# jQuery

Là một viện của JavaScript giúp việc viết mã JS đơn giản hơn với lập trình viên.

Nó nhẹ nhanh và dễ mở rộng nên được sử dụng hầu hết các trang web.

Để học tốt jQuery các bạn cần biết về HTML, CSS, JavaScript, các khái niệm DOM, BOM

Các tính năng của jQuery:

* Thao tác HTML/ DOM
* Thao tác CSS
* Các Event
* Effects và animations
* AJAX
* Utilities

Ngoài ta còn các plugin phổ biến: jquery-validation để validate những giá trị từ các biểu mẫu. Hoặc data-table dùng để thao tác với bảng….Để hỗ trợ lập trình viên và mã code đơn giản, viết rất dễ dàng

## Cài đặt jQuery cho website.

jQuery là một thư viện JS đóng gói file .js, khi ta cài đặt jQuery cho website ta chỉ cần chèn file .js đó vào website.

Các cách chèn file .js vào website như các bạn đã biết khi học JS.

Link để chèn jQuery như sau:

* Tải và chèn file: js/jquery.js
* Sử dụng link Google CDN: <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js">
* Sử dụng link của Microsoft CDN: <script src="https://ajax.aspnetcdn.com/ajax/jQuery/jquery-3.3.1.min.js"></script>

Cách sử dụng link của Google hoặc Microsoft sẽ nhanh hơn so với việc tải và đặt trên website. Vì file có thể được lưu trong bộ nhớ của trình duyệt và nó sẽ được tải từ các máy chủ gần nhất của Internet.

## Cú pháp jQuery

Chú pháp jQuery như sau:

|  |
| --- |
| **$(*selector*).*action*()** |

* **selector** là các HTML element, hoặc thành phần bên trong của nó. **selector** bản chất theo cú pháp CSS. jQuery sử dụng cách chọn của CSS để chọn đối tượng của mình
* **$** để biết rằng đang sử dụng jquery
* **action** là các hàm, các thuộc tính của HTML element

Các ví dụ:

* $(this).hide() - ẩn element hiện tại
* $("p").hide() - ẩn tất cả các thẻ p.
* $(".test").hide() - ẩn cac element với class="test".
* $("#test").hide() - ẩn các element với id="test".
* $("p.test").hide() - ẩn các element với p với class="test".

Lưu ý:

* **selector** trong jQuery viết theo “phong cách” của CSS
* $ = jQuery là tên biến được tạo trong thư viện
* Hàm ready trong jQuery có 2 cách viết:

|  |
| --- |
| $(document).ready(function(){     *// jQuery methods go here...*  }); |

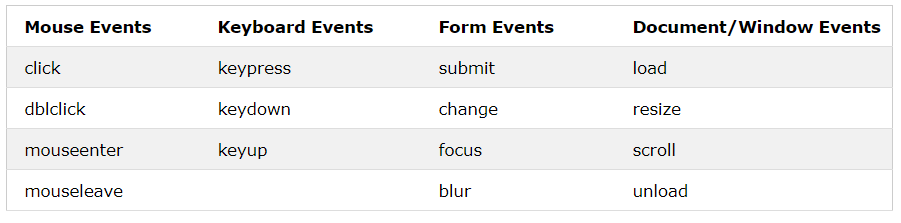
và

|  |
| --- |
| $(function(){     *// jQuery methods go here...*  }); |

Việc viết mã jQuery thực sự rất đơn giản nếu bạn biết chọn selector của CSS

## Event trong jQuery

Sử kiện được người dùng thao tác được gọi chung Event, sau đây là các Event:



Các events trong jQuery

Để cài đặt sự kiện cho events thì ta sử dụng hàm $(document).ready() của jQuery để cài đặt sẵn. Để hiểu rõ chúng ta sẽ xem ví dụ phần so sánh JS và jQuery

Có một số cách viết Event như sau:

### Cách viết 1: Sử dụng .click

|  |
| --- |
| $("p").click(function(){     $(this).hide(); }); |

### Cách viết 2: Sử dụng .on

|  |
| --- |
| $("p").on("click", function(){     $(this).hide(); }); |

### Ví dụ: sử dụng 'mousedown'

|  |
| --- |
| <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>  <script>  $(document).ready(function(){  $("#p1").on('mousedown', myFuntion);  });  function myFuntion(){  alert("Mouse down over p1!");  }  </script>  </head> |

# Một vài ví dụ so sánh cách viết JS và jQuery

## onclick và click

|  |  |
| --- | --- |
| **JS** | **jQuery** |
| <button onclick=”myFunction()”> <button> | Đặt class hoặc id để có thể tìm selector có thể không cần onlick  <button class=”myButton”> <button> |
| Viết hàm myFunction() trong cặp thẻ <script> và nhúng vào HTML:  <script>  //myFunction  function myFunction(){  }  </script> | Sử dụng “myButton” cài đặt click trong  $( document).ready() như sau:  //ready  $(document).ready(function(){     $(button.myButton").click(myFunction); });  //myFunction  function myFunction(){  } |

Nhìn có vẻ rắc rối hơn tuy vậy ưu điểm tên hàm myFunction sẽ không nằm trong mã HTML nữa. Điều đó giúp ta có thể tìm hàm và sử hàm khi cần thiết

## .style và .css (hoặc .attr)

|  |  |
| --- | --- |
| **JS** | **jQuery** |
| Element.style = “mã CSS”  Ví dụ:  var element = document.getElementsByClassName(‘myButton’)[0]  element.style = “color: red”;  Nếu không phải sử dụng phần tử 0 thì bạn phải làm vòng lặp for | Sử dụng “myButton” cài đặt click trong  $( document).ready() như sau:  $(.myButton")[0].css(‘color, ‘red’); });  $(.myButton")[0].attr(‘color’, ‘red’); |

