# Giới thiệu jQuery

## Giới thiệu jQuery

jQuery là một bộ công cụ JavaScript được thiết kế để đơn giản hóa các tác vụ khác nhau bằng cách viết ít code hơn. Dưới đây liệt kê một số tính năng tối quan trọng được hỗ trợ bởi jQuery:

+ **Thao tác DOM** − jQuery giúp dễ dàng lựa chọn các phần tử DOM để traverse (duyệt) một cách dễ dàng như sử dụng CSS, và chỉnh sửa nội dung của chúng bởi sử dụng phương tiện Selector mã nguồn mở, mà được gọi là Sizzle.

+ **Xử lý sự kiện** − jQuery giúp tương tác với người dùng tốt hơn bằng việc xử lý các sự kiện đa dạng mà không làm cho HTML code rối tung lên với các Event Handler.

+ **Hỗ trợ AJAX** − jQuery giúp bạn rất nhiều để phát triển một site giàu tính năng và phản hồi tốt bởi sử dụng công nghệ AJAX.

+ **Hiệu ứng động** − jQuery đi kèm với rất nhiều hiệu ứng động đẹp mà bạn có thể sử dụng cho các website của mình.

+ **Gọn nhẹ** − jQuery là thư viện gọn nhẹ - nó chỉ có kích cỡ khoảng 19KB (gzipped).

+ **Được hỗ trợ hầu hết bởi các trình duyệt hiện đại** − jQuery được hỗ trợ hầu hết bởi các trình duyệt hiện đại, và làm việc tốt trên IE 6.0+, FF 2.0+, Safari 3.0+, Chrome và Opera 9.0+.

+ **Cập nhật và hỗ trợ các công nghệ mới nhất** − jQuery hỗ trợ CSS3 Selector và cú pháp XPath cơ bản.

### So sánh một số cú pháp của Javascript với jQuery

* Gọi Class name : document.getElementsByClassName(‘className’) ⬄ $(‘.className’)
* Gọi Id name : document.getElementsById(‘idName’) ⬄ $(‘.idName’)
* Gọi Tag name: document.getElementsByTagName(‘tagName’) ⬄ $(‘.tagName’)

## Cài đặt thư viện jQuery

### Cài đặt nội bộ

**Bước 1**: Download jQuery.js file. (<https://jquery.com/download/>)

**Bước 2**: Thêm file jQuery,js vào file project HTML. <script src="jquery-3.5.1.js"></script>

**Bước 3**: Viết code trong thẻ <script>

<script>

        $(document).ready(function() {

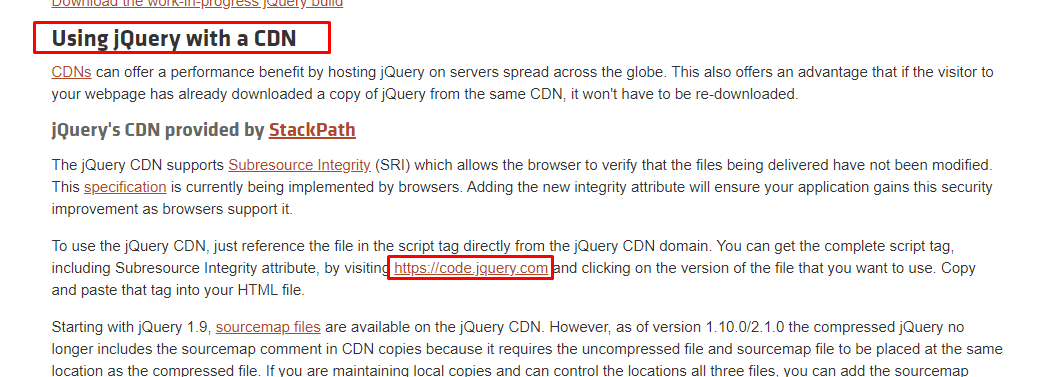
            alert(1);

        });

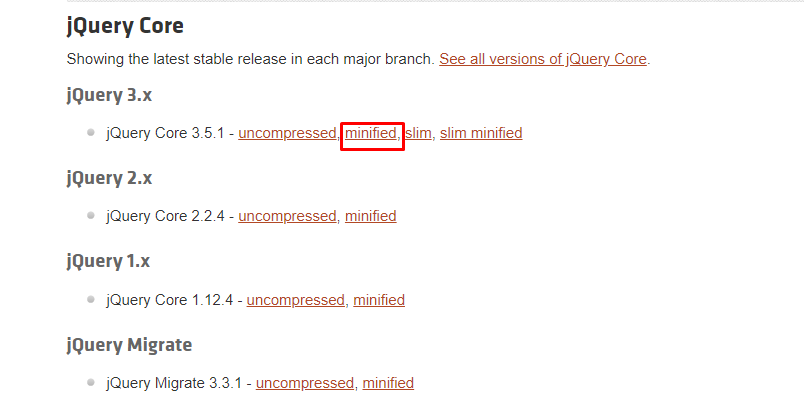
    </script>

### Cài đặt bằng CDN

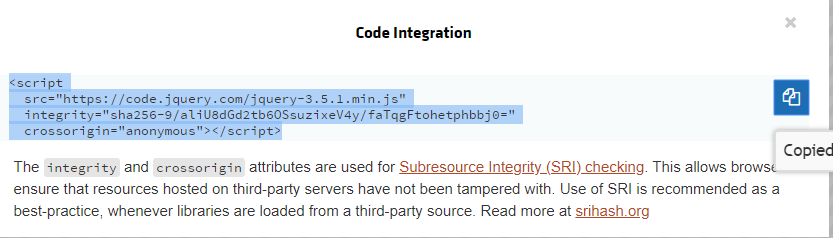
**Bước 1**: Lấy CDN jQuery. (<https://jquery.com/download/>)

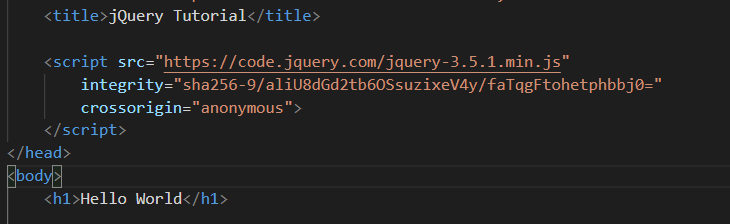


Hoặc truy cập: <https://code.jquery.com/>



**Bước 2**: Copy đoạn script và dán vào file HTML.





# Câu lệnh

## Cú pháp cơ bản

* Cú pháp mở đầu jQuery:

$(document).ready(function() {

     // Code jQuery

});

