

Introduction

jQuery





WSA

Forward looking IT finishing school

Introduction to jQuery

(Write less, Do more)

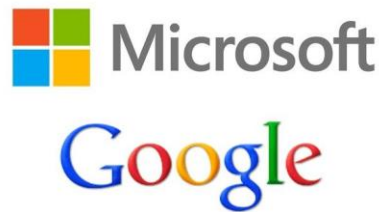
What is a jQuery?

- jQuery (<https://jquery.com/>) is a **lightweight, feature rich JavaScript library**
- jQuery provides easy-to-use APIs to achieve the following:
 - Manipulate HTML, CSS and DOM
 - Handling HTML events
 - Applying effects and animations
 - AJAX
 - Utilities
- This is one of the most popularly used **OpenSource** libraries
- Developers can do more with lesser number of lines of code, compared with vanilla JavaScript



jQuery is very popular

- Most popular and extendable library for websites and mobile single page app development
- As of June 2018, jQuery is used on **73% of the top 1 million websites**, and by **22.4% of overall number of websites**
- Most of the big companies (ex: Google) use jQuery library and host jQuery CDN (Both Google and Microsoft)
- jQuery simplifies the syntax for finding, selecting, and manipulating DOM elements
- Supported by all modern browsers



History of jQuery



- Originally created in Jan 2006 at a BarCamp by **John Resig** (who became very popular for his work in jQuery)
- Official major releases were made in the following time-lines:
 - **1.0 - Aug 2006**
 - **2.0 - Jan 2014**
 - **3.0 - Jun 2016**
 - **3.3.1 - Jan 2018 (which we will be using!)**
- The jQuery team also developed **jQuery mobile** (for mobile / touch devices) and **jQuery UI** (for developing GUI) which has also become popular

jQuery CDN

- Both Google and Microsoft host jQuery, links are provided below

Google CDN :

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">  
</script>
```

Microsoft CDN :

```
<script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js">  
</script>
```

- For our learning, you can either use CDN or include a local copy of jQuery library
- Local downloaded copy is available at [Google Drive](#)

jQuery Development Principles

- **P1: Separation of HTML and JS**

- jQuery library provides simple syntax for handling DOMs (Ex: Events)
- Hence, developers can avoid adding HTML event attributes to call JavaScript functions
- By this way JavaScript code can be completely separated code from HTML

- **P2: Elimination of cross-browser incompatibilities**

- The JavaScript engines of different browsers differ from each other
- Therefore, 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

- **P3: Extensibility**

- New events, elements, and methods can be easily added and then reused as a plugin



WSA

Forward looking IT finishing school

jQuery syntax

(Getting hands-on)

jQuery syntax

- The jQuery syntax is tailored to select HTML elements and performing some action on them

Syntax : `$(selector).action()`

Selector : `this | id | class | element`

`$(this).action();` *// Take action on current element*

`$("div").action();` *// Take action on all divisions*

`$(".abc").action();` *// Take action on all abc class*

- Actions can be of varied in nature (ex: DOM manipulation, events etc..) which we will learn in subsequent chapters

jQuery syntax

- The \$ used is actually a jQuery function

```
<script src="../../jquery/jquery-3.3.1.min.js"
type="text/javascript"></script>
```

```
<script>
```

```
/* If you remove jQuery inclusion above, you will get undefined here */
document.write("Type of $ = " + typeof $);
```

```
if($ == jQuery) {
    document.write("<br>$ Function is referencing to jQuery ");
}
</script>
```

jQuery ready event

- Before accessing any element with jQuery, the DOM tree should be completely created
- There could be following possible failures if document is not fully loaded
 - Finding an element that is not created
 - Hiding an element that is not created
- jQuery detects the state of readiness of document then perform actions

```
<script>
$(document).ready(function() {
    document.write("Welcome to jQuery...");
    document.write("DOM is created...");
});
</script>
```

jQuery load event

- Sometimes you may want the entire window to be loaded
- Hence use the "load" method

```
<script>
$(window).on("load", function() {
    document.write("Window loaded!!")
});
</script>
```

jQuery passing named functions

- U can also pass named functions to jQuery as follows

```
<script>
$(document).ready( myFunc );

// Calling your named function
function myFunc() {
    console.log("DOM loaded");
}
</script>
```

jQuery no-conflict mode

- By default, jQuery uses \$ as a shortcut for jQuery
- Thus, if you are using another JavaScript library that uses the \$ variable, you can run into conflicts with jQuery
- In order to avoid these conflicts, you need to put jQuery in **no-conflict mode** immediately after it is loaded onto the page and before you attempt to use jQuery in your page



jQuery no-conflict mode

```
<script>
/* Creating alias for jQuery function */
jQuery.noConflict();

jQuery(document).ready(function( $ ) {
    /* You can use the locally-scoped $ here */
    $( "#hidden-para" ).hide();
    console.log("jQuery in no-conflict mode");
});

/* You can call other 3rd party library functions here
 * $(document).thirdPartyFunc();
 * The $ above is 3rd party function reference.
 */
</script>
```

*Thank
you*

WebStack Academy

#83, Farah Towers,
1st Floor, MG Road,
Bangalore – 560001

M: +91-809 555 7332

E: training@webstackacademy.com

WSA in Social Media:

