(document).ready(function() {  
    // do stuff when DOM is ready  
});
```

To call upon any jQuery library function, use HTML script tags as shown below –

```
<html>  
  <head>  
    <title>The jQuery Example</title>  
    <script type = "text/javascript"  
      src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">  
    </script>  
  
    <script type = "text/javascript" language = "javascript">  
      $(document).ready(function() {  
        $("div").click(function() {alert("Hello, world!");});  
      });  
    </script>  
  </head>  
  
  <body>  
    <div id = "mydiv">  
      Click on this to see a dialogue box.  
    </div>  
  </body>  
</html>
```

This will produce following result –

It is better to write our custom code in the custom JavaScript file : **custom.js**, as follows –



```
/* Filename: custom.js */
$(document).ready(function() {

    $("div").click(function() {
        alert("Hello, world!");
    });
});
```

Now we can include **custom.js** file in our HTML file as follows –

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript"
      src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>

    <script type = "text/javascript" src = "/jquery/custom.js">
    </script>
  </head>

  <body>
    <div id = "mydiv">
      Click on this to see a dialogue box.
    </div>
  </body>
</html>
```

This will produce following result –

Click on this to see a dialogue box.

## 1.3: Why to Learn Ajax?

AJAX stands for **A**synchronous **J**avaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.

- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

## Rich Internet Application Technology

AJAX is the most viable Rich Internet Application (RIA) technology so far. It is getting tremendous industry momentum and several tool kit and frameworks are emerging. But at the same time, AJAX has browser incompatibility and it is supported by JavaScript, which is hard to maintain and debug.

## AJAX is Based on Open Standards

AJAX is based on the following open standards –

- Browser-based presentation using HTML and Cascading Style Sheets (CSS).

- Data is stored in XML format and fetched from the server.
- Behind-the-scenes data fetches using XMLHttpRequest objects in the browser.
- JavaScript to make everything happen.

## AJAX Technologies

AJAX cannot work independently. It is used in combination with other technologies to create interactive webpages.

### JavaScript

- Loosely typed scripting language.
- JavaScript function is called when an event occurs in a page.
- Glue for the whole AJAX operation.

### DOM

- API for accessing and manipulating structured documents.
- Represents the structure of XML and HTML documents.

### CSS

- Allows for a clear separation of the presentation style from the content and may be changed programmatically by JavaScript

### XMLHttpRequest

- JavaScript object that performs asynchronous interaction with the server.
-

## Here is a list of some famous web applications that make use of AJAX.

A user can drag an entire map by using the mouse, rather than clicking on a button.

- <https://maps.google.com/>

As you type, Google offers suggestions. Use the arrow keys to navigate the results.

- <https://www.google.com/webhp?complete=1&hl=en>

Gmail is a webmail built on the idea that emails can be more intuitive, efficient, and useful.

- <https://gmail.com/>

AJAX is an acronym standing for Asynchronous JavaScript and XML and this technology helps us to load data from the server without a browser page refresh.

If you are new with AJAX, I would recommend you go through our [Ajax Tutorial](#) before proceeding further.

JQuery is a great tool which provides a rich set of AJAX methods to develop next generation web application.

This is very easy to load any static or dynamic data using JQuery AJAX. JQuery provides **load()** method to do the job –

## Syntax

Here is the simple syntax for **load()** method –

```
[selector].load( URL, [data], [callback] );
```

Here is the description of all the parameters –

- **URL** – The URL of the server-side resource to which the request is sent. It could be a CGI, ASP, JSP, or PHP script which generates data dynamically or out of a database.
- **data** – This optional parameter represents an object whose properties are serialized into properly encoded parameters to be passed to the request. If specified, the request is made using the **POST** method. If omitted, the **GET** method is used.
- **callback** – A callback function invoked after the response data has been loaded into the elements of the matched set. The first parameter passed to this function is the response text received from the server and second parameter is the status code.

## Example

Consider the following HTML file with a small JQuery coding –

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript"
      src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>

    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("#driver").click(function(event) {
          $('#stage').load('/jquery/result.html');
        });
      });
    </script>
  </head>
</html>
```

```
    });  
</script>  
</head>  
  
<body>  
  <p>Click on the button to load /jquery/result.html file -</p>  
  
  <div id = "stage" style = "background-color:cc0;">  
    STAGE  
  </div>  
  
  <input type = "button" id = "driver" value = "Load Data" />  
</body>  
</html>
```

Here **load()** initiates an Ajax request to the specified URL **/jquery/result.html** file. After loading this file, all the content would be populated inside `<div>` tagged with ID `stage`. Assuming, our `/jquery/result.html` file has just one HTML line –

```
<h1>THIS IS RESULT...</h1>
```

When you click the given button, then `result.html` file gets loaded.

There would be a situation when server would return JSON string against your request. JQuery utility function **getJSON()** parses the returned JSON string and makes the resulting string available to the callback function as first parameter to take further action.

## Syntax

Here is the simple syntax for **getJSON()** method –