Đây là cú pháp bắt buộc phải có khi làm việc với jQuery

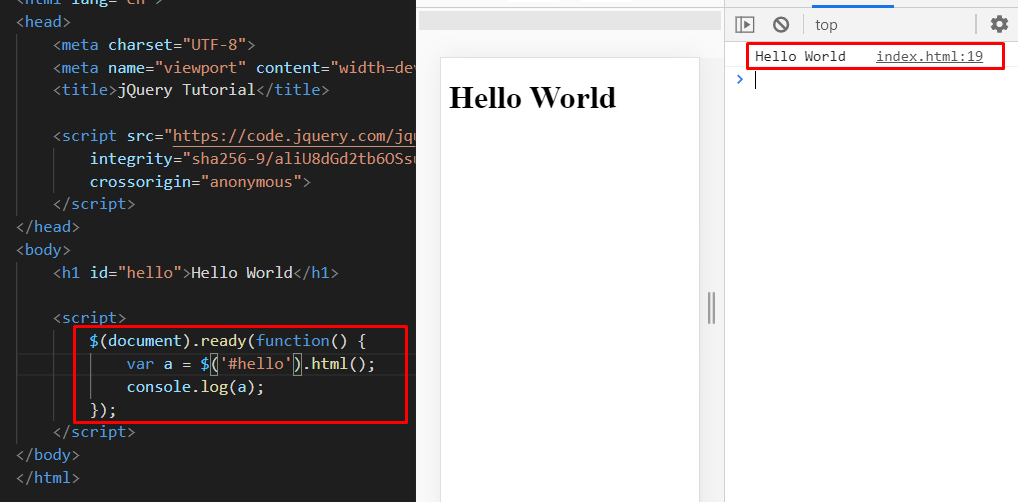
* **Cú pháp tổng quát**:

$(document).ready();

Với: + document sẽ được thay thế bằng các Selector

+ ready() sẽ được thay thế bằng các hàm có sẵn của jQuery

VD:



Cách khai báo một số Selector:

* $(‘#idName’): Các element có id
* $(‘.className’) : Các element có class
* $(window) : Khai báo khung nhìn windows
* $(‘p’) : Các elment cùng thể loại
* $(‘\*’) : Tất cả các elment,…
* $(this) : Element đang được lựa chọn.
* Có thể tham khảo thêm cách khai báo selector tại: <https://www.w3schools.com/jquery/jquery_ref_selectors.asp>
* **Câu lệnh kết hợp:**

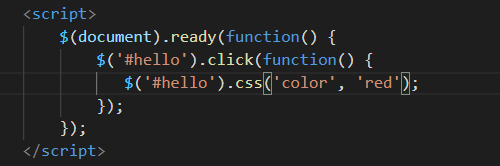
$('#box2').slideUp(1000).css('background','pink').slideDown(1000);

## Events trong jQuery

### Sự kiện với chuột trong jQuery

* ***Click()*** : Thay đổi khi click vào element
* ***Dblclick()*** : Thay đổi khi click 2 lần vào element
* ***Contextmenu()*** : Thay đổi khi chuột phải vào element
* ***Mouseenter()*** : Thay đổi khi di chuột vào element
* ***Mouseleave()*** : Thay đổi khi di chuột ra khỏi element
* Có thể tham khảo thêm một số events với chuột tại: <https://www.w3schools.com/jquery/jquery_ref_events.asp>

VD:



### Sự kiện với bàn phím trong jQuery

* ***Keypress()***: Sự kiện được chạy khi một phím được nhấn trên bàn phím và sự kiện sẽ không chạy khi ta ấn các phím không hiển thị được trên màn hình. Và khi xoá các ký tự thì Keypress() sẽ không xoá được
* ***Keydown()***: Sự kiện được chạy khi một phím được nhấn trên bàn phím và sự kiện cũng được chạy khi ta ấn các phím không hiển thị được trên màn hình (ví dụ CTRL hoặc ALT hoặc SHIFT)
* ***Keyup()***: Sự kiện được chạy khi một phím được nhả ra khỏi bàn phím
* Có thể tham khảo thêm một số events với bàn phím tại: <https://www.w3schools.com/jquery/jquery_ref_events.asp>

VD:

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>jQuery Tutorial</title>

    <script src="https://code.jquery.com/jquery-3.5.1.min.js"

        integrity="sha256-9/aliU8dGd2tb6OSsuzixeV4y/faTqgFtohetphbbj0="

        crossorigin="anonymous">

    </script>

</head>

<body>

    <input type="number" name="" id="a"><br>

    <input type="number" name="" id="b"><br>

    <h1 id="hello">KQ với keyup: <span id="sum"></span> </h1>

   <input type="text" name="" id="test"><br>

   <h1>Keydown: <span id="keydown"></span> </h1>

    <h1>Keypress: <span id="keypress"></span> </h1>

    <h1>Keyup: <span id="keyup"></span> </h1>

    <script>

        $(document).ready(function() {

            var a = 0;

            var b = 0;

            var c;

            $('#a, #b').keyup(function() {

                a = $('#a').val();

                b = $('#b').val();

                c = parseInt(a)+parseInt(b);

                $('#sum').text(c);

            });

            $('#test').keydown(function() {

                var test = $('#test').val();

                $('#keydown').text(test);

            });

            $('#test').keypress(function() {

                var test = $('#test').val();

                $('#keypress').text(test);

            });

            $('#test').keyup(function() {

                var test = $('#test').val();

                $('#keyup').text(test);

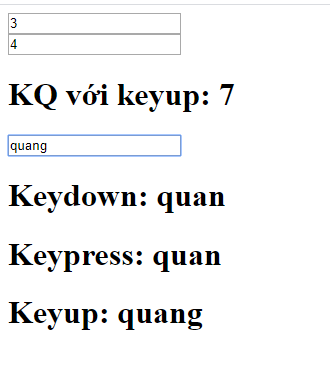
            });

        });

    </script>

</body>

</html>



### Sự kiện với form trong jQuery

* **Focus()**: Khi nhấn vào filed thì sự kiện sẽ được thực thi
* **Blur()**: Khi dời khỏi filed thì sự kiện mới được thực thi
* **Change()**: Khi dời khỏi filed mà trong filed đó phải có sự thay đổi text thì sự kiện mới được thực thi
* **Select()**: Khi chúng ta bôi đen các dữ liệu trong filed thì sự kiện được thực thi
* **Submit()**: Khi chúng ta nhấn vào nút submit thì sự kiện được thực thi. Lưu ý: Để sự kiện được thực thi thì ta lấy id của form để khai báo sự kiện cho nút submit

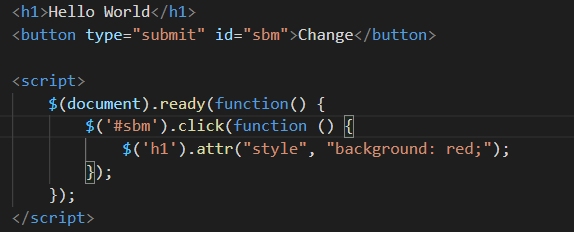
### Sự kiện với windows trong jQuery

* Nếu sử khai báo cho windows thì ta sử dụng cú pháp: $(window).scroll()
* **Scroll()**: Sự kiện xảy ra khi trên window hoặc trên element có thay đổi theo chiều dọc
* **Resize()**: Sự kiện xảy ra khi trên window hoặc trên element có thay đổi theo chiều ngang

