



# Front-end Essentials

*jQuery*



- jQuery được tạo ra bởi John Resig trong năm 2006
- Nó được thiết kế để xử lý các vấn đề:
  - ✓ Không tương thích trên các trình duyệt
  - ✓ Đơn giản hóa HTML DOM
  - ✓ xử lý sự kiện
  - ✓ Hiệu ứng
  - ✓ Ajax
  - ✓ Đa nền tảng
- Hiện tại jQuery đang được sử dụng khá phổ biến



Nếu bạn đã nắm được kiến thức cơ bản về Javascript thì việc sử dụng JQuery không có gì quá khó

- Cách 1: Tải trực tiếp thư viện jQuery trên trang chủ:  
<https://jquery.com/download/>
- Cách 2: Liên kết thư viện jQuery thông qua CDN của Google:  
<https://developers.google.com/speed/libraries#jquery>
- Cách 3: Liên kết thư viện jQuery thông qua CDN của Microsoft: <https://ajax.aspnetcdn.com/ajax/jquery/jquery-3.4.1.js>

- Để đảm bảo các đoạn mã JQuery không chạy trước khi trang đã được tải xong → cần sử dụng một sự kiện xác định sẵn tài liệu sẵn sàng:
- Cú pháp:

```
<!-- Liên kết thư viện jquery -->  
<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
```

```
<script>  
    $(document).ready(function(){  
        // Các đoạn JQuery  
    });  
</script>
```

Hoặc

```
<script>  
    $(function(){  
        // Các đoạn JQuery  
    });  
</script>
```

# window.onload() vs \$(document).ready()

```
<script>
    window.onload = function () {
        alert('window loaded');
    };

    $(document).ready(function () {
        alert('document loaded');
    });
</script>
```



**Cái nào sẽ hiển thị trước?**

## ■ Lấy phần tử trong HTML

- ✓ Trong Javascript thuần, muốn lấy một phần tử trong HTML bạn sẽ có cú pháp `document.getElementById|ClassName|TagName("element")`
- ✓ Trong JQuery để lấy ra một phần tử trong HTML chỉ cần sử dụng **`$(selector)`**:
  - `$(This)` : Lấy phần tử hiện tại.
  - `$("h1")` : Lấy các phần tử thẻ **H1**.
  - `$(".Test")` : Lấy các phần tử có class **“Test”**.
  - `$("# Test")` : Lấy phần tử có ID là **“Test”**.
  - `$("*")` : Lấy tất cả phần tử.
  - `$("p.intro")` : Lấy các phần tử p có class là **“intro”**
  - `$("p:first")` : Lấy phần tử **p** đầu tiên trong tài liệu HTML.

## ■ Lấy phần tử trong HTML với jQuery:

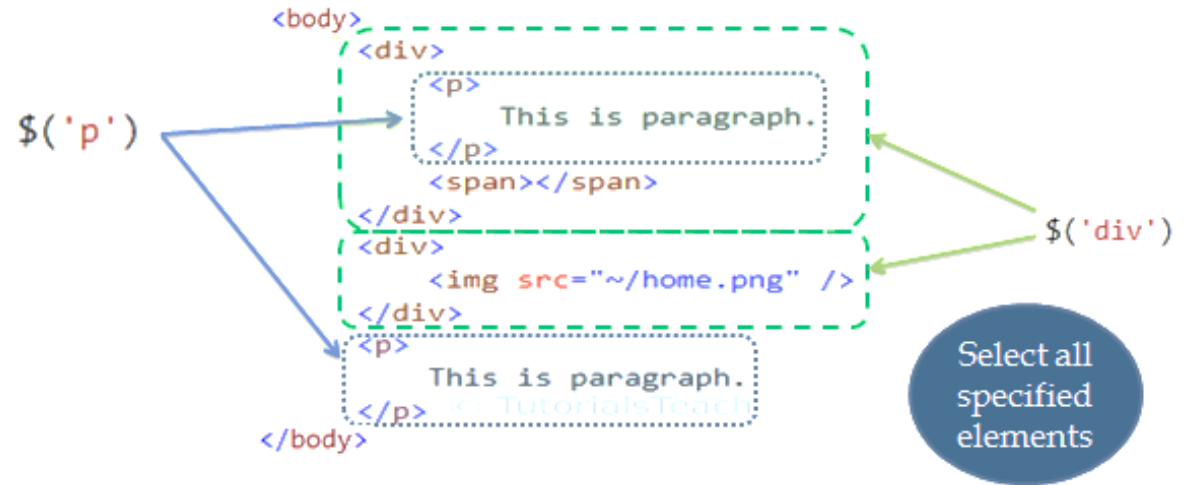
- ✓ `$("p:last")` : Lấy phần tử **p** cuối cùng trong tài liệu HTML.
- ✓ `$("ul li:first")` : Lấy phần tử **li** đầu tiên trong phần tử **ul**
- ✓ `$("[href]")` : Lấy các phần tử có thuộc tính **href**.
- ✓ `$("a[target='_blank']")` : Lấy các phần tử a có thuộc tính **target='\_blank'**
- ✓ `$("a[target!='_blank']")` : lấy các phần tử a có thuộc tính **target** nhưng không phải là **'\_blank'**
- ✓ `$(":button")` : Lấy các phần tử button hoặc các phần tử input có loại là button.
- ✓ `$("tr:even")` : Lấy các phần tử **tr** chẵn trong bảng.
- ✓ `$("tr:odd")` : Lấy các phần tử **tr** lẻ trong bảng.

# Xác định phần tử HTML thông qua tên thẻ

```
<div>  
    <p></p>  
    <p></p>  
</div>  
<p></p>  
<div></div>
```



This is paragraph.  
This is paragraph.  
This is div.  
This is paragraph.  
This is div.