Nhìn vào trong hàm trên ta thấy việc viết mã JS và jQuery hoàn toàn không có gì khác biệt. Khác biệt ở đây là việc sử mã jQuery đơn giản và dễ dàng sử dụng hơn, tối ưu code đạt được

# Các hàm cần biết trong jQuery

## Các hàm thao tác tìm DOM

* find()
* child() <=> parent, parents()
* hasClass, removeClass ⬄ addClass
* attr()

## Các hàm tạo animation

Thực sự thì chúng ta hiểu rằng CSS3 có thể tạo ra animation. Chúng ta có thể tạo ra một class animation và sử dụng jQuery để addClass animation kia. Đó là cách không cần viết mã JS.

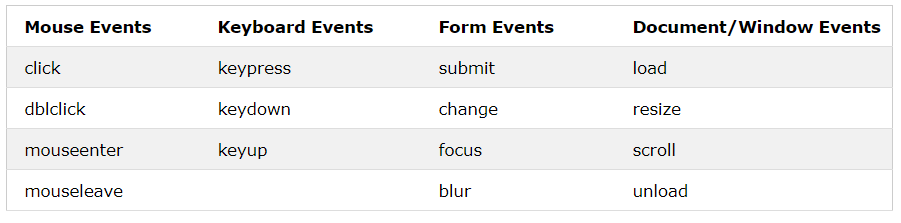
Ngoài ra chúng ta nên biết tới một vài hàm sau

* animate
* hide ⬄show
* fade
* slideUp ⬄slideDown
* stop

Tuy nhiên tùy vào bài toán phù hợp mà chúng ta sử dụng

## Cách hàm về Envent

Như bảng trên



Chúng ta sử dụng các hàm:

* .on()
* .off()
* bind() và unbind()
* .one()

# AJAX trong jQuery

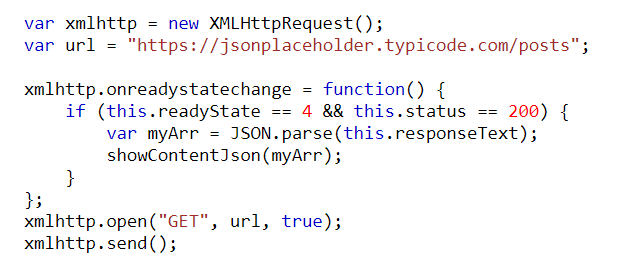
## Nhìn lại AJAX trong JS

Ajax trong jQuery đơn giản hơn rất nhiều trong JS. Vì XMLHTTPRequest phụ thuộc vào trình duyệt. Ví dụ IE6 không có XMLHTTPRequest thay thế bằng ActiveXObject

|  |
| --- |
| if (window.XMLHttpRequest) {  //Firefox, Opera, IE7, and other browsers will use the native object  var request = new XMLHttpRequest();  } else {  //IE 5 and 6 will use the ActiveX control  var request = new ActiveXObject("Microsoft.XMLHTTP");  } |

Khởi tạo XMLHTTPRequest với các trình duyệt

Và việc viết mã trong JS như sau:



## AJAX trong jQuery

Để đơn giản hóa code jQuery hỗ trợ cú pháp sau:

**$(*selector*).methodName(*URL,data,callback*);**

* Trong hàm callback chúng ta lưu ý các từ khóa sau:

Function callback có dạng sau:

|  |
| --- |
| function(responseTxt, statusTxt, xhr){  if(statusTxt == "success")  alert("successfully!");  if(statusTxt == "error")  alert("Error: " + xhr.status + ": " + xhr.statusText);  });  }); |

Với những từ khóa **bắt buộc** sau:

|  |  |
| --- | --- |
| AJAX - JQUERY | AJAX - JQUERY |
| ***responseTxt*** | responseText |
| ***statusTxt*** | Là statusText |
| ***xhr*** | Là đối tượng XMLHTTPRequest |

* **methodName** với các trường hợp dưới đây

## Các hàm thường sử dụng

### ajax()

Hàm này được cài đặt chung có thể sử dụng thay cho các hàm khác

### load()

**$(*selector*).load(*URL,data,callback*);.** Hàm này dùng để load một HTML từ một url của AJAX.

Sau đây là ví dụ minh họa đầy đủ:

|  |
| --- |
| $("#div1").load("demo\_test.txt", function(responseTxt, statusTxt, xhr){         if(statusTxt == "success")             alert("successfully!");         if(statusTxt == "error")             alert("Error: " + xhr.status + ": " + xhr.statusText);     }); |

### getJson()

Cú pháp như sau:

**$(*selector*).getJSON(*url,data,success(data,status,xhr))***

Sau đây là ví dụ: <https://www.w3schools.com/code/tryit.asp?filename=FRK761BTJUYJ>

|  |
| --- |
| $.getJSON("https://jsonplaceholder.typicode.com/posts", function(result){  console.log(result);  }); |

### .get()

**$.get(*URL,callback*);** Sử dụng với GET method

### .post()

**$.post(*URL,data,callback*);** Sử dụng với POST method

### serialize(). (chỉ dùng với form)

Với hàm này form với class formName sẽ lấy ra các dữ liệu input được nhập vào

Ví dụ: <https://www.w3schools.com/code/tryit.asp?filename=FRK7Z5IVHK06>