### Sự kiện khi truyền phương thức

* **Val()**: Lấy biến từ filed
* **Html()**: Thay đổi HTML của element
* **Text()**: Thêm text vào trong element cần thiết
* **Attr()**: Thay đổi thuộc tính của element, hoặc có thể thay đổi tên một class hoặc id

VD:

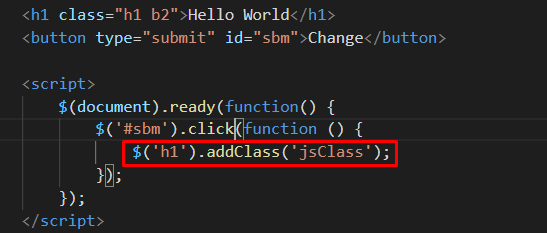




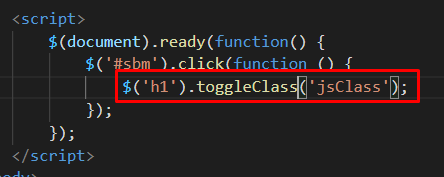
### Sự kiện với phương thức class

* **AddClass()**: Thêm một class vào element



* **RemoveClass()**: Xoá một class khỏi element
* **ToggleClass()**: Thêm và xoá một class được thay đổi luân phiên nhau khi click vào.

VD:



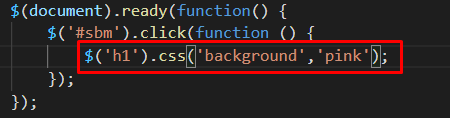
## Method jQuery

### CSS() Method

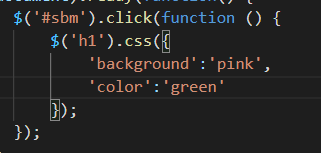
* **Css()**: Thay đổi một hoặc nhiều thuộc tính CSS của element

VD:

+) Thay đổi một thuộc tính CSS



+ Thay đổi nhiều thuộc tính CSS



### Animate() Method

* Animate() giúp tạo hiệu ứng di chuyển cho element

Cú pháp: **$(selector).animate({params},speed,callback);**

Với: +) Params: Gồm các thuộc tính sẽ được thay đổi sau khi thực thi animate

+) Speed: Thời gian để thực hiện animate

+) Callback: Gọi lại một hàm nữa khi đã thực hiện xong hiệu ứng

$('#cloneBtn').click(function () {

$('#box2').animate({

     'width': '150px',

        'height': '150px',

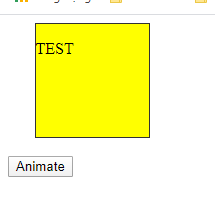
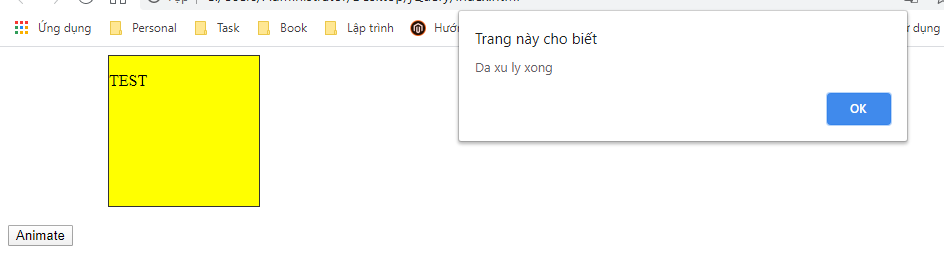
        'margin-left': '100px'

      },3000,function () {

       alert('Da xu ly xong');

      });

});

### Stop() Method

* Dùng để dừng hiệu ứng animate()

VD:

$('#stopBtn').click(function () {

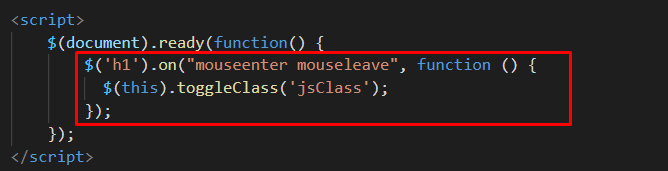
$('#box2').stop();

});

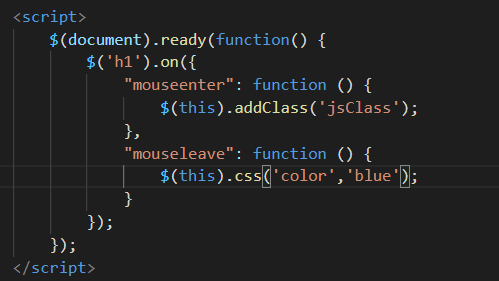
### On(), Off() Method

* **On()**: Nếu ta muốn bật nhiều sự kiện cùng lúc như: click(), mouseenter(), mouseleave(),.. mà không cần phải khai báo lại cú pháp giống nhau thì ta sử dụng phương thức on() của jQuery

VD:

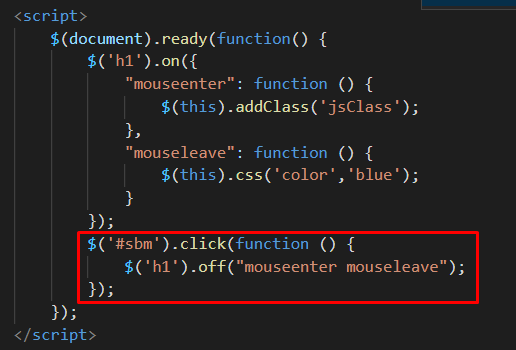


Hoặc cũng có thể viết như sau:



* **Off()**: Tắt hết các sự kiện được khai báo trong hàm

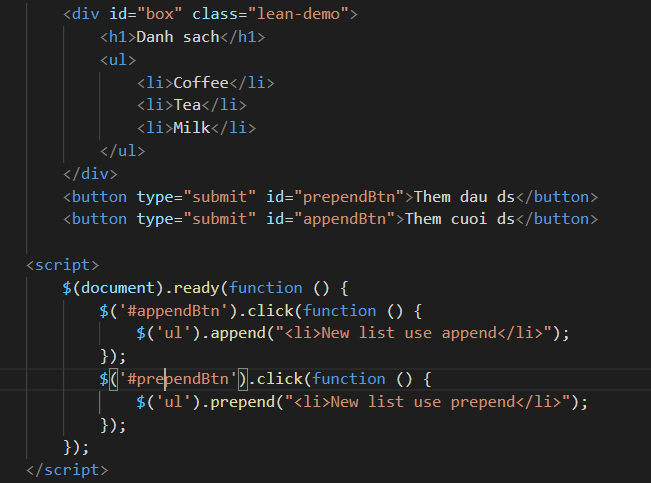
VD:

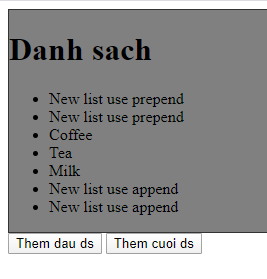


### Append(), Prepend() Method

* **Append()**: Có thể thêm text hoặc html ở bên trong một khối nhưng ở phía dưới
* **Prepend()**: Có thể thêm text hoặc html ở bên trong một khối nhưng ở phía trên

VD:

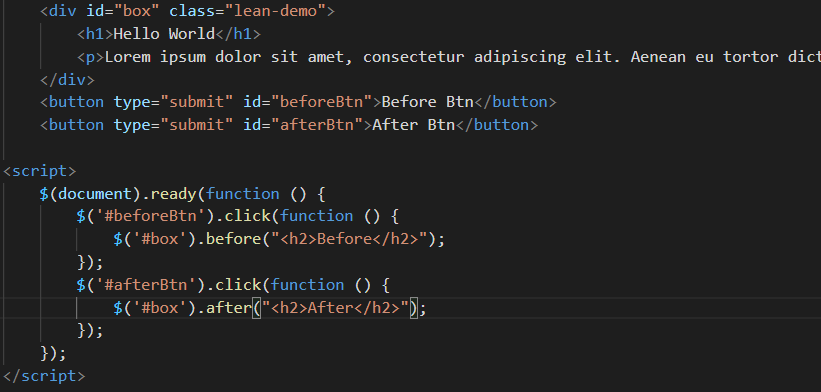




### After(), Before() Method

* After(): Có thể thêm text hoặc html ở bên ngoài một khối nhưng ở phía dưới
* Before(): Có thể thêm text hoặc html ở bên ngoài một khối nhưng ở phía trên

VD:





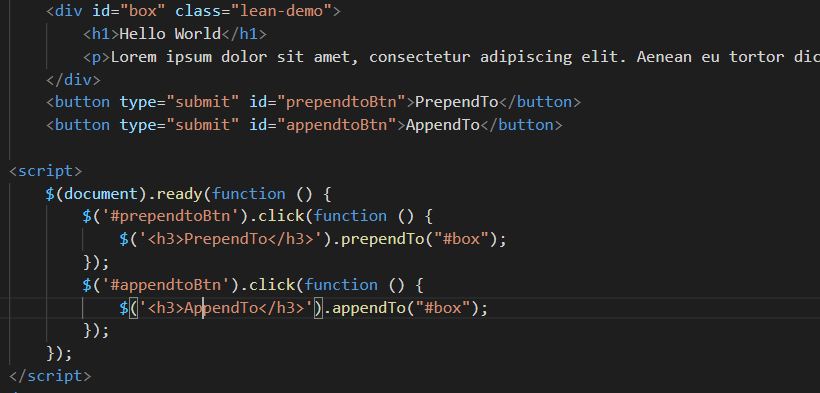
### Empty(), Remove() Method

* **Empty()**: Khi hàm này được thực thi nó sẽ xoá dữ liệu bên trong thẻ
* **Remove()**: Khi hàm này được thực thi nó sẽ xoá luôn thẻ đó.

### AppendTo(), PrependTo() Method

* Các hàm này ngược với Append() và Prepend()

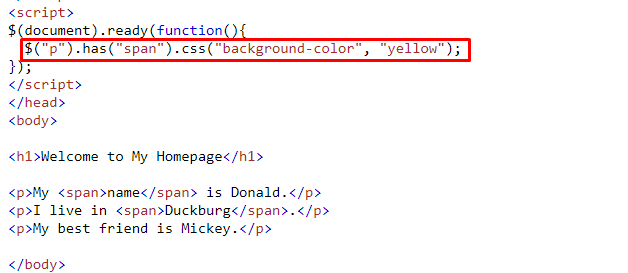
VD:

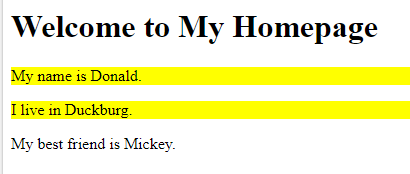




### Has(), Is() Method

* **Has()**: Phương thức trả về tất cả các phần tử có một hoặc nhiều phần tử bên trong chúng, khớp với bộ chọn được chỉ định.



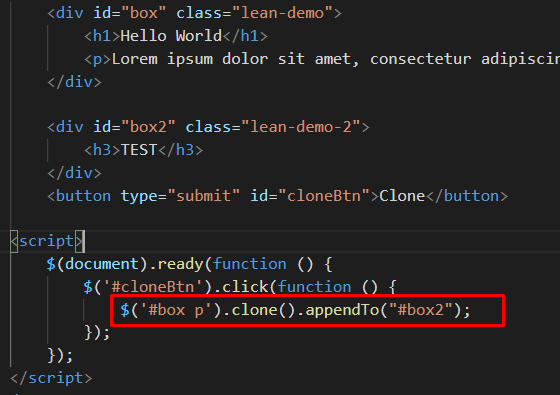


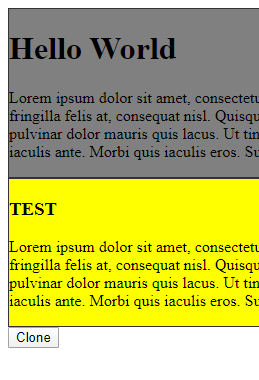
* **Is()**:

### Clone() Method

* **Clone()**: Là hàm dùng để sao chép trong jQuery. Thường được đi kèm bởi các hàm Append() và Prepend()

VD:

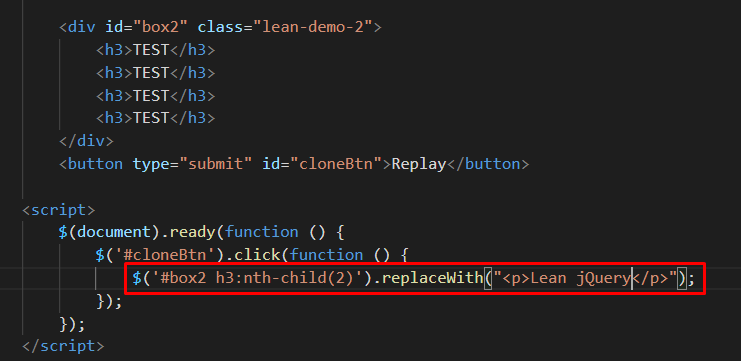


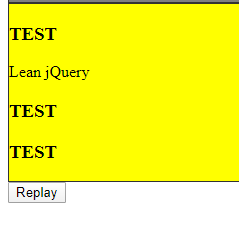


### ReplaceWith(), ReplaceAll() Method

* Các hàm ReplaceWidth() và ReplaceAll() dùng để thay thế text trong html

VD:

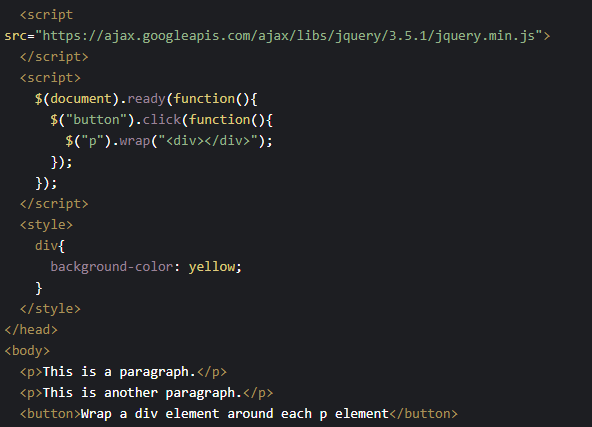


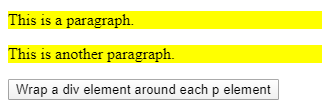


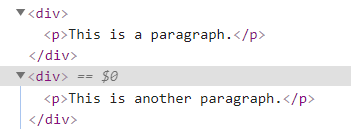
### Wrap() & UnWrap(), WrapAll(), WrapInner() Method

* **Wrap()**: Hàm này sẽ tìm các thẻ được khai báo và bao ngoài từng thẻ một bằng một thẻ html trong dấu ().
* **UnWrap()**: Loại bỏ thẻ đã được chọn trong dấu ().
* **WrapAll()**: Hàm này sẽ tìm các thẻ được khai báo và bao ngoài các thẻ đó bằng một thẻ html trong dấu ().
* **WrapInner()**:

VD:





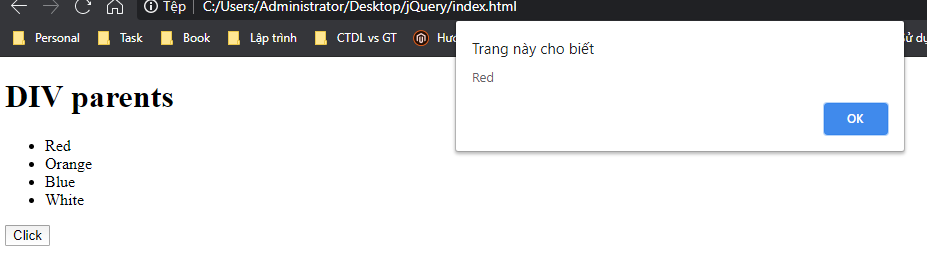


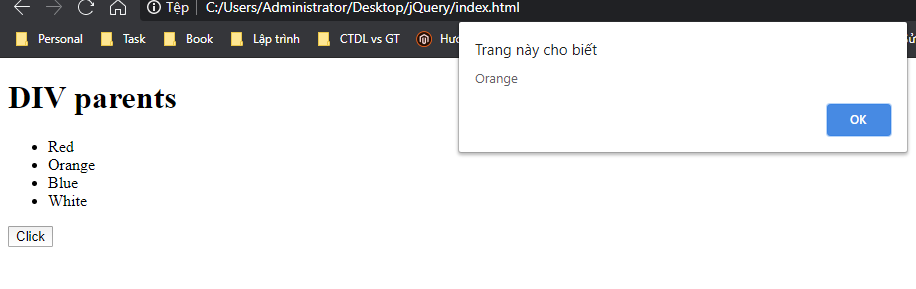
### Vòng lặp

* Hàm each() là hàm tạo vòng lặp trong jQuery.

VD:







## Effect jQuery

### Hide(), Show(), Toggle()

* **Hide()**: Ẩn một element và nó được thực hiện ẩn ngay khi phương thức đó chạy

$('#cloneBtn').click(function () {

$('#box2 p').hide();

});

* **Show()**: Hiện một element và nó được thực hiện ẩn ngay khi phương thức đó chạy

$('#cloneBtn').click(function () {

    $('#box2 p').show();

});

* Toggle(): Ẩn hiện element theo kiểu so le và nó được thực hiện ẩn ngay khi phương thức đó chạy

$('#cloneBtn').click(function () {

     $('#box2 p').toggle();

});

Chú ý: hide(), show(), toggle() cũng có thể viết hide(1000), show(1000), toggle(1000) có nghĩa là các hiệu ứng sẽ thực thi trong một giây

### FadeIn(), FadeOut(), FadeToggle(), FadeTo()

* **FadeIn()**: Hiện một element và nó được thực hiện mờ dần khi phương thức đó chạy. Tạo hiệu ứng chạy đẹp hơn.
* **FadeOut()**: Ẩn một element và nó được thực hiện mờ dần khi phương thức đó chạy. Tạo hiệu ứng chạy đẹp hơn.
* **FadeToggle()**: Ẩn, hiện một element so le với nhau và nó được thực hiện mờ dần khi phương thức đó chạy. Tạo hiệu ứng chạy đẹp hơn.
* **FadeTo()**: Làm mờ một element và nó được thực hiện mờ dần khi phương thức đó chạy.

VD:

$('#cloneBtn').click(function () {

   $('#box').fadeTo(3000,0.5);

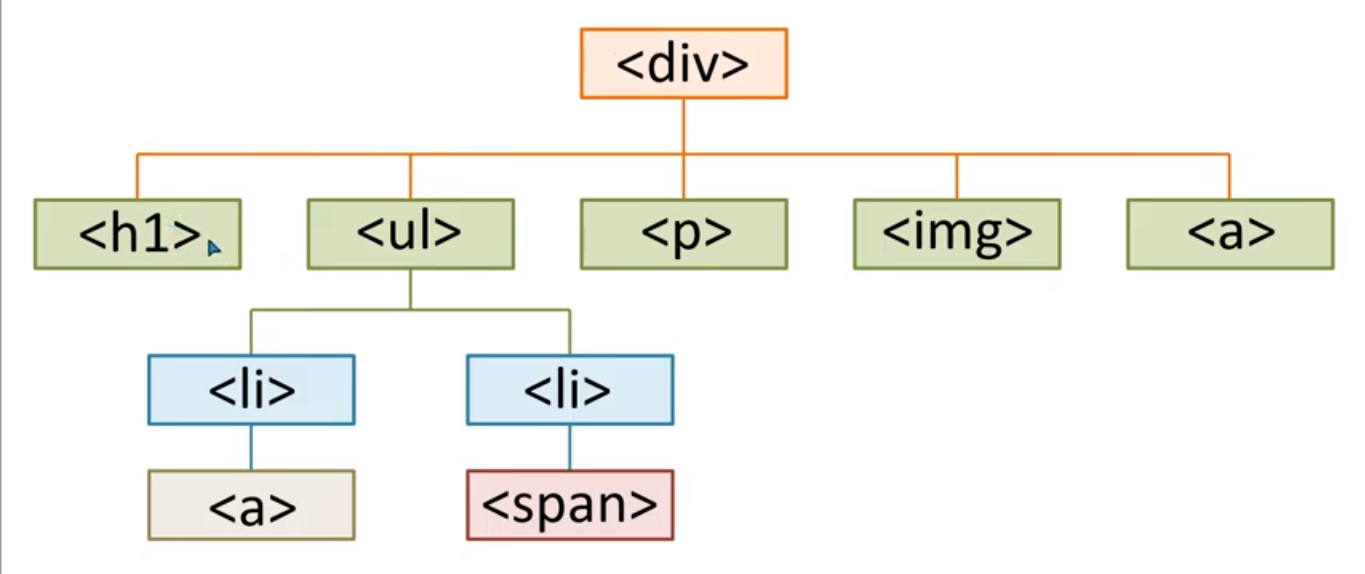
});

### SlideUp(), SlideDown(), SlideToggle()

* **SlideUp()**: Dùng để ẩn một element có kèm theo hiệu ứng trượt.
* **SlideDown()**: Dùng để hiện một element có kèm theo hiệu ứng trượt.
* **SlideToggle()**: Dùng để ẩn, hiện so le một element có kèm theo hiệu ứng trượt.

# JQuery Traversing

Cây DOM:



* JQuery tree thể hiện các element có mối quan hệ cha con.

## Ancestors Methods

* **Parent()**: Các hiệu ứng sẽ được áp dụng cho lớp cha trên một cấp của lớp đó trong cây DOM. Nó chỉ duyệt duy nhất một lớp cha ở trên.

$('.div-item').parent().css('background','red');

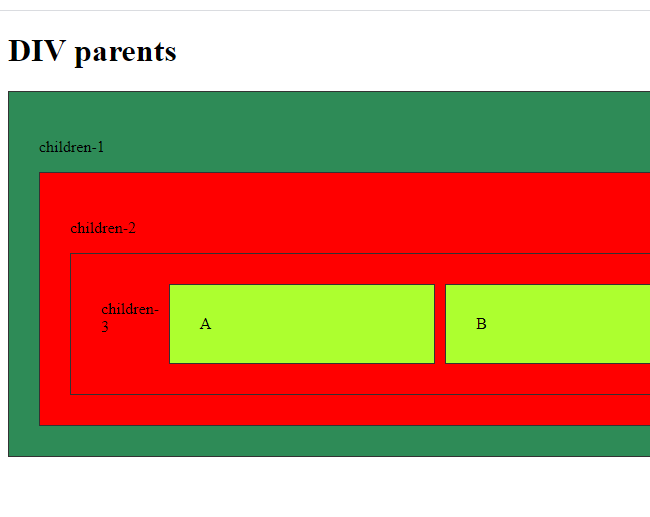
* **parents()**: Các hiệu ứng sẽ được áp dụng cho các lớp cha của lớp đó trong cây DOM. Nó duyệt các lớp cha ở trên.

$('.div-item').parents().css('background','red');

* **parentsUntil()**: Các hiệu ứng sẽ được áp dụng từ lớp cha đầu tiên của lớp được khai báo cho đến dưới lớp được khai báo trong tham số truyền vào. Nó duyệt các lớp cha ở trên.

VD:

$('.div-item').parentsUntil('.children-1').css('background','red');



* **closest()**: Chạy vào trong cây DOM để tìm những phần tử phù hợp với tham số truyền vào.

$('.children-2').closest('div').css('color','red');

* **offsetParent()**: Các hiệu ứng sẽ được áp dụng cho lớp cha đầu tiên trong cây DOM. Hiệu ứng có thể áp dụng cho thuộc tính position: relactive

$('.children-2').offsetParent().css('background','red');

## Descendants Methods

* **Children()**: Khi khai báo hiệu ứng này thì tất cả các lớp con bên trong lớp được khai báo đều nhận hiệu ứng.

$('.children-3').children().css('background-color','red');

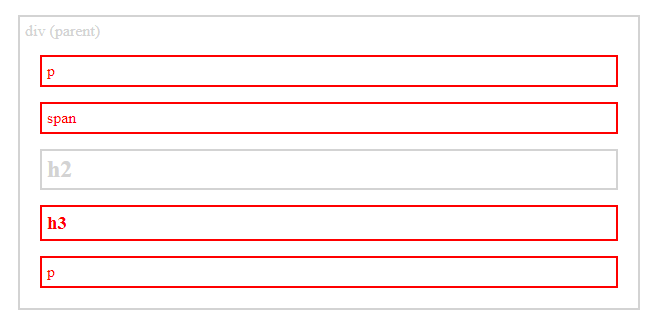
* **Find()**: Khi khai báo hiệu ứng này thì jquery sẽ tìm trong lớp đã khai báo lúc xem phần tử nào phù hợp với tham số truyền vào thì mới áp dụng hiệu ứng

$('.children-3').find('.div-item-2').css('background-color','red');

## Siblings Method

* **Siblings()**: Trả về tất cả các element anh em của thành phần được chọn

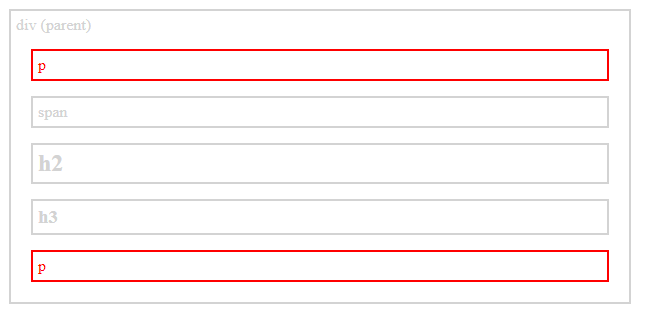




**Chú ý**: Ta cung có thể tuỳ chọn các element anh em theo ý của mình

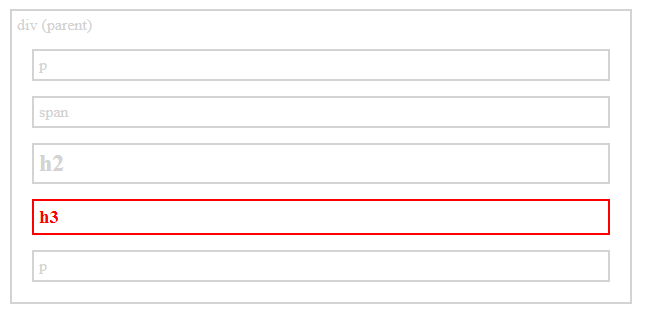
**VD**: Lấy tất cả các phần tử là anh em của h2 và là thẻ p





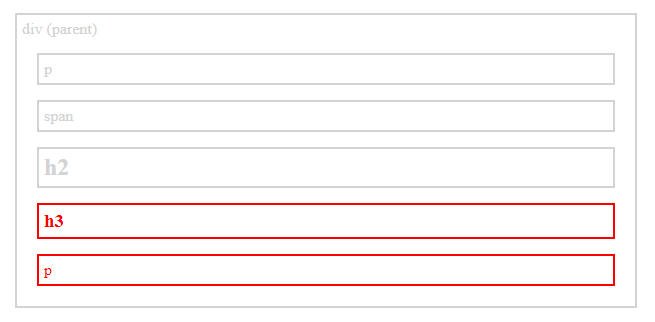
* **Next()**: Trả về các một phương thức anh em tiếp theo của phần tử được chọn