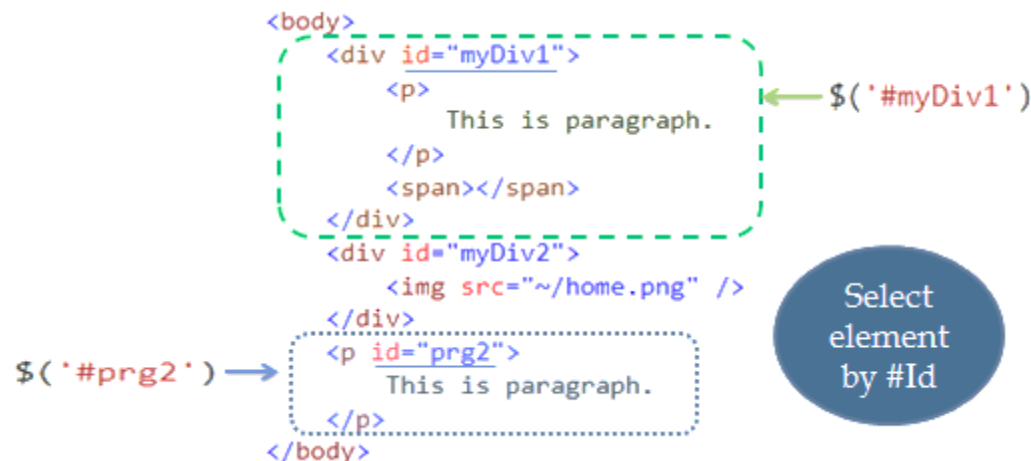


```
<script>  
    $(document).ready(function () {  
  
        $('p').append('This is paragraph.');// appends text to all p elements  
  
        $('div').append('This is div.');// appends text to all div elements  
  
    });  
</script>
```



# Xác định phần tử HTML thông qua id

```
<div id="myDiv1" >  
  <p></p>  
</div>  
<p id="impPrg"></p>  
<div id="myDiv2"></div>
```



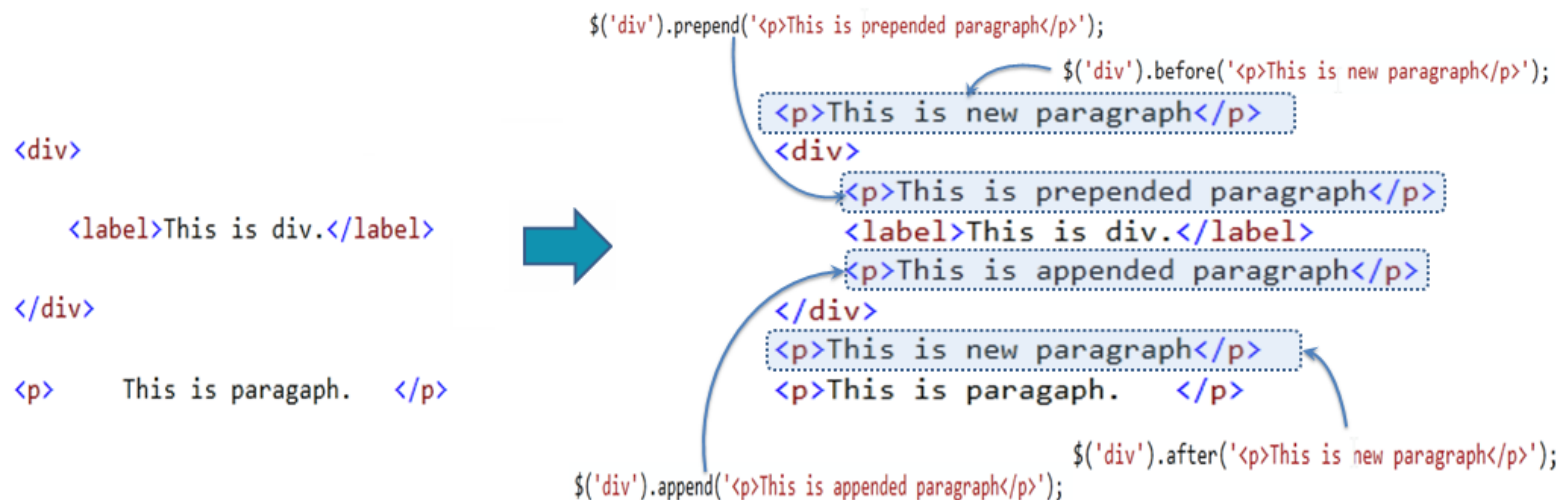
```
$(document).ready(function () {  
  $('#impPrg').append('This element\'s id is "impPrg"');  
  $('#myDiv2').append('This element\'s id is "myDiv2"');  
});
```



This is paragraph.This element's id is "impPrg"

This is div.This element's id is "myDiv2"

| Phương thức  | Mô tả  |
|--------------|--|
| append()     | Chèn nội dung vào cuối (các) phần tử được chỉ định bởi bộ chọn.                                |
| before()     | Chèn nội dung (các phần tử DOM mới hoặc hiện tại) trước một phần tử được chỉ định bởi bộ chọn. |
| after()      | Chèn nội dung (các phần tử DOM mới hoặc hiện tại) sau một phần tử được chỉ định bởi bộ chọn.   |
| prepend()    | Chèn nội dung vào đầu (các) phần tử được chỉ định bởi bộ chọn.                                 |
| remove()     | Loại bỏ các yếu tố (s) từ DOM được chỉ định bởi bộ chọn.                                       |
| replaceAll() | Thay thế (các) phần tử đích với phần tử được chỉ định.   |
| wrap()       | Bao ngoài thành phần được chọn bởi một cấu trúc HTML.  |



# Thao tác trên thuộc tính HTML trong jq

| Phương thức | Mô tả   |
|-------------|---|
| attr()      | Get hoặc set giá trị của thuộc tính được chỉ định của (các) phần tử đích. |
| prop()      | Get hoặc set giá trị của thuộc tính được chỉ định của (các) phần tử đích. |
| html()      | Get hoặc set html nội dung HTML cho (các) phần tử đích đã chỉ định.       |
| text()      | Get hoặc set văn bản (text) cho (các) phần tử đích đã chỉ định.           |
| val()       | Get hoặc set thuộc tính giá trị của phần tử đích đã chỉ định.             |

# Thao tác trên thuộc tính HTML trong jq

```
$('#myDiv').attr('class')  $('#myDiv').prop('class')  
  
<div id="myDiv" class="divCls">  
  <p style="background-color:yellow;width:100%">  
    This is paragraph.  
  </p>  
</div>  
  
<div id="firstNameDiv">  
  <label>First Name</label><input type="text" value="John" />  
</div>  
  
<input type="button" value="Get Value" id="addBtn" style="width:100px" />
```