```
[selector].getJSON( URL, [data], [callback] );
```

Here is the description of all the parameters –

- **URL** – The URL of the server-side resource contacted via the GET method.

- **data** – An object whose properties serve as the name/value pairs used to construct a query string to be appended to the URL, or a preformatted and encoded query string.
- **callback** – A function invoked when the request completes. The data value resulting from digesting the response body as a JSON string is passed as the first parameter to this callback, and the status as the second.

## Example

Consider the following HTML file with a small JQuery coding –

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript"
      src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>

    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("#driver").click(function(event) {

          $.getJSON('/jquery/result.json', function(jd) {
            $('#stage').html('<p> Name: ' + jd.name + '</p>');
            $('#stage').append('<p>Age : ' + jd.age+ '</p>');
            $('#stage').append('<p> Sex: ' + jd.sex+ '</p>');
          });

        });
      });
    </script>
  </head>

  <body>
```

```
<p>Click on the button to load result.json file -</p>

<div id = "stage" style = "background-color:#eee;">
  STAGE
</div>

<input type = "button" id = "driver" value = "Load Data" />
</body>
</html>
```

Here JQuery utility method **getJSON()** initiates an Ajax request to the specified URL **result.json** file. After loading this file, all the content would be passed to the callback function which finally would be populated inside `<div>` tagged with ID *stage*. Assuming, our `result.json` file has following json formatted content –

```
{
  "name": "Zara Ali",
  "age"  : "67",
  "sex": "female"
}
```

When you click the given button, then `result.json` file gets loaded.

Many times you collect input from the user and you pass that input to the server for further processing. JQuery AJAX made it easy enough to pass collected data to the server using **data** parameter of any available Ajax method.

## Example

This example demonstrate how can pass user input to a web server script which would send the same result back and we would print it –

```
<html>
```



```
<head>
  <title>The jQuery Example</title>
  <script type = "text/javascript"
    src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
  </script>

  <script type = "text/javascript" language = "javascript">
    $(document).ready(function() {
      $("#driver").click(function(event) {
        var name = $("#name").val();
        $("#stage").load('/jquery/result.php', {"name":name} );
      });
    });
  </script>
</head>

<body>
  <p>Enter your name and click on the button:</p>
  <input type = "input" id = "name" size = "40" /><br />

  <div id = "stage" style = "background-color:cc0;">
    STAGE
  </div>

  <input type = "button" id = "driver" value = "Show Result" />
</body>
</html>
```

Here is the code written in **result.php** script –

```
<?php
if( $_REQUEST["name"] ){
    $name = $_REQUEST['name'];
```

```
    echo "Welcome ". $name;  
  }  
?>
```

Now you can enter any text in the given input box and then click "Show Result" button to see what you have entered in the input box.

You have seen basic concept of AJAX using JQuery. Following table lists down all important JQuery AJAX methods which you can use based your programming need –

Sr.No.	Methods & Description
1	<code>jQuery.ajax( options )</code>  Load a remote page using an HTTP request.
2	<code>jQuery.ajaxSetup( options )</code>  Setup global settings for AJAX requests.
3	<code>jQuery.get( url, [data], [callback], [type] )</code>  Load a remote page using an HTTP GET request.
4	<code>jQuerygetJSON( url, [data], [callback] )</code>  Load JSON data using an HTTP GET request.
5	<code>jQuery.getScript( url, [callback] )</code>

	Loads and executes a JavaScript file using an HTTP GET request.
6	jQuery.post( url, [data], [callback], [type] ) Load a remote page using an HTTP POST request.
7	load( url, [data], [callback] ) Load HTML from a remote file and inject it into the DOM.
8	serialize( ) Serializes a set of input elements into a string of data.
9	serializeArray( ) Serializes all forms and form elements like the .serialize() method but returns a JSON data structure for you to work with.

You can call various JQuery methods during the life cycle of AJAX call progress. Based on different events/stages following methods are available –

<b>Sr.No.</b>	<b>Methods &amp; Description</b>
1	ajaxComplete( callback )

	Attach a function to be executed whenever an AJAX request completes.
2	<code>ajaxStart( callback )</code> Attach a function to be executed whenever an AJAX request begins and there is none already active.
3	<code>ajaxError( callback )</code> Attach a function to be executed whenever an AJAX request fails.
4	<code>ajaxSend( callback )</code> Attach a function to be executed before an AJAX request is sent.
5	<u><code>ajaxStop( callback )</code></u> Attach a function to be executed whenever all AJAX requests have ended.
6	<u><code>ajaxSuccess( callback )</code></u> Attach a function to be executed whenever an AJAX request completes successfully.

## Self-Review Questions (SRQ) For Study Session 13

---

Now that you have completed this study unit, you can assess how well you have achieved its Learning Outcomes by answering these questions. Write your answers in your Study Diary and discuss them with your Tutor at the next Study Support Meeting or Online interactive sessions. You can also check your answers at the Self-Review Answers section which is located at the end of this Module.

1. What is JavaScript where we can use JavaScript and write the names of JavaScript frameworks?	4. What is Ajax and its technologies? Describe the syntax of Ajax?
2. What is jQuery which features are supported by jQuery?	
3. How we can call jQuery functions?	

## Self-Review Answers (SRA) for Study Unit 13

### 1. What is JavaScript where we can use JavaScript and write the names of JavaScript frameworks?

**JavaScript** is a lightweight, interpreted **programming** language. It is designed for creating network-centric applications. It is complimentary to and integrated with Java. **JavaScript** is very easy to implement because it is integrated with HTML. It is open and cross-platform.

➤ There are many useful **Javascript frameworks** and libraries available:

- Angular
- React
- jQuery
- Vue.js
- Ext.js
- Ember.js
- Meteor
- Mithril
- Node.js
- Polymer
- Aurelia
- Backbone.js

## 2. What is jQuery which features are supported by jQuery?

jQuery is a fast and concise JavaScript Library created by John Resig in 2006 with a nice motto: **Write less, do more.** jQuery simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development. jQuery is a JavaScript toolkit designed to simplify various tasks by writing less code. Here is the list of important core features supported by jQuery –

- **DOM manipulation** – The jQuery made it easy to select DOM elements, negotiate them and modifying their content by using cross-browser open source selector engine called **Sizzle**.
- **Event handling** – The jQuery offers an elegant way to capture a wide variety of events, such as a user clicking on a link, without the need to clutter the HTML code itself with event handlers.
- **AJAX Support** – The jQuery helps you a lot to develop a responsive and featurerich site using AJAX technology.
- **Animations** – The jQuery comes with plenty of built-in animation effects which you can use in your websites.
- **Lightweight** – The jQuery is very lightweight library - about 19KB in size (Minified and gzipped).
- **Cross Browser Support** – The jQuery has cross-browser support, and works well in IE 6.0+, FF 2.0+, Safari 3.0+, Chrome and Opera 9.0+
- **Latest Technology** – The jQuery supports CSS3 selectors and basic XPath syntax

## 3. How we can call jQuery functions?

As almost everything, we do when using jQuery reads or manipulates the document object model (DOM), we need to make sure that we start adding events etc. as soon as the DOM is ready.

If you want an event to work on your page, you should call it inside the `$(document).ready()` function. Everything inside it will load as soon as the DOM is loaded and before the page contents are loaded.

To do this, we register a ready event for the document as follows –

```
$(document).ready(function() {  
    // do stuff when DOM is ready
```

```
});
```

To call upon any jQuery library function, use HTML script tags as shown below –

```
<html>
  <head>
    <title>The jQuery Example</title>
    <script type = "text/javascript"
      src = "https://ajax.googleapis.com/ajax/libs/jquery/2.1.3/jquery.min.js">
    </script>

    <script type = "text/javascript" language = "javascript">
      $(document).ready(function() {
        $("div").click(function() {alert("Hello, world!");});
      });
    </script>
  </head>

  <body>
    <div id = "mydiv">
      Click on this to see a dialogue box.
    </div>
  </body>
</html>
```

#### 4. What is Ajax and its technologies? Describe the syntax of Ajax?

AJAX stands for **A**synchronous **J**avaScript and **X**ML. AJAX is a new technique for creating better, faster, and more interactive web applications with the help of XML, HTML, CSS, and Java Script.



- Ajax uses XHTML for content, CSS for presentation, along with Document Object Model and JavaScript for dynamic content display.
- Conventional web applications transmit information to and from the sever using synchronous requests. It means you fill out a form, hit submit, and get directed to a new page with new information from the server.
- With AJAX, when you hit submit, JavaScript will make a request to the server, interpret the results, and update the current screen. In the purest sense, the user would never know that anything was even transmitted to the server.
- XML is commonly used as the format for receiving server data, although any format, including plain text, can be used.
- AJAX is a web browser technology independent of web server software.
- A user can continue to use the application while the client program requests information from the server in the background.
- Intuitive and natural user interaction. Clicking is not required, mouse movement is a sufficient event trigger.
- Data-driven as opposed to page-driven.

### Syntax

```
$.ajax({name:value, name:value, ... })
```

```
<!DOCTYPE html>
<html>
<head>
<script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js">
</script>
<script>
$(document).ready(function(){
  $("button").click(function(){
    $.ajax({url: "demo_test.txt", success: function(result){
      $("#div1").html(result);
    }});
  });
});
</script>
</head>
<body>

<div id="div1"><h2>Let jQuery AJAX Change This Text</h2></div>

<button>Get External Content</button>

</body>
</html>
```

## jQuery and AJAX is FUN!

This is some text in a paragraph.

Get External Content

## References and Additional Reading Materials

[https://www.w3schools.com/jquery/ajax\\_ajax.asp](https://www.w3schools.com/jquery/ajax_ajax.asp)

<https://www.w3schools.com/js/default.asp>

<https://www.w3schools.com/jquery/>