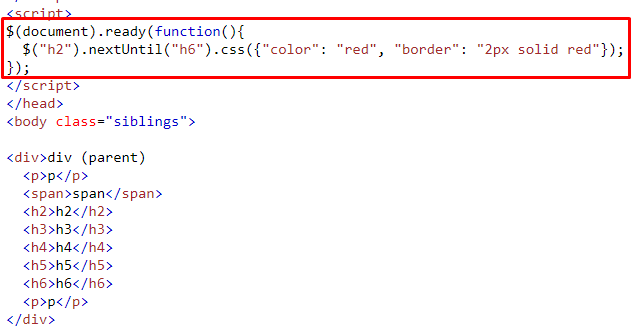


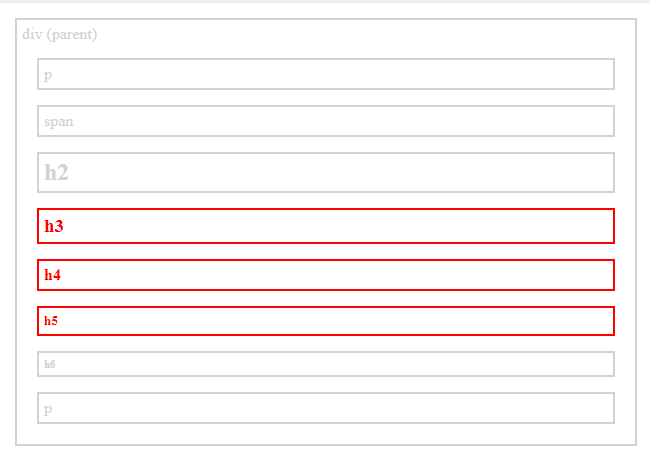
* **NextAll()**: Trả về các tất cả các phương thức anh em tiếp theo của phần tử được chọn





* **NextUntil()**: Trả về các các phương thức anh em tiếp theo giữa hai phần tử được chọn

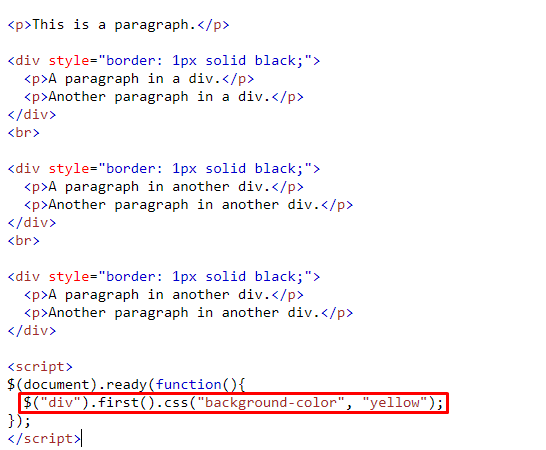


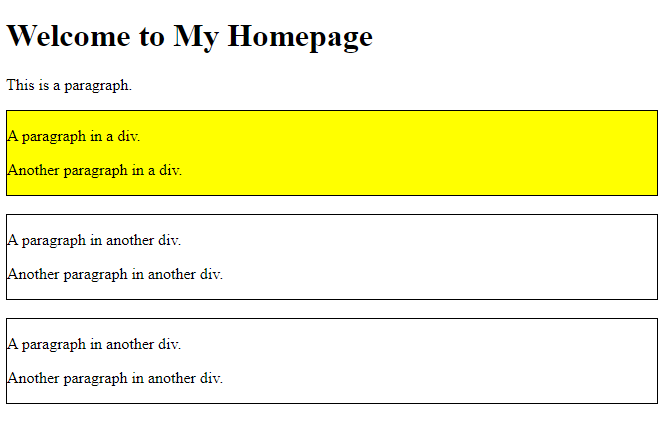


* Các phương thức prev(), prevAll() và prevUntil() tương tự như các phương thức next nhưng ngược lại.

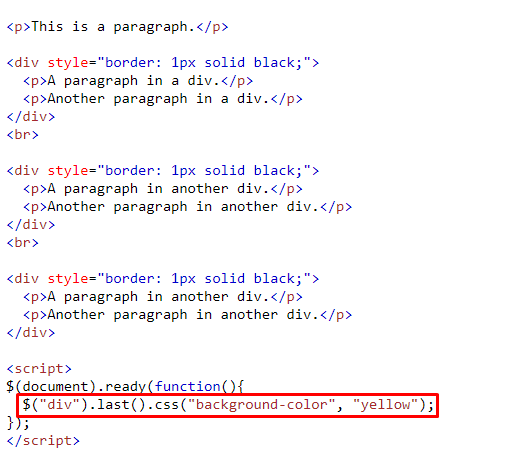
## Filtering Methods

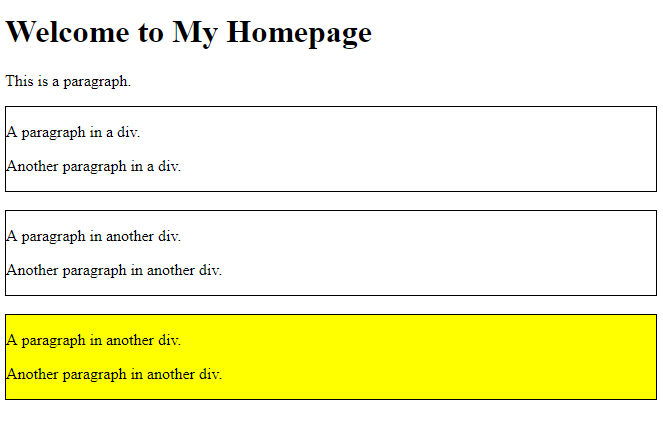
* Cho phép ta chọn một yếu tố cụ thể dựa trên vị trí của nó trong một nhóm các yếu tố này.
* **first()**: Phương thức sẽ áp dụng cho phần tử đầu tiên của các yếu tố quy định



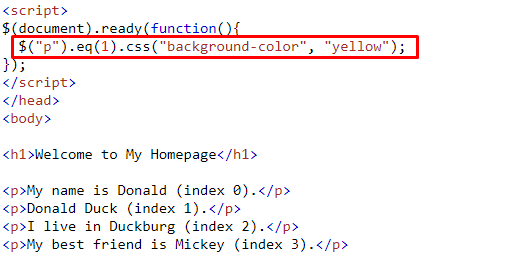


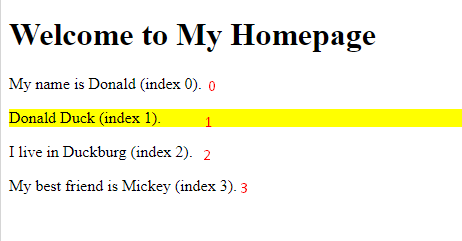
* **last()**: Phương thức sẽ áp dụng cho phần tử cuối cùng của các yếu tố quy định



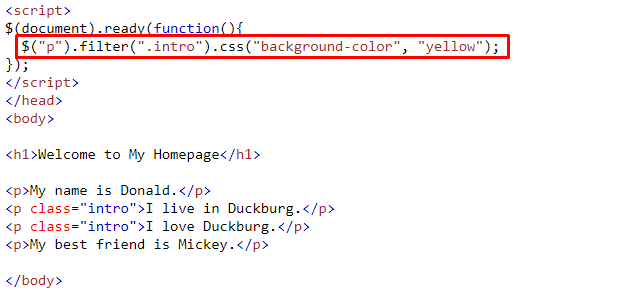


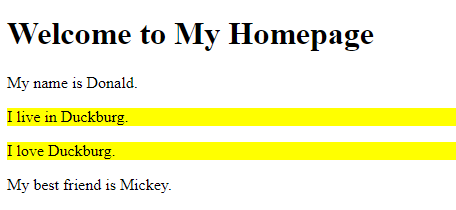
* **eq()**: Phương thức sẽ áp dụng cho phần tử được truyền vào chỉ số