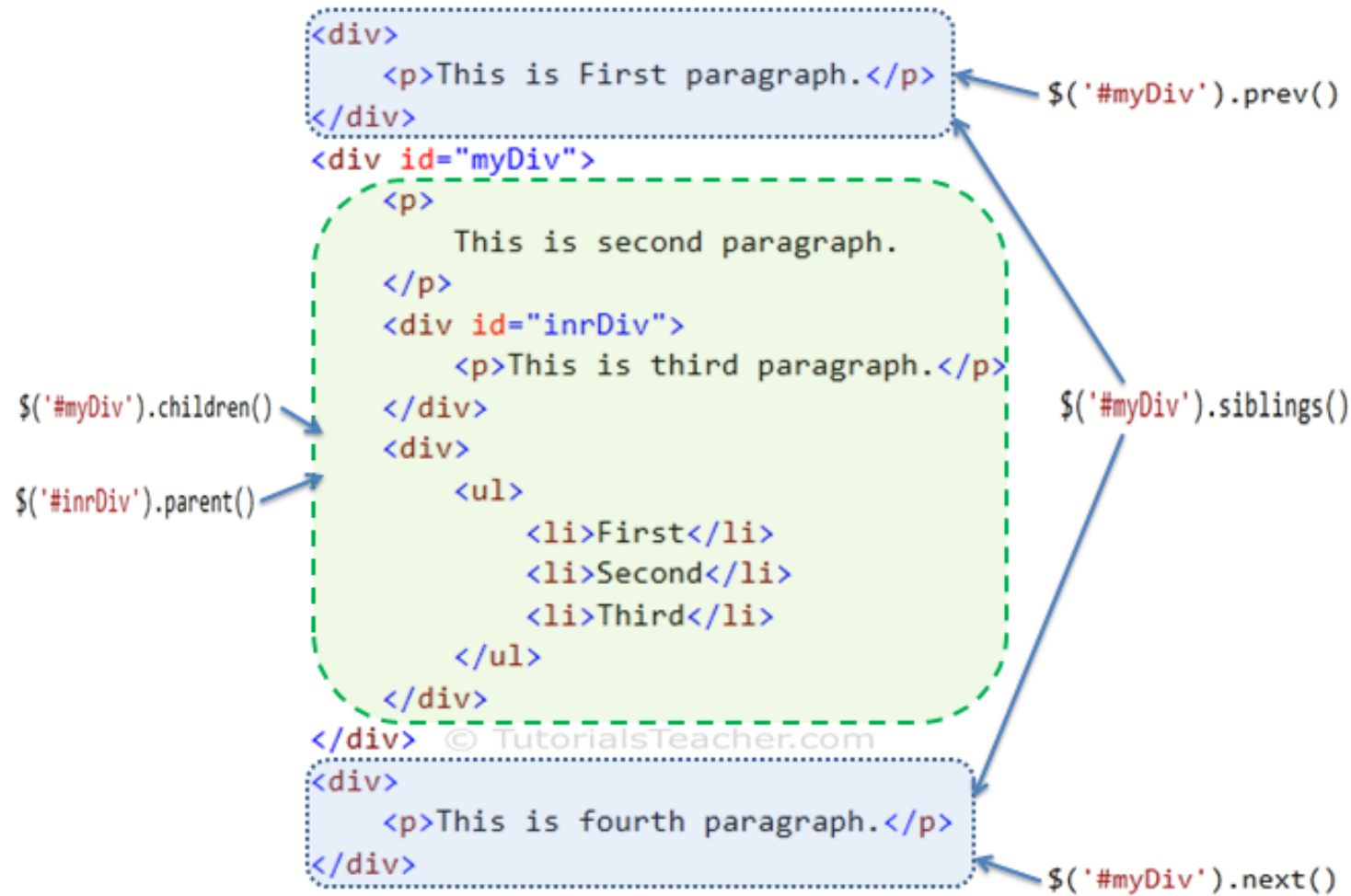
Arrows indicate the following jQuery methods and their targets:

- `$('#myDiv').attr('class')` points to the `class="divCls"` attribute in the first `<div>` tag.
- `$('#myDiv').prop('class')` points to the `class="divCls"` attribute in the first `<div>` tag.
- `$('#myDiv').html()` points to the entire content of the first `<div>` (the `<p>` tag and its text).
- `$('#myDiv').text()` points to the text content of the first `<div>` ("This is paragraph.").
- `$('#input:text').val()` points to the `value="John"` attribute of the text input.
- `$('#label').text()` points to the text content of the label "First Name".
- `$('#input:button').val()` points to the `value="Get Value"` attribute of the button.
- `$('#input:button').prop('style').width` points to the `width:100px` style attribute of the button.

# Duyệt HTML DOM với jq

| Phương thức | Mô tả  |
|-------------|--|
| add()       | Thêm các đối tượng vào một tập hợp   |
| children()  | Trả về tất cả các con trực tiếp của đối tượng đang chọn                      |
| closest()   | Trả về tổ tiên đầu tiên của đối tượng đang chọn                              |
| each()      | Thực thi một hàm với mỗi đối tượng được tìm thấy                             |
| find()      | Trả về các đối tượng con cháu của đối tượng đang chọn.                       |
| first()     | Trả về đối tượng đầu tiên của các đối tượng đang chọn.                       |
| has()       | Trả về các đối tượng có chứa một hoặc nhiều đối tượng khác.                  |
| last()      | Trả về đối tượng cuối cùng của các đối tượng đang chọn.                      |
| next()      | Trả về đối tượng là anh chị em ruột kế tiếp của đối tượng đang chọn.         |
| nextAll()   | Trả về tất cả đối tượng là anh chị em ruột kế tiếp của đối tượng đang chọn.  |
| parent()    | Trả về đối tượng cha trực tiếp của đối tượng đang chọn                       |
| parents()   | Trả về tất cả các đối tượng là tổ tiên của đối tượng đang chọn.              |
| prev()      | Trả về đối tượng là anh chị em ruột trước đó của đối tượng đang chọn.        |
| prevAll()   | Trả về tất cả đối tượng là anh chị em ruột trước đó của đối tượng đang chọn. |
| siblings()  | Trả về tất cả đối tượng là anh chị em ruột của đối tượng đang chọn.          |

# Duyệt HTML DOM với jq

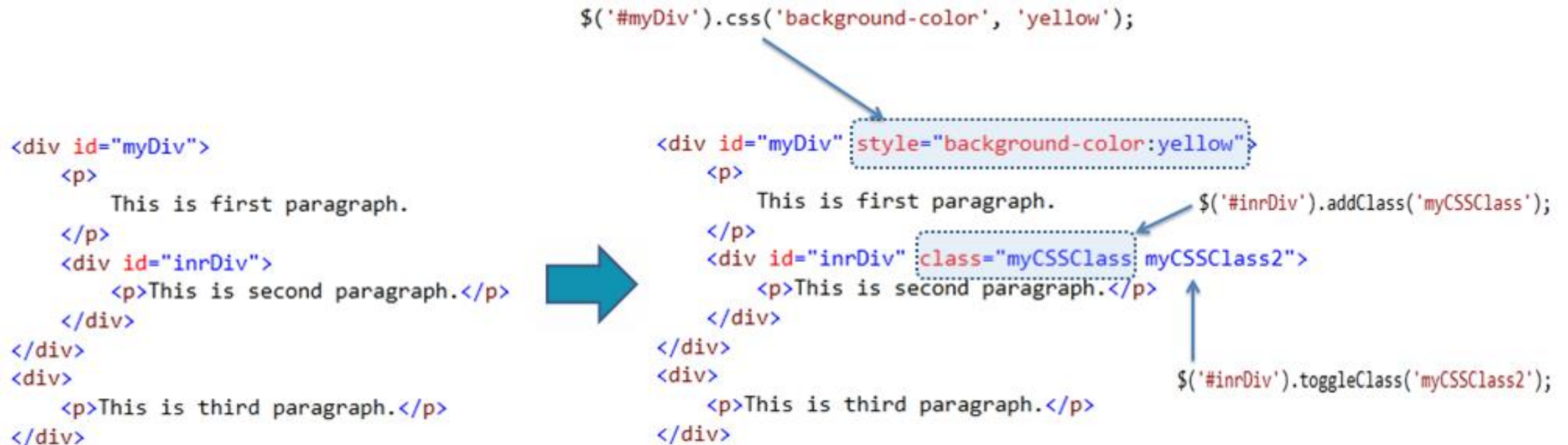


# Thao tác với CSS sử dụng jQuery

| Phương thức   | Mô tả  |
|---------------|--|
| css()         | Lấy hoặc gán một hoặc nhiều thuộc tính style của các đối tượng html  |
| addClass()    | Thêm một hoặc nhiều class vào các đối tượng html                     |
| removeClass() | Loại bỏ một hoặc nhiều class khỏi các đối tượng html                 |
| toggleClass() | Chuyển đổi giữa add và remove một hoặc nhiều class từ đối tượng html |
| hasClass()    | Kiểm tra một đối tượng lựa chọn có tên class xác định không.         |



# Thao tác với CSS sử dụng jQ





# CSS sử dụng jQ

```
<ul>
  <li>Pine nuts</li>
  <li>Honey</li>
  <li>Vines</li>
</ul>
```

```
<script>
  $(document).ready(function(){
    var bgColor = $('li').css('background-color');
    $('ul').append('<p> Color was: '+bgColor+'</p>');
    $('li').css({
      'background-color': '#c5a',
      'border': '1px solid #fff',
      'color': '#000',
      'font-family': 'Georgia',
      'padding-left': '2rem'
    });
  });
</script>
```

```
<style>
  ul{
    list-style-type: none;
    margin: 0;
    padding: 0;
  }
  ul li{
    background-color: coral;
  }
</style>
```

Pine nuts  
Honey  
Vines

Change  
CSS

Pine nuts  
Honey  
Vines

Color was:rgb(255, 127, 80)

| Phương thức               | Mô tả hoạt động  |
|---------------------------|--|
| <code>animate()</code>    | Thực thi một hiệu ứng tùy chỉnh trên đối tượng.  |
| <code>clearQueue()</code> | Xóa tất cả các hàm còn lại trong hàng đợi xử lý của đối tượng được chọn.               |
| <code>delay()</code>      | Thiết lập độ trễ cho tất cả các hàm trong hàng đợi của các đối tượng được chọn.        |
| <code>fadeIn()</code>     | Đối tượng được chọn sẽ xuất hiện với hiệu ứng từ mờ dần tới rõ nét.                    |
| <code>fadeOut()</code>    | Đối tượng được chọn sẽ biến mất với hiệu ứng mờ dần.                                   |
| <code>fadeTo()</code>     | Phương thức làm mờ dần đối tượng được chọn với thời gian, độ mờ xác định               |
| <code>fadeToggle()</code> | Phương thức chuyển đổi giữa 2 hiệu ứng <code>fadeIn()</code> và <code>fadeOut()</code> |
|                           |  |

| Phương thức   | Mô tả hoạt động  |
|---------------|--|
| hide()        | Hiệu ứng ẩn các đối tượng được lựa chọn                            |
| show()        | Hiệu ứng hiển thị các đối tượng được lựa chọn (ngược lại với hide) |
| slideUp()     | Hiệu ứng trượt lên (ẩn) các đối tượng đi.                          |
| slideDown()   | Hiệu ứng trượt xuống (hiển thị) các đối tượng lên.                 |
| slideToggle() | Phương thức chuyển đổi giữa 2 hiệu ứng slideUp() và slideDown()    |

## jQuery **hide()** → Syntax:

```
$(selector).hide();  
$(selector).hide(speed, callback);  
$(selector).hide(speed, easing, callback);
```

## jQuery **toggle()** → Syntax:

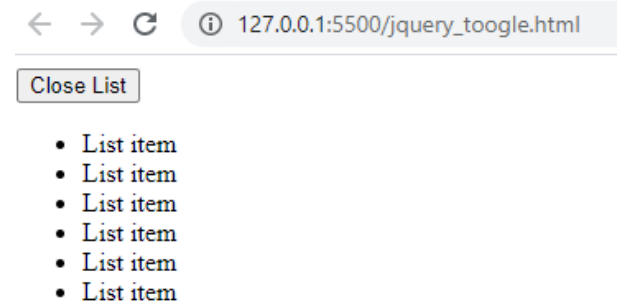
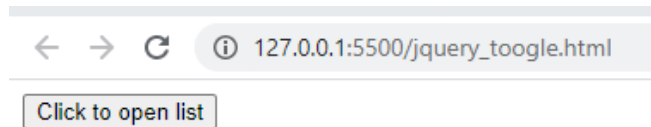
```
$(selector).toggle();  
$(selector).toggle(speed, callback);  
$(selector).toggle(speed, easing, callback);  
$(selector).toggle(display);
```

## Tham khảo thêm:

[https://www.w3schools.com/jquery/jquery\\_animate.asp](https://www.w3schools.com/jquery/jquery_animate.asp)

<https://www.javatpoint.com/jquery-tutorial>

# jQuery Animation – hide, toggle

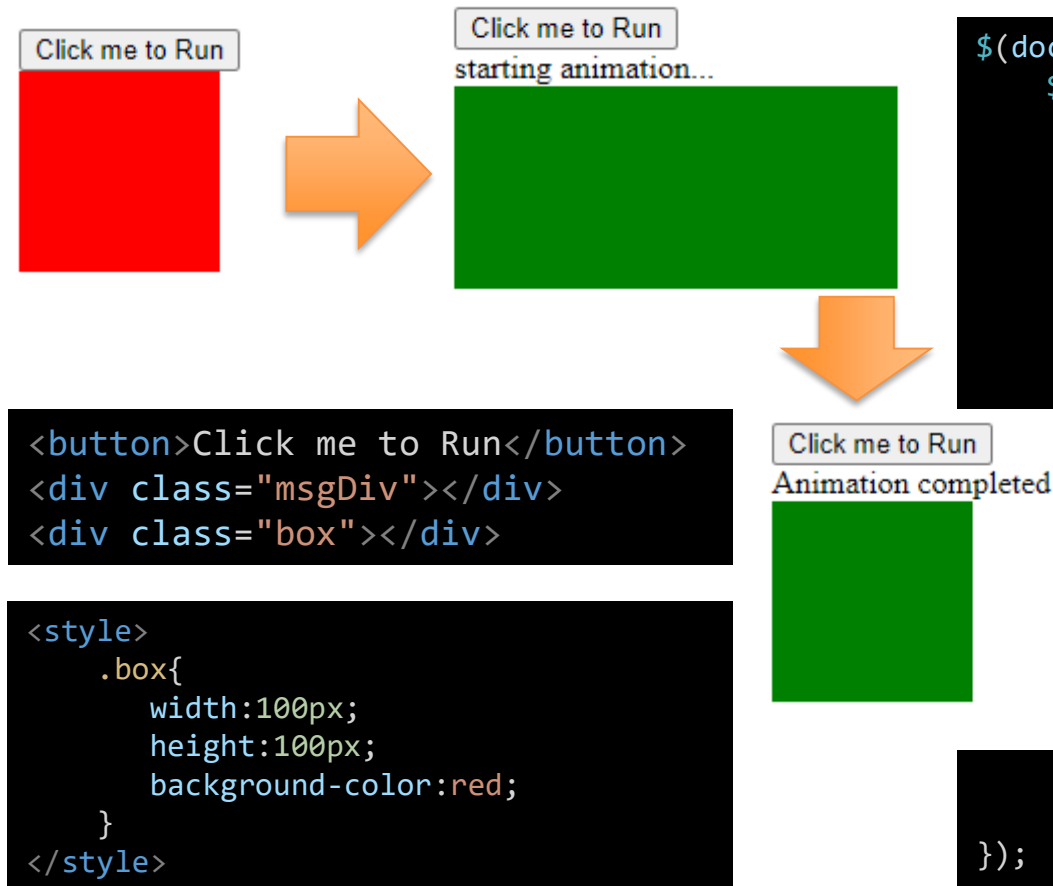


```
<button>
  <span> Click to open list </span>
  <span style="display:none"> Close List </span>
</button>
<ul id="list">
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
  <li>List item</li>
</ul>
```

```
<script>
  $(document).ready(function () {
    $('#list').hide();
    //toggle button text
    $('button').click(function () {
      $('span', this).toggle();
    });
    //toggle list
    $('button').click(function () {
      $('#list').toggle('slow');
    });
  });
</script>
```

# jQuery Animation - **animate**

`$(selector).animate({ properties }, duration, callback);`



```
$(document).ready(function(){
  $("button").click(function(){
    //Call function runAnimation
    runAnimation();
    //function run animation
    function runAnimation(){
      $('.box').animate({height:300},"slow",startAnimation);
      $('.box').animate({width:300},"slow");
      $('.box').css("background-color","green");
      $('.box').animate({height:100},"slow");
      $('.box').animate({width:100},"slow",callBackAnimation);
    }
    //function startAnimation
    function startAnimation(){
      $('.msgDiv').text('starting animation...');
    }
    //function callBackAnimation
    function callBackAnimation(){
      $('.msgDiv').text('Animation completed');
    }
  });
});
```

# Code gợi ý khác

```
<script>
$(document).ready(function(){
    $("button").click(function(){
        $('.box').animate(
            {height:'300px'},
            { // options parameter
              duration: 1000,
              complete: function () {
                $(this).animate({
                  width: '100px',
                  height: '100px'
                }, 1000,
                function () {
                  $('.msgDiv').text('Animation completed..');
                });
              },
            },
            start: function () {
              $(this).animate({
                width: '300px',
                height: '100px'
              }, 1000);
              $('.msgDiv').text('Starting animation..');
              $('.box').css("background-color","green");
            }
        );
    });
});
</script>
```

Click me to Run



Click me to Run

Starting animation..



Click me to Run

Animation completed..



# HTML5 Progress Bar

```
Simple Progress Bar:<progress value="0.4"></progress>
<br>
Animated Progress Bar:
<progress id="progressbar" value="0" max="100"></progress>
<div class="progress-value"></div>
```

Simple Progress Bar:



Animated Progress Bar:



25%



Simple Progress Bar:



Animated Progress Bar:



61%



Simple Progress Bar:



Animated Progress Bar:



100%

```
/* setting the dimensions */
progress {
    width: 400px;
    height: 24px;
    display: block;
    /* disable defaults */
    -webkit-appearance: none;
    border: none;
}

/*Styling the background of the progress bar */
progress::-webkit-progress-bar {
    background: white;
    border-radius: 45px;
    padding: 2px;
    box-shadow: 0 1px 0px 0 rgba(255, 255, 255, 0.2);
}

/* Setting the appearance of the meter */
progress::-webkit-progress-value {
    border-radius: 25px;
    box-shadow: inset 0 1px 1px 0 rgba(255, 255, 255, 0.4);
    background-size: 250px 25px;
    background-image: linear-
gradient(135deg, rgba(255, 255, 255, .15) 25%,
            transparent 25%, transparent 75%, rgba(255, 255, 255, .15) 75%,
            rgba(255, 255, 255, .15) 75%,transparent 75%, transparent);
}
```

# HTML5 Progress Bar (tt)

```
$(document).ready(function() {  
    var progressbar = $('#progressbar'),  
    max = progressbar.attr('max'),  
    time = (500/max)*5,  
    value = progressbar.val();  
  
    function loading() {  
        value += 1;  
        addValue = progressbar.val(value);  
        $('.progress-value').html(value + '%');  
        if (value == max) {  
            clearInterval(animate);  
        }  
    };  
    var animate = setInterval(function() {  
        loading();  
    }, time);  
});
```

Simple Progress Bar:



Animated Progress Bar:



25%



Simple Progress Bar:



Animated Progress Bar:



61%



Simple Progress Bar:



Animated Progress Bar:



100%



# jQuery Form

```
<form action="">
  First name: <input type="text" name="FirstName" placeholder="Enter Your First Name"><br>
  Last name: <input type="text" name="LastName" placeholder="Enter Your Last Name"><br>
</form>
<button>Serialize form values</button>
<div class="result"></div>
```

First name:   
Last name:

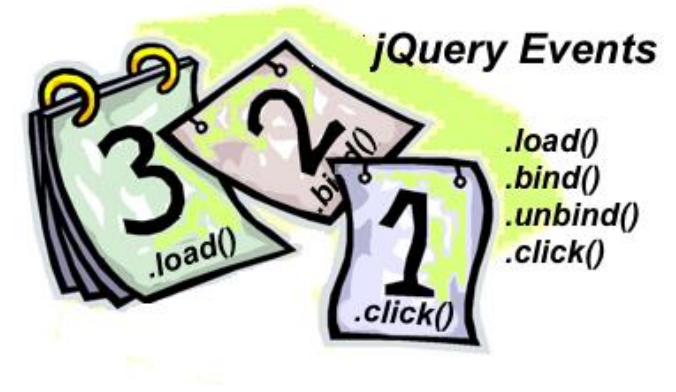


First name:   
Last name:

FirstName=FPT&LastName=Software

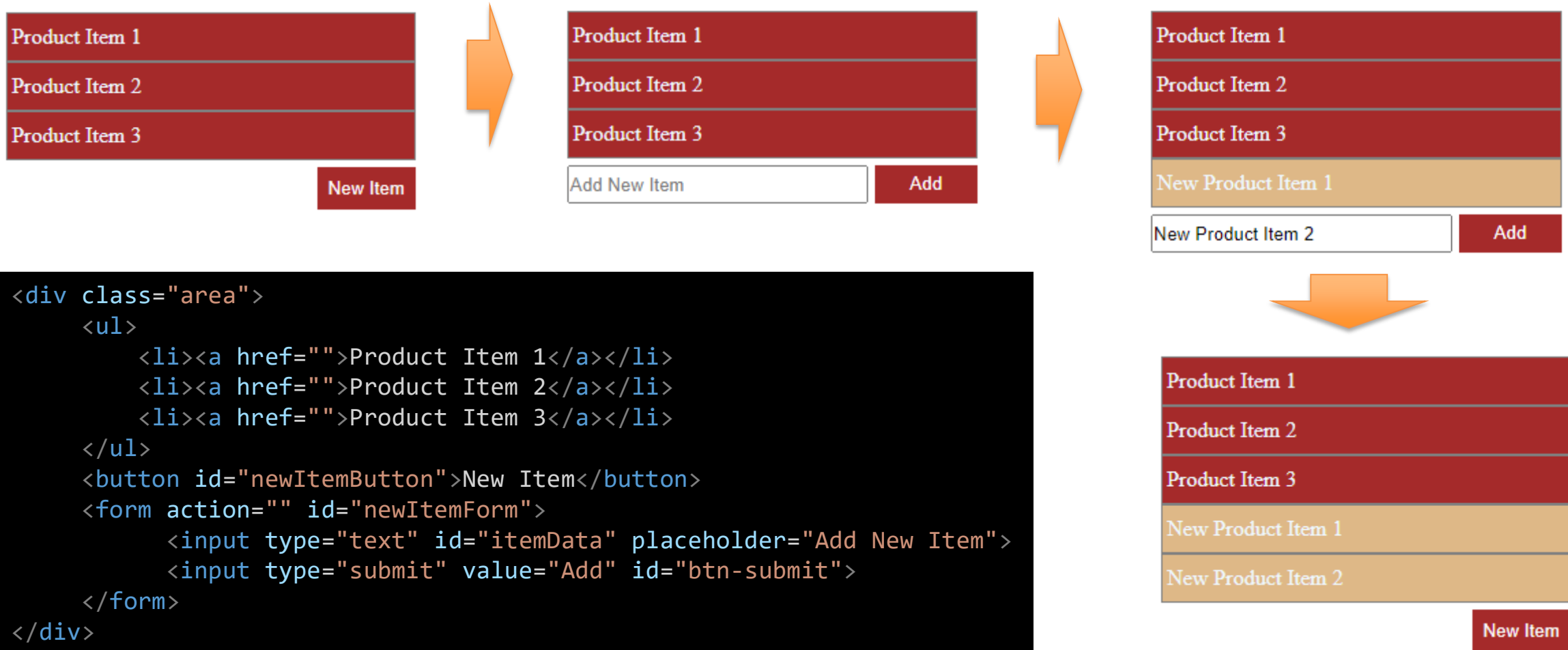
```
<script>
  $(document).ready(function(){
    $("button").click(function(){
      $(".result").text($("#form").serialize());
    });
  });
</script>
```

- Một số DOM Event thông dụng



| Mouse Events | Keyboard Events | Form Events | Document/Window Events |
|--------------|-----------------|-------------|------------------------|
| click        | keypress        | submit      | load                   |
| dblclick     | keydown         | change      | resize                 |
| mouseenter   | keyup           | focus       | scroll                 |
| mouseleave   |                 | blur        | unload                 |

# Ví dụ



```
.area{
    margin: 0 auto;
    width: 300px;
    position: relative;
}
.area ul{
    margin: 0;
    list-style-type: none;
    padding: 0;
}
.area ul li{
    padding: 0.5rem 0.2rem;
    background-
color: brown;
    border: 1px solid gray;
}
.area ul li a{
    display: block;
    text-decoration: none;
    color: aliceblue;
}
```

```
#newItemButton{
    position: absolute;
    right: 0;
    margin: 0.3rem 0;
    padding: 0.5rem;
    background-color: brown;
    color: white;
    border: none;
}
#btn-submit{
    float: left;
    width: 25%;
    padding: 0.4rem 0;
    background-color: brown;
    color: white;
    border: none;
}
#itemData{
    float: left;
    width: 72%;
    margin-right: 0.3rem;
    padding: 0.3rem 0;
}
```

```
#newItemForm{
    position: absolute;
    left: 0;
    width: 100%;
    padding: 0.3rem 0rem;
}
li.newItem{
    background-
color: burlywood !important;
}
```

# jQuery Code gợi ý

```
$(document).ready(function(){
    var newItemButton = $('#newItemButton');
    var newItemForm = $('#newItemForm');
    var textInput = $('#itemData');
    newItemButton.show();
    newItemForm.hide();
    //Show Form when clicked on New Item button
    $('#newItemButton').on('click', function(){
        newItemButton.hide();
        newItemForm.show();
    });
    //Add new Item when click on Submit button
    $('#newItemForm').on('submit',function(e){
        e.preventDefault();
        var newText = $('#itemData').val();
        $('ul li:last').after('<li class="newItem"><a href="">'+newText+'</a></li>');
        newItemForm.hide();
        newItemButton.show();
        textInput.val('');
    });
});
```

# Front-end Essentials

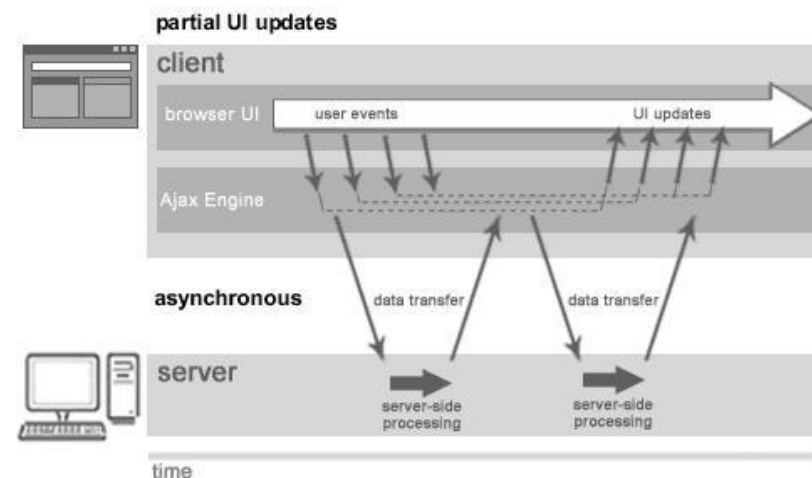
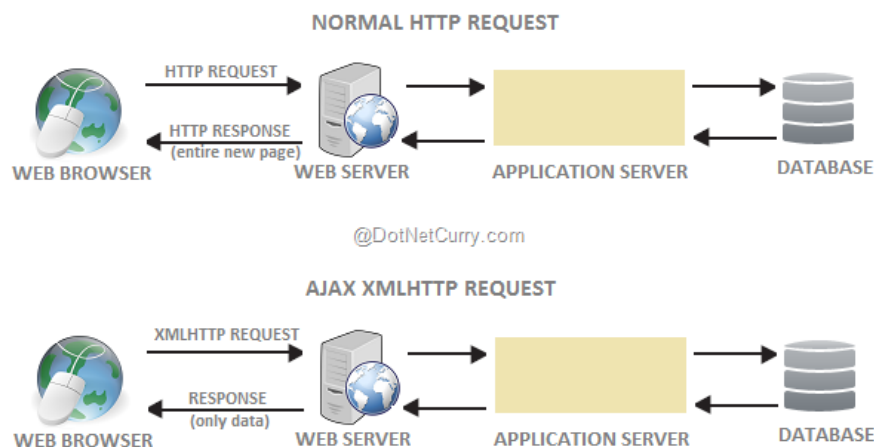
## *jQuery & Ajax*



- AJAX là chữ viết tắt của Asynchronous JavaScript and XML
    - ✓ Asynchronous (Async – bất đồng bộ): Một chương trình có thể xử lý không theo tuần tự các hàm, không có quy trình, có thể nhảy đi bỏ qua bước nào đó
    - ✓ XML (eXtensible Markup Language): được thiết kế để chứa dữ liệu theo cấu trúc đặc thù để truyền đi trong môi trường mạng
  - Cả JavaScript và XML đều hoạt động bất đồng bộ trong AJAX.
- Ứng dụng web có thể sử dụng AJAX để gửi và nhận data từ server mà không phải toàn bộ trang.

# Web sử dụng **AJAX** có gì khác so với Web thông thường

| Thông thường   | Có AJAX  |
|--|--|
| <ol style="list-style-type: none"><li>1. HTTP được gửi từ trình duyệt lên máy chủ.</li><li>2. Máy chủ nhận, sau đó phản truy xuất thông tin.</li><li>3. Server gửi dữ liệu được yêu cầu lại cho trình duyệt.</li><li>4. Trình duyệt nhận dữ liệu và tải lại trang để hiển thị dữ liệu lên.</li></ol> | <ol style="list-style-type: none"><li>1. Trình duyệt tạo một lệnh gọi JavaScript để kích hoạt XMLHttpRequest.</li><li>2. Ở trình duyệt tạo một yêu cầu HTTP gửi lên server.</li><li>3. Server tiếp nhận, truy xuất và gửi lại dữ liệu cho trình duyệt.</li><li>4. Trình duyệt nhận dữ liệu từ server và ngay lập tức hiển thị lên trang. Không cần tải lại toàn bộ trang</li></ol> |





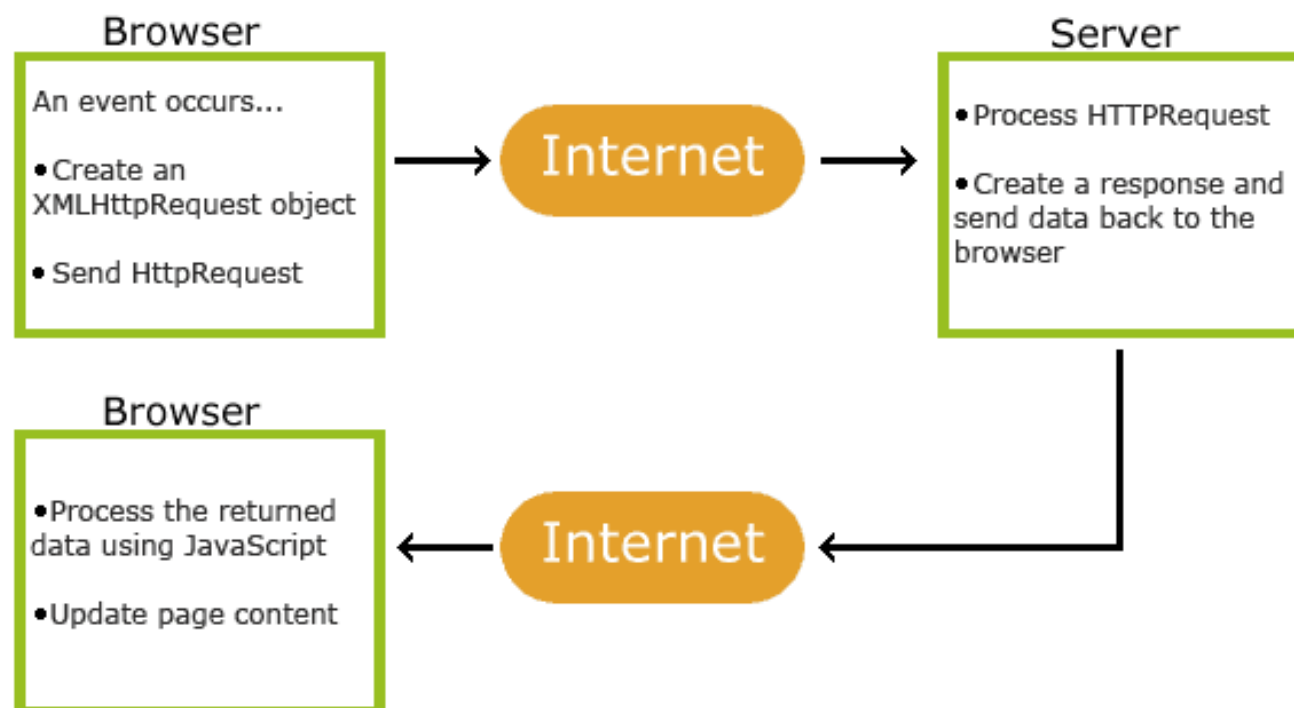
- Ajax là một công nghệ tích hợp bởi các công nghệ sau đây:
  - ✓ XML và XSLT
  - ✓ HTML hoặc XHTML
  - ✓ CSS
  - ✓ HTML DOM
  - ✓ JavaScript
  - ✓ Đối tượng XMLHttpRequest

## AJAX WEB APPLICATION DEVELOPMENT



# Ajax hoạt động như thế nào

## How AJAX Works



- ❖ Cập nhật trang web mà không cần tải lại trang
- ❖ Yêu cầu dữ liệu từ máy chủ - sau khi trang tải xong
- ❖ Nhận dữ liệu từ máy chủ - sau khi trang tải xong
- ❖ Gửi dữ liệu đến máy chủ - ngầm (trong suốt với người dùng)

- Tất cả các trình duyệt hiện đại đều hỗ trợ đối tượng XMLHttpRequest.
- Các trạng thái của XMLHttpRequest

`XMLHttpRequest.readyState`

Read only

Returns an **unsigned short**, the state of the request:

| Value | State            | Description   |
|-------|------------------|---|
| 0     | UNSENT           | <code>open ()</code> has not been called yet.                               |
| 1     | OPENED           | <code>send ()</code> has been called.                                       |
| 2     | HEADERS_RECEIVED | <code>send ()</code> has been called, and headers and status are available. |
| 3     | LOADING          | Downloading; <code>responseText</code> holds partial data.                  |
| 4     | DONE             | The operation is complete.  |

## ■ Lấy dữ liệu từ file .txt

```
<div id="demo">
  <h2>The XMLHttpRequest Object</h2>
  <button type="button" onclick="loadDoc()">Change Content</button>
</div>

<script>
function loadDoc() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("demo").innerHTML =
        this.responseText;
    }
  };
  xhttp.open("GET", "ajax_info.txt", true);
  xhttp.send();
}
</script>
</div>
```

## ▪ Load dữ liệu từ file xml

```
<h2>The XMLHttpRequest Object</h2>
<h2>Retrieve data from XML file</h2>
<p><b>Status:</b> <span id="A1"></span></p>
<p><b>Status text:</b> <span id="A2"></span></p>
<p><b>Response:</b> <span id="A3"></span></p>
<button onclick="loadDoc('note.xml')">Get XML data</button>
<script>
    function loadDoc(url) {
        var xhttp = new XMLHttpRequest();
        xhttp.onreadystatechange = function() {
            if (this.readyState == 4 && this.status == 200) {
                document.getElementById('A1').innerHTML = this.status;
                document.getElementById('A2').innerHTML = this.statusText;
                document.getElementById('A3').innerHTML = this.responseText;
            }
        };
        xhttp.open("GET", url, true);
        xhttp.send();
    }
</script>
```

note.xml

```
<note>
    <to>Tove</to>
    <from>Jani</from>
    <heading>Reminder</heading>
    <body>Don't forget me this
weekend!</body>
</note>
```

# jQuery Ajax

`jQuery.get( url, [data], [callback],  
[type] )`

`jQuerygetJSON( url, [data],  
[callback] )`

`jQuery.getScript( url, [callback] )`

`jQuery.post( url, [data], [callback],  
[type] )`

`load( url, data, callback):`

`serialize( )`



`serializeArray(`

`ajaxComplete( callback )`

`ajaxStart( callback )`

`ajaxError( callback)`

`ajaxSend( callback)`

`ajaxStop( callback)`

# jQuery Ajax syntax

```
$.ajax({  
    type: "method_request", //Get, Post, put  
    url: "page_url",  
    data: {  
        "field1": "value-1",  
        "field2": "value-2",  
    },  
    dataType: "data_type", //HTML, json, xml  
    success: function(data, textStatus, jqXHR) { },  
    error: function(xhr) { },  
    statusCode: {  
        404: function() {  
            alert("page not found");  
        }  
    },  
    beforeSend: function(){ },  
    complete: function(){ } // to be called when the request finishes  
});
```

Good Job



# Thank you

