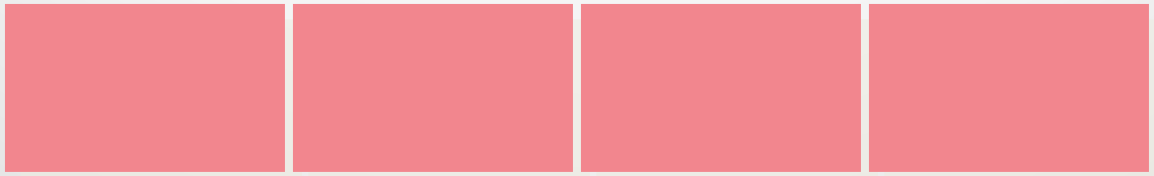
Bài giảng

PHÁT TRIỂN ỨNG DỤNG WEB

Lê Đình Thanh
Khoa Công nghệ Thông tin
Trường Đại học Công nghệ, ĐHQGHN
E-mail: thanhld@vnu.edu.vn Mobile: 0987.257.504

Chương 4

Quản lý trang web bằng JavaScript (tiếp)



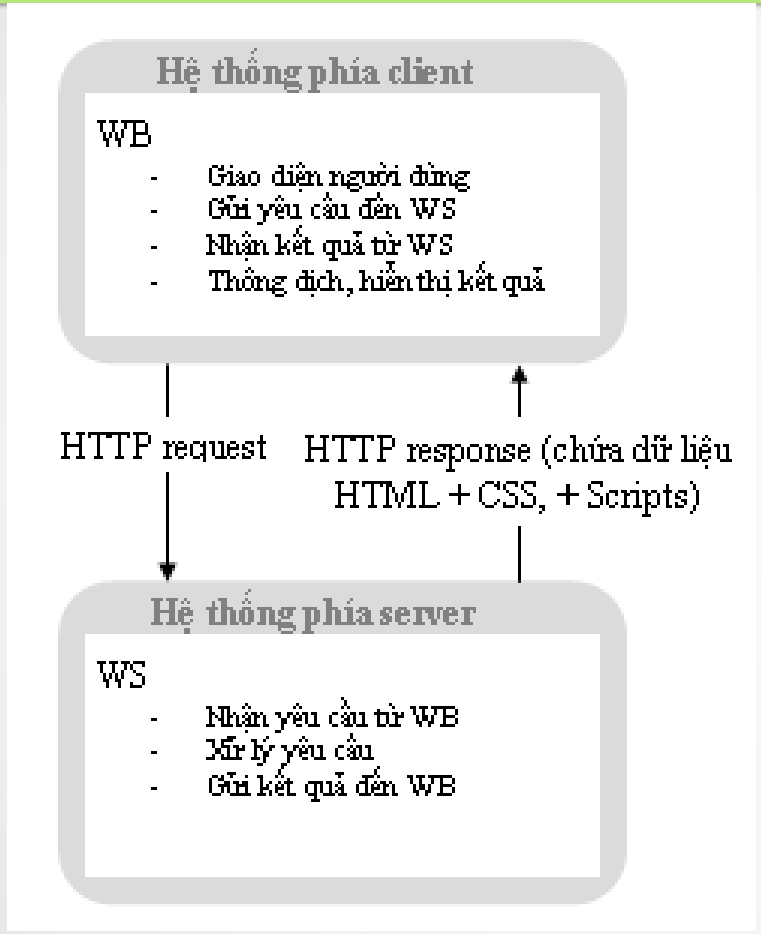
Nội dung

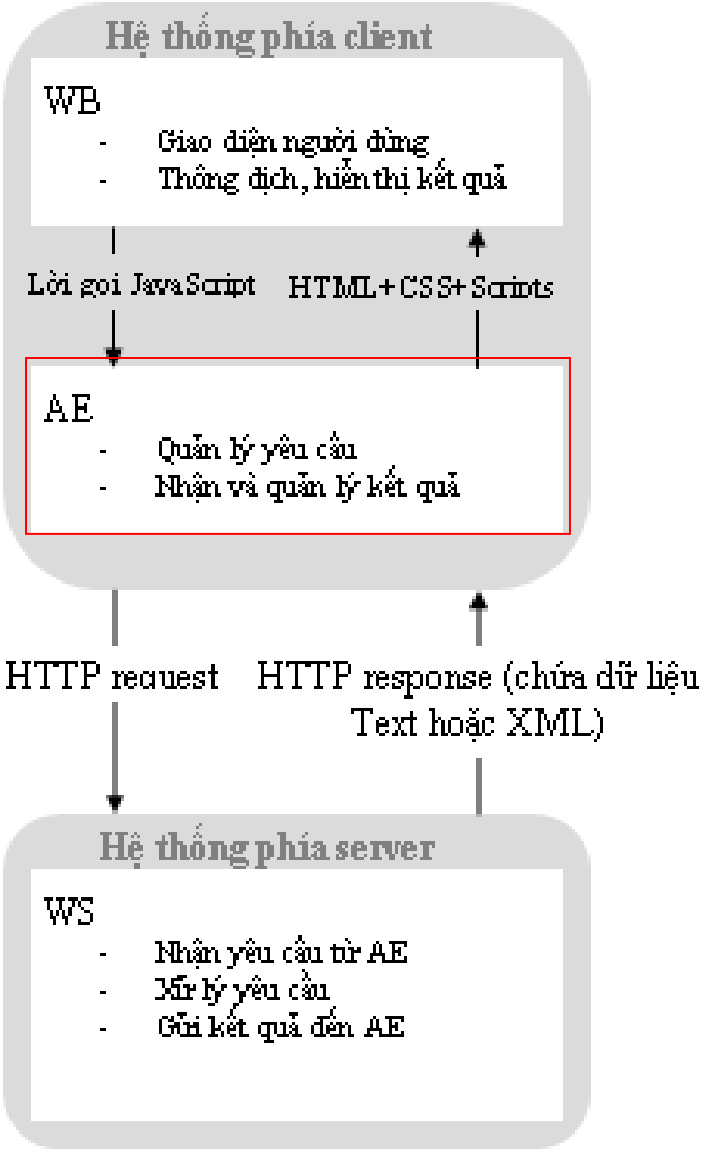
- AJAX
- JSON

AJAX là gì?

- AJAX (Asynchronous Javascripts and XML) là một kỹ thuật kết hợp một số công nghệ web để xây dựng các ứng dụng web mà tương tác giữa người dùng với ứng dụng được thực hiện không đồng bộ. Các công nghệ bao gồm:
  - Hiển thị dựa trên chuẩn sử dụng HTML và CSS
  - Tương tác động sử dụng DOM
  - Trao đổi và xử lý dữ liệu sử dụng XML, text
  - Thu nhận dữ liệu không đồng bộ sử dụng XMLHttpRequest
  - Kết hợp các công nghệ sử dụng JavaScript

Web truyền thống <> Ajax Web

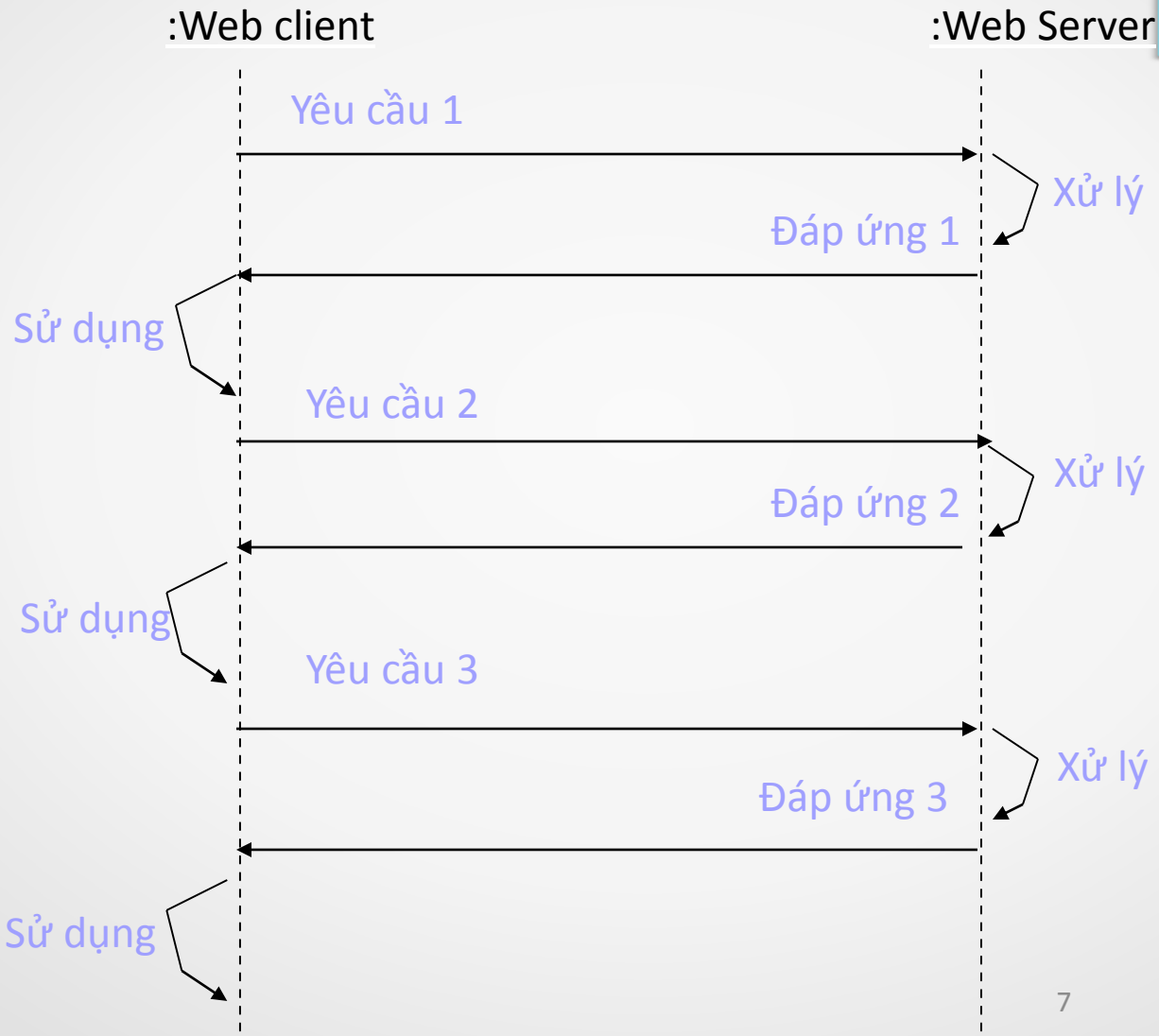




Web truyền thống

- Yêu cầu của người dùng được gửi trực tiếp từ browser đến Web server thông qua HTTP request
- Khi nhận được HTTP request, Web server xử lý yêu cầu, sinh ra trang HTML mới, rồi gửi toàn bộ trang HTML (chứa HTML và CSS) mới đến browser. Browser xóa trang cũ và hiển thị trang mới.
- Từ khi gửi yêu cầu đi, người dùng *không được làm thêm bất kỳ thao tác gì* cho đến khi trang HTML mới được gửi đến client: mỗi yêu cầu phải được giải quyết dứt điểm trước khi có yêu cầu tiếp theo = đồng bộ.

Hoạt động của web truyền thống



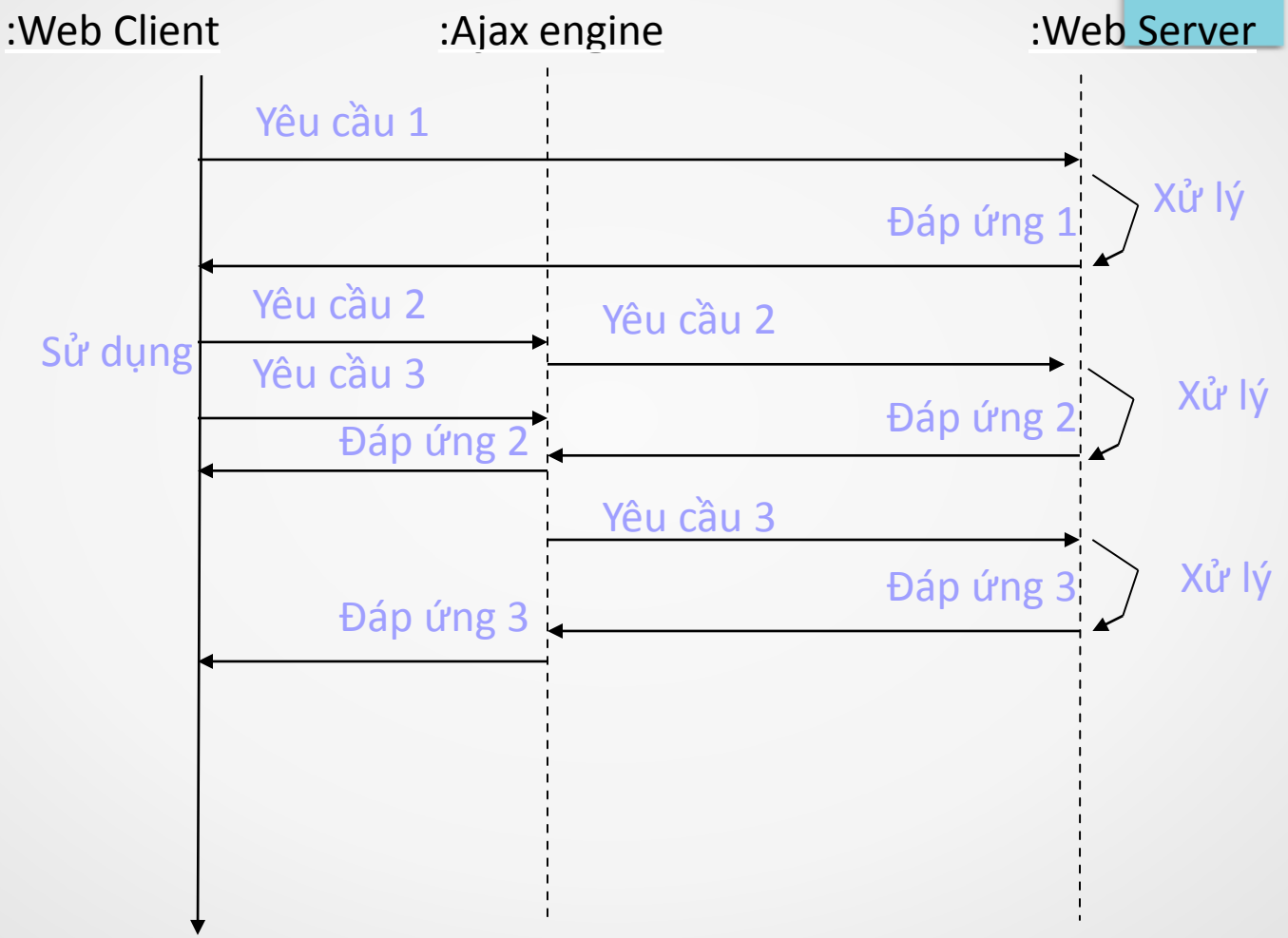
Web truyền thống: Hạn chế

- Khi người dùng thao tác thì server "nghỉ" và ngược lại
  - Lãng phí thời gian, hiệu quả sử dụng thấp
  - Người dùng phải vừa làm vừa đợi: gửi yêu cầu $\rightarrow$ đợi $\rightarrow$ nhận kết quả $\rightarrow$ gửi yêu cầu $\rightarrow$ đợi $\rightarrow$ ... $\Rightarrow$ Người dùng phải đợi lâu nếu yêu cầu xử lý lớn và server mất nhiều thời gian xử lý + Hiển thị không liên tục, "nhấp nháy" gây khó chịu (! HCI).
- Toàn bộ trang HTML mới được gửi từ server đến client
  - Không cần thiết vì có thể nhiều chi tiết trên trang mới vẫn như trang cũ
  - Lượng thông tin trao đổi giữa client-server lớn $\Rightarrow$ chi phí truyền thông (thời gian, băng thông) lớn.

Ajax Web

- Ajax engine được cài trên client, làm nhiệm vụ giao tiếp trung gian giữa browser với web server
  - Browser gửi yêu cầu đến Ajax engine bằng lời gọi Javascript.
  - Ajax engine chuyển yêu cầu của Client thành HTTP request và gửi cho web server
  - Web server xử lý yêu cầu rồi gửi kết quả cho Ajax engine ở dạng XML
  - Ajax engine biên dịch XML thành HTML và gửi HTML cho browser
- Một yêu cầu của người dùng chưa cần được giải quyết xong thì người dùng đã có thể đưa ra yêu cầu khác
  - Trao đổi giữa Browser với Ajax engine và giữa Ajax engine với Server để thực hiện các yêu cầu diễn ra không đồng bộ.

Hoạt động của Ajax web



Ajax Web: Ưu điểm

- Người dùng và server thực hiện một cách song hành
  - Không lãng phí thời gian, hiệu quả sử dụng cao
  - Người dùng không phải vừa làm vừa đợi
  - Hiển thị liên tục, không gây khó chịu (HCI).
- Chỉ phần khác biệt của trang mới so với trang cũ mới được gửi từ server đến client
  - Lượng thông tin trao đổi giữa client-server tối thiểu $\Rightarrow$ tiết kiệm chi phí (thời gian, băng thông) truyền thông.

Vì sao dùng Ajax

- Để tạo ra các ứng dụng web mà giao tiếp của nó với người dùng diễn ra như giao tiếp của ứng dụng Winform với người dùng: liên tục.
- hiệu quả trong sử dụng và trong truyền thông

Sử dụng AJAX

- Sử dụng Ajax Engine (đối tượng Javascript XMLHttpRequest) để gửi yêu cầu đến server và lấy dữ liệu về từ server.
- Sau khi XmlHttpRequest nhận được dữ liệu từ server, sử dụng javascript để sửa đổi trang web trên client với dữ liệu mới nhận được.

Lấy đối tượng XMLHttpRequest

```
function getXmlHttpObject()  {
     var xmlHttp = null;
 try    {
                  // Firefox, Opera 8.0+, Safari
                   xmlHttp = new XMLHttpRequest();
       }  catch (e) {
                 // Internet Explorer
                    try  {
                        xmlHttp = new ActiveXObject("Msxml2.XMLHTTP");
                         }  catch (e) {
                              try {
                                       xmlHttp=new
   ActiveXObject("Microsoft.XMLHTTP");
                                    } catch (e)  {
                                        alert("Trinh duyet khong ho tro
   AJAX!");
                                    }
                         }
       }
   return xmlHttp;
 }
```

XMLHttpRequest::readyState

```
if(xmlHttp.readyState==4)
    {
   // Đã nhận đủ trả lời từ
    server
   if (xmlHttp.status ==
    200) {
   //Đã thực hiện thành
    công trên server
   //Dùng javascript để sửa
    đổi trang
    }
}
```

ready State	Ý nghĩa
0	Chưa thiết lập yêu cầu
1	Đã thiết lập yêu cầu
2	Đã gửi yêu cầu
3	Đang trả lời
4	Đã trả lời xong

XMLHttpRequest.onreadystatechange

Là một con trỏ hàm không đổi, được kích hoạt mỗi khi thuộc tính readyState thay đổi.

```
xmlHttp.onreadystatechange = tenHamXuly;
function tenHamXuly() {}
```

Hoặc

```
xmlHttp.onreadystatechange = function() {
    //Noi dung xu ly
}
```

Gửi yêu cầu lên server theo GET

xmlHttp.open(“GET”, path?querystring, asynchronous);

xmlHttp.send(null);

Ví dụ:

```
xmlHttp.open("GET", "time.php?zone=7", true);
xmlHttp.send(null);
```

Gửi yêu cầu lên server theo POST

```
xmlHttp.open(“POST”, url, asynchronous);
xmlHttp.setRequestHeader("Content-Type",
    "application/x-www-form-urlencoded");
xmlHttp.send(params);
```

Ví dụ:

```
xmlHttp.open(“POST", " time.php", true);
xmlHttp.setRequestHeader("Content-Type",
    "application/x-www-form-urlencoded");
xmlHttp.send(“zone=7”);
```

XMLHttpRequest.responseText

Nội dung dạng text/html do server gửi về.

Muốn sử dụng thuộc tính này, server phải thiết lập thuộc tính ContentType của trang là text/html (mặc định)

Server gửi dữ liệu dạng text

```
$time = 100;
echo $time;
```

Trình duyệt nhận và xử lý dữ liệu dạng text

```
xmlHttp.onreadystatechange = function() {
if(xmlHttp.readyState==4 &&
   xmlHttp.status==200) {
   document.write(xmlHttp.responseText);
 }
}
```

XMLHttpRequest. responseXML

Nội dung dạng XML do server gửi về.

Muốn sử dụng thuộc tính này, server phải thiết lập thuộc tính ContentType của trang là text/xml

Server gửi dữ liệu dạng XML

```
echo "<?xml version='1.0' encoding='UTF-8'?>“;
echo "<company>“;
echo "<compname>$companyname</compname>“;
echo "</company>“;
```

Trình duyệt nhận và xử lý XML

```
xmlHttp.onreadystatechange = function() {
  if (xmlHttp.readyState == 4 && xmlHttp.status=200)    {
    var xmlDoc=xmlHttp.responseXML.documentElement;
    document.getElementById("companyname").innerHTML
    =
    xmlDoc.getElementsByTagName("compname")[0].childNodes[0].no
    deValue;
  }
}
```

Ví dụ hoàn chỉnh

```
<!DOCTYPE html><html><head>
        <title>AJAX</title>
        <meta charset="utf-8">
</head><body>
        <div id="div1"></div>
        <script type="text/javascript" src=”xmlhttp.js”></script>
        <script type="text/javascript">
                var xmlhttp = getXmlHttpObject();
                xmlhttp.onreadystatechange = function() {
                        if (this.readyState == 4) {
                                if (this.status == 200) {
        document.getElementById("div1").innerHTML = this.responseText;
                                }
                        }
                };
                xmlhttp.open("GET", "webpart.htm", true);
                xmlhttp.send(null);
        </script>
 </body></html>
```

JSON

JavaScript Object Notaion

Giới thiệu JSON

- Cú pháp tương tự với Javascript, có thể chuyển đổi đối tượng Javascript thành JSON và ngược lại một cách đơn giản
- Được sử dụng rộng rãi trong lưu trữ cũng như trao đổi dữ liệu

Dữ liệu JSON

- Dữ liệu được biểu diễn bằng các giá-trị.
  - số, xâu, null, boolean, mảng và đối tượng
  - không có ngày tháng, hàm, undefined.
- Mảng
  - là danh sách các giá-trị được phân cách bởi dấu phẩy và đặt giữa các dấu mở và đóng ngoặc vuông ([và]).
- Đối tượng
  - có cú pháp tựa đối tượng Javascript
  - tên thuộc tính phải được đặt giữa hai dấu nháy kép.

Ví dụ: Dữ liệu JSON

```
[
        {
            "name":"John", "age":30,
            "cars": [
                    { "name":"Ford", "models":[ "Fiesta", "Focus", "Mustang" ] },
                    { "name":"BMW", "models":[ "320", "X3" ] }
            ]
        }, {
            "name":"Maria", "age":25,
            "cars": [
                    { "name":"Fiat", "models":[ "500", "Panda" ] }
            ]
        }, {
            "name":"David", ”age":40,
            "cars": [
                    { "name":"Ford", "models":[ "Fiesta", "Focus", "Mustang" ] },
                    { "name":"BMW", "models":[ "320", "X3", "X5" ] },
                    { "name":"Fiat", "models":[ "500", "Panda" ] }
            ]
        }
]
```

Chuyển đổi dữ liệu JSON và đối tượng Javascript

- Chuyển dữ liệu JSON thành đối tượng Javascript

  var obj = JSON.parse(jsonData);

- Chuyển đổi đối tượng Javascript thành dữ liệu JSON

  var s = JSON.stringify(obj);

Chuyển dữ liệu JSON thành đối tượng Javascript

```
var obj = JSON.parse(xhr.responseText);
for (var i = 0; i < obj.length; i++) {
        var r = document.createElement("tr");
        var c1 = document.createElement("td");
        var c2 = document.createElement("td");
        var c3 = document.createElement("td");
        c1.innerHTML = obj[i].name;
        c2.innerHTML = obj[i].age;
        c3.innerHTML = obj[i].cars.length;
        for (var j = 0; j < obj[i].cars.length; j++)
                c3.innerHTML += "<br>" +
                        obj[i].cars[j].name + " - " +
                        obj[i].cars[j].models;
        r.appendChild(c1);
        r.appendChild(c2);
        r.appendChild(c3);

        document.querySelector("#tbl1 tbody").appendChild(r);
}
```

Tiếp theo

JavaScript không đồng bộ