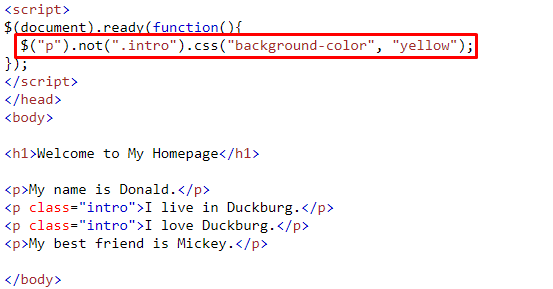


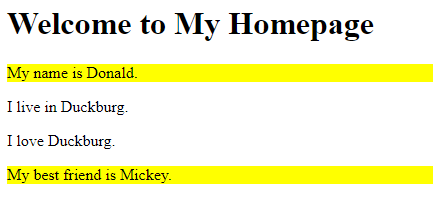
* **filter()**: Phương thức sẽ áp dụng cho phần tử đáp ứng được tiêu chí được truyền vào.



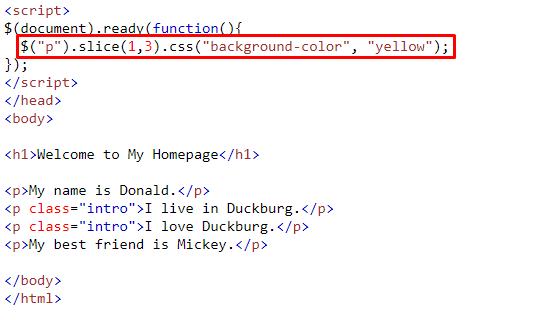


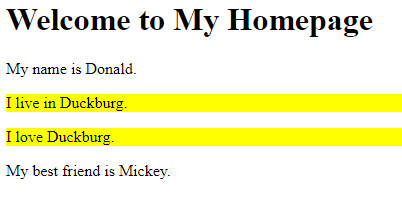
* **not()**: Phương thức sẽ áp dụng cho phần tử không đáp ứng được tiêu chí được truyền vào. Ngược lại với filter()





* **Slice()**: Phương thức sẽ áp dụng cho các phần tử được khai báo chỉ số cho đến các phần tử cuối danh sách. Hoặc ta cũng có thể khai báo điểm bắt đầu và điểm kết thúc của danh sách đó.





# Plugins Jquery

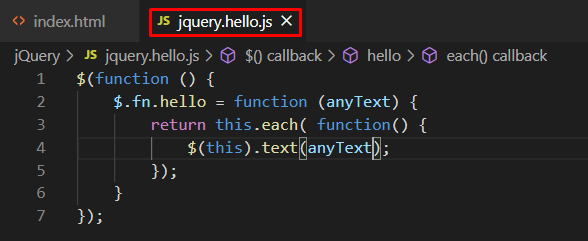
## Tự tạo Plugin

**Bước 1**: Gọi plugin vào Project



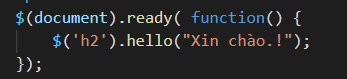
**Bước 2**: Tạo Plugin

* Với bước 2 nếu chúng ta muốn lấy plugin bên ngoài vào thì ta download đó về và copy file js vào thư mục source

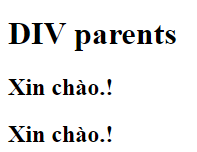


Với: $.fn: Bắt đầu định nghĩa plugin.

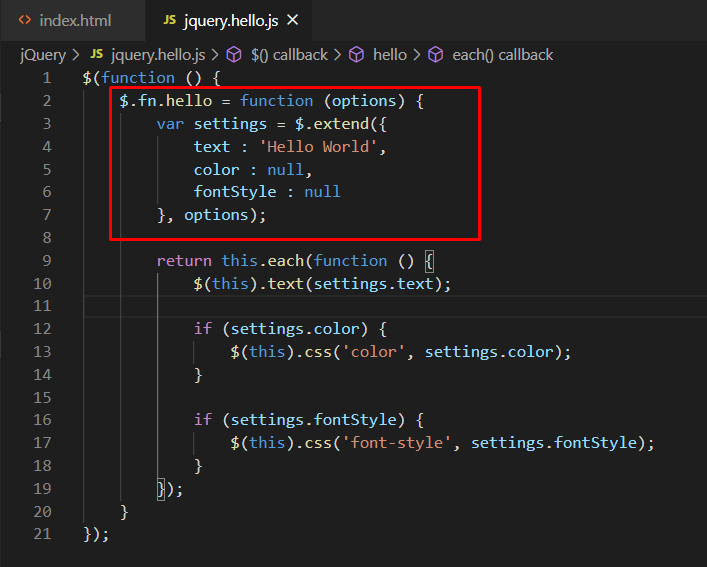
**Bước 3**: Áp dụng plugin



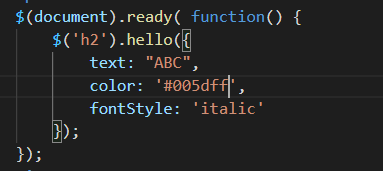
KQ:



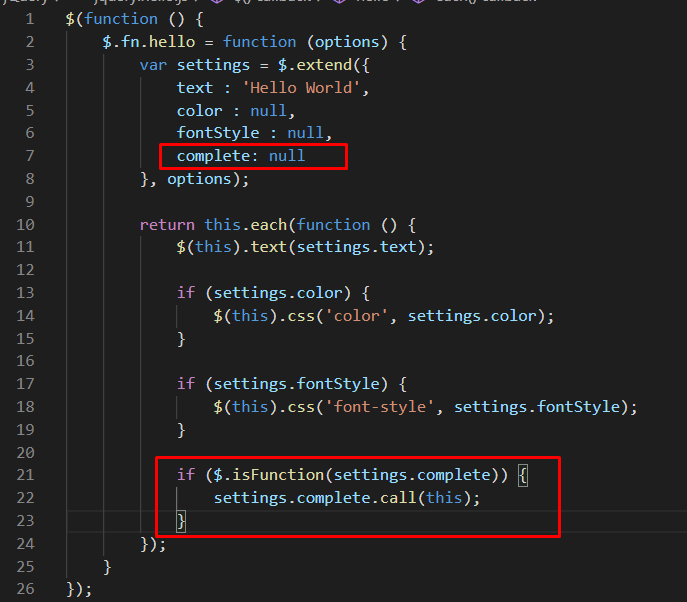
* Ta có thể thay đổi anyText bằng một option như sau:



* Trong file index.html khai báo như sau:



* Chúng ta thêm 1 biến complete vào settings:



* Trong index.html



KQ:

