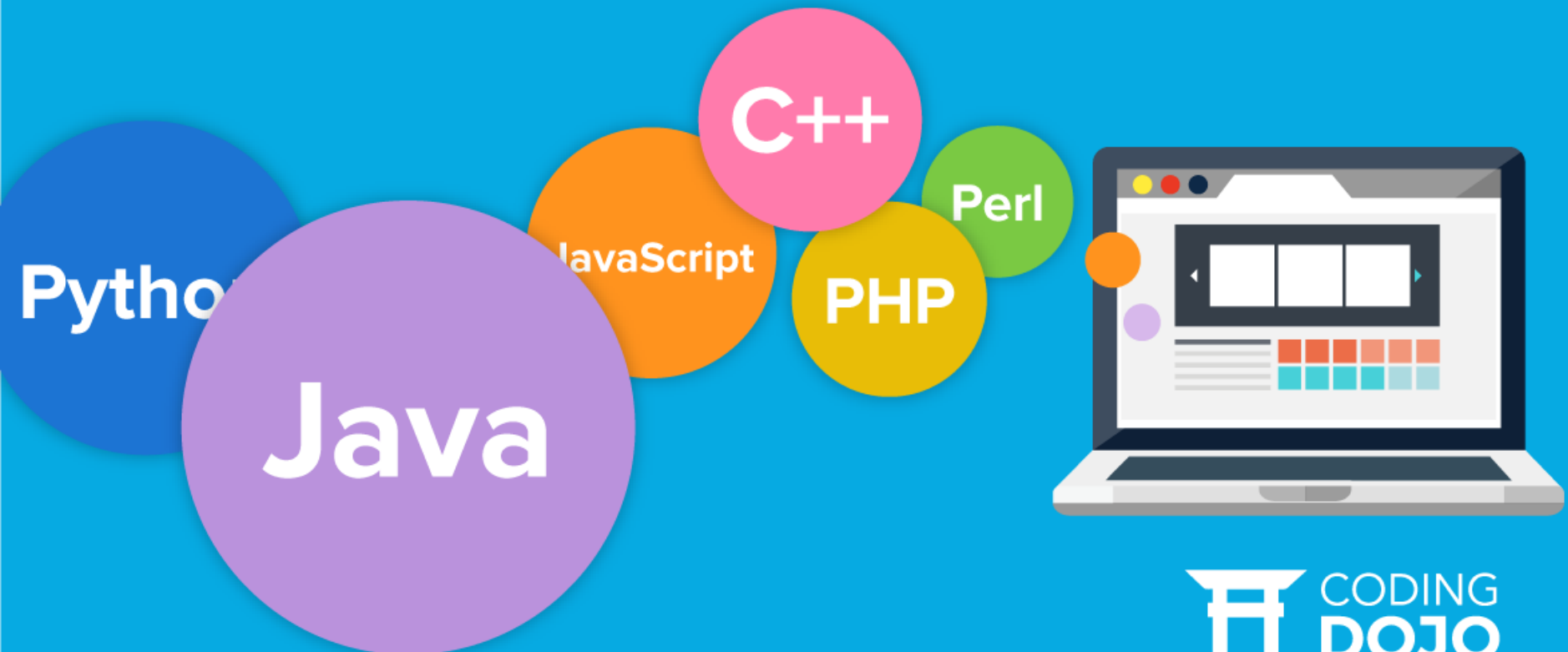


Document Object Model



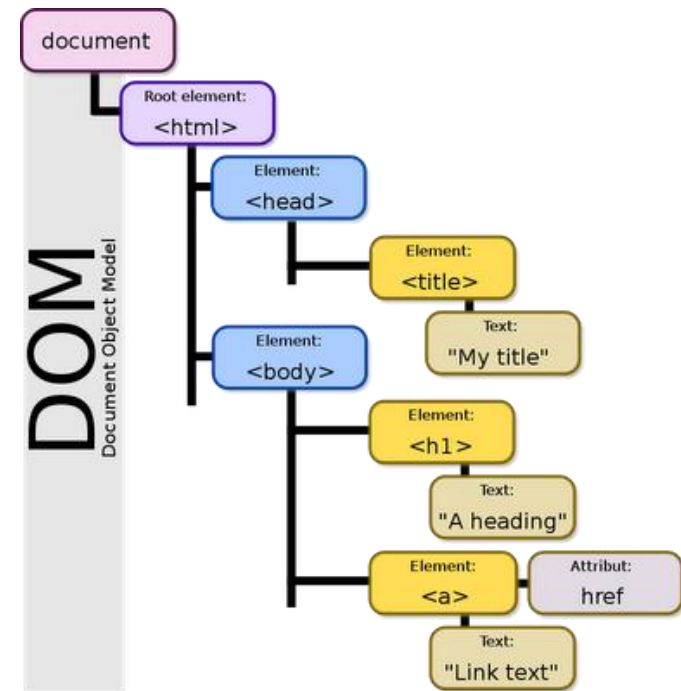


Document Object Model 



DOM

JS HTML



Document Object Model

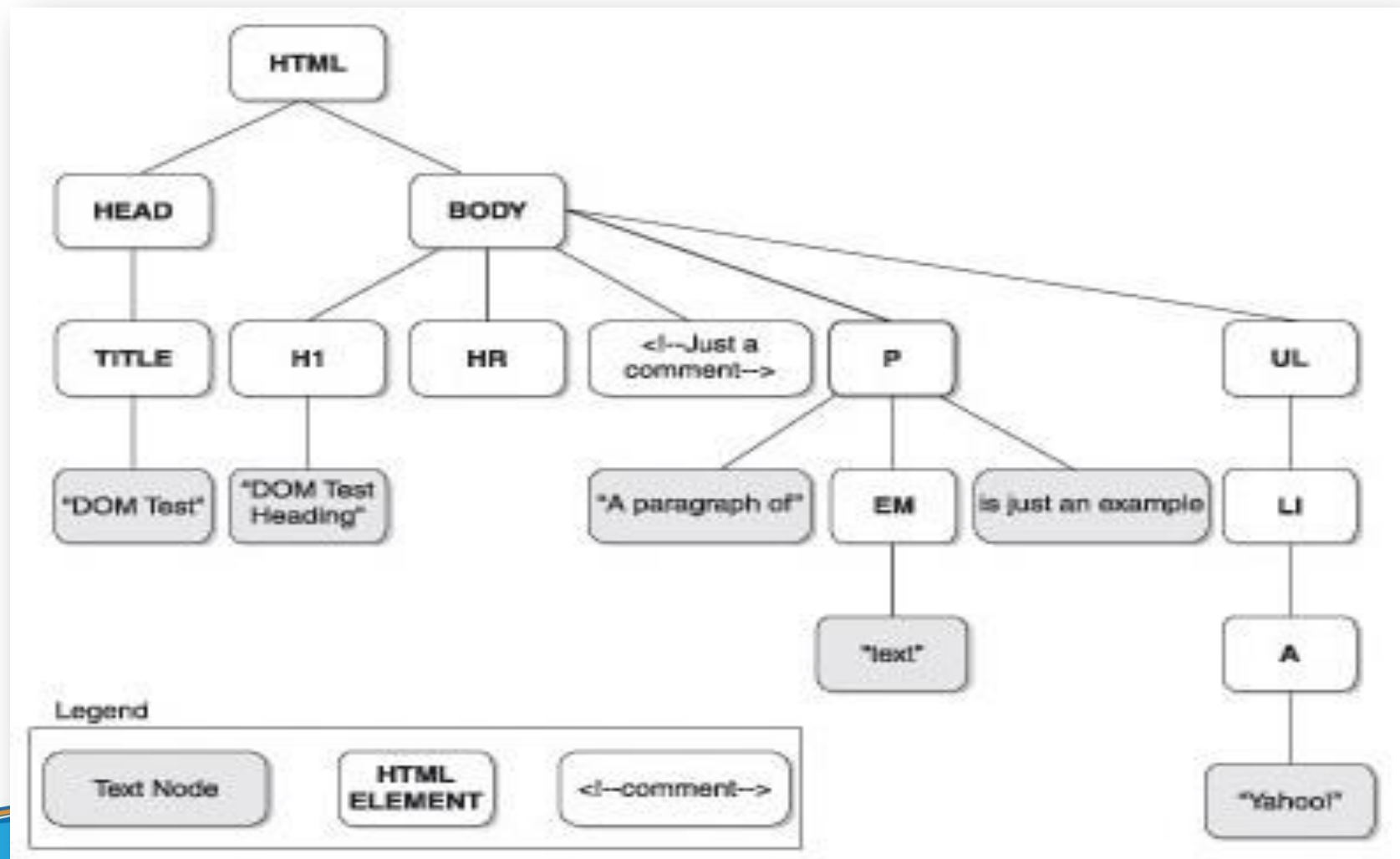
- £ **DOM** = Document Object Model
- £ Là mô hình đối tượng HTML chuẩn định nghĩa cách thức truy xuất và thay đổi thành phần HTML trong trang web (thay đổi nội dung tài liệu của trang)
- £ Mỗi thẻ HTML là một Node đối tượng
 - ❑ Danh sách các thuộc tính (Properties)
 - ❑ Danh sách các phương thức (Methods)
 - ❑ Danh sách các sự kiện (Events)

Đối tượng Document

£ Biểu diễn nội dung của tài liệu theo cấu trúc cây

```
<html>
  <head>
    <title>DOM Test</title>
  </head>
  <body>
    <h1>DOM Test Heading</h1>
    <hr />
    <!-- Just a comment -->
    <p id="p1" >A paragraph of <em>text</em> is just an example</p>
    <ul>
      <li>
        <a href="http://www.yahoo.com" >Yahoo!</a>
      </li>
    </ul>
  </body>
</html>
```

£ Cấu trúc cây nội dung tài liệu



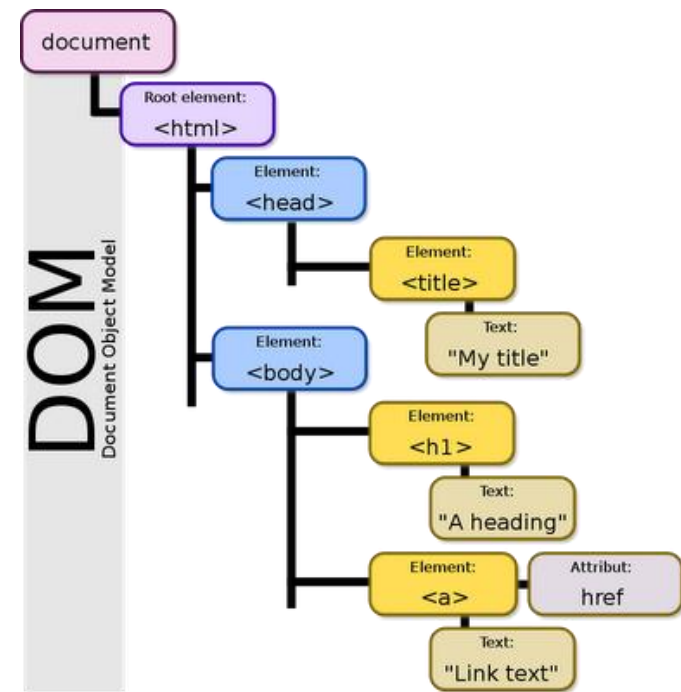
£ Các loại *DOM Node* chính

Node Type Number	Loại	Mô tả	Ví dụ
1	Element	(X)HTML/XML Element	<p>...</p>
2	Attribute	Thuộc tính của HTML/XML Element	align="center"
3	Text	Nội dung chứa trong HTML/XML Element	This is a text fragment!
8	Comment	HTML comment	<!-- This is a comment -->
9	Document	Đối tượng tài liệu gốc, thường là element nằm ở cấp cao nhất trong cây cấu trúc của tài liệu	<html>
10	DocumentType	Định nghĩa loại tài liệu	<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" http://www.w3.org/TR/html4/loose.dtd >



DOM

Các xử lý chính



Document Object Model

Các xử lý chính trên DOM



- £ Truy xuất tìm kiếm đối tượng trên trang web
- £ Thay đổi nội dung trang web



Truy xuất tìm kiếm đối tượng



- **Properties**

- anchors[]
- body
- domain
- lastModified
- title
- URL
- forms[]
- images[]
- childNodes[]
- ...

- **Methods**

- getElementById("id")
- getElementsByName("name")
- getElementsByTagName("tagName")
- getElementsByClass("class")
- ...

Truy xuất trực tiếp với **name**



Ví dụ:

```
//<form name="myForm" ... >  
//  <input type="text" name="myTextbox" value="some text"/>  
//</form>
```

```
var node = document.myForm.myTextbox;  
var text = node.value ; // some text
```

£ **document.getElementById** ("*id1*")

Trả về node có giá trị thuộc tính **id** = *id1*

Ví dụ:

```
//<p id="id1" >  
//      some text  
//</p>
```

Text Node

```
var node = document.getElementById("id1");  
var nodeName = node.nodeName; // p  
var nodeType = node.nodeType; // 1  
var nodeValue = node.nodeValue; // null  
var text = node.innerText; // some text
```

£ **document.getElementsByTagName (“p”)**

Trả về danh sách các node có **tagName = p**

Ví dụ:

```
//<p> paragraph 1</p>  
// <p> paragraph 2</p>
```

```
var nodeList = document.getElementsByTagName(“p”);  
var ptext1 = nodeList[0].innerText ; // paragraph 1  
var ptext2 = nodeList[1].innerText ; // paragraph 2
```

Truy xuất sử dụng DOM Navigation

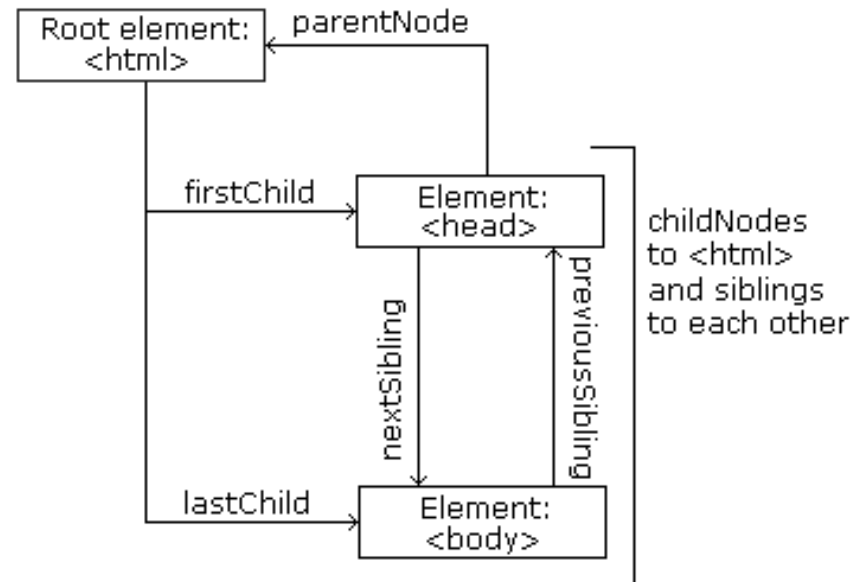


```
<html>

  <head>
    <title>DOM Tutorial</title>
  </head>

  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>

</html>
```



- parentNode
- childNodes[]
- firstChild
- lastChild
- nextSibling
- previousSibling

```
<html>
```

```
<body>
```

```
<h1 id="intro">My First Page</h1>
```

```
<p id="demo">Hello World!</p>
```

```
<script>
```

```
myText = document.getElementById("intro").firstChild.nodeValue;  
document.getElementById("demo").innerHTML = myText;
```

```
</script>
```

```
</body>
```

```
</html>
```

Thay đổi nội dung trang web



- **Properties**

- innerHTML
- innerText
- style
- property

- **Methods**

- document.write("text")
- document.writeln("text")
- document.createTextNode("text")
- document.createElement("HTMLtag")
- removeChild()
- appendChild()
- replaceChild()
- setProperty()
- ...

£ innerHTML

Chỉ định nội dung HTML bên trong một node.

Ví dụ:

```
//<p id="para1" >  
//  some text  
//</p>
```

```
var theElement = document.getElementById("para1");  
theElement.innerHTML = "Some <b> new </b> text";
```

```
// Kết quả :  
// <p id="para1" >  
// Some <b> new </b> text  
// </p>
```

Chỉ định nội dung text bên trong node - innerText



£ innerText

Tương tự *innerHTML*, tuy nhiên bất kỳ nội dung nào đưa vào cũng được xem như là text hơn là các thẻ HTML.

Ví dụ:

```
//<p id="para1" >  
//  some <b>new</b> text  
//</p>  
var theElement = document.getElementById("para1");  
document.write(theElement.innerText);  
  
// Kết quả hiển thị trên trình duyệt  
// "Some new text"
```

Thay đổi CSS - style



£ document.getElementById("id").style.property
= "new style"

```
<html>
<body>

<p id="p2">Hello World!</p>

<script>
document.getElementById("p2").style.color = "blue";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

£ `document.createElement` (“*nodeName*”)

Cho phép tạo ra 1 node HTML mới tùy theo đối số *nodeName* đầu vào

Ví dụ:

```
var imgNode = document.createElement("img");  
imgNode.src = "images/test.gif";
```

```
// 
```

£ document.createTextNode (“*content*”)

Ví dụ:

```
var textNode = document.createTextNode(“New text”);  
var pNode = document.createElement(“p”);  
pNode.appendChild(textNode);
```

```
// <p>New text</p>
```

Chèn node mới - appendChild

£ document.appendChild (newNode)

Chèn node mới **newNode** vào cuối danh sách các node con của một node.

Ví dụ:

```
//<p id="id1" >  
//      some text  
//</p>
```

```
var pNode = document.getElementById("id1");  
var imgNode = document.createElement("img");  
imgNode.src = "images/test.gif";  
pNode.appendChild(imgNode);
```

```
//<p id="id1" > some text  </p>
```



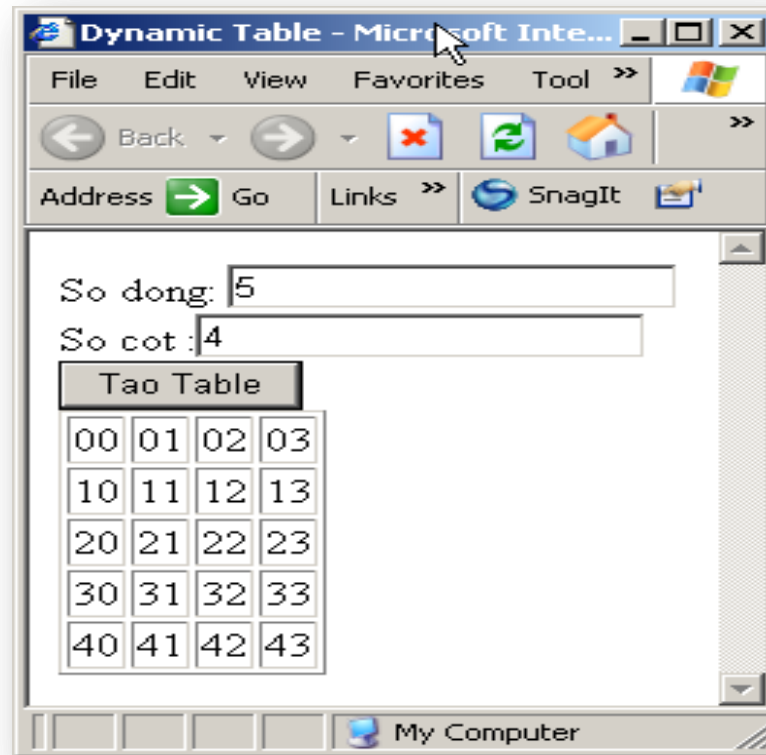
Ví dụ
Ứng dụng



Ngôn ngữ JavaScript



- £ Viết trang web cho phép tạo table có số dòng, số cột do người dùng nhập vào.



Ví dụ Dynamic Table

`<Table>`

```
<Tr>
  <td> 12 </td>
  <td> 13 </td>
</Tr>
```

```
<Tr>
  <td> 21 </td>
  <td> 22 </td>
</Tr>
```

`</Table>`



Ví dụ

body

Table

TBody

Tr

Td

Td

Tr

Td

Td

<Table>

<Tbody>

<Tr>

<td> 12 </td>

<td> 13 </td>

</Tr>

<Tr>

<td> 21 </td>

<td> 22 </td>

</Tr>

</Tbody>

</Table>

Ví dụ Dynamic Table



```
function CreateTable(divTable)
```

```
{
```

```
    var tagTable = document.createElement("table");  
    tagTable.border = 1;
```

```
    var tagTBody = document.createElement("tbody");  
    tagTable.appendChild(tagTBody);
```

```
    var nDong = txtSoDong.value;  
    var nCot = txtSoCot.value;
```

```
    for (i=0; i<nDong; i++)  
    {
```

```
        var tagTR = document.createElement("tr");  
        for (j=0; j<nCot; j++)  
        {
```

```
            var tagTD = document.createElement("td");  
            var textNode = document.createTextNode(i+" "+j);  
            tagTD.appendChild(textNode);
```

```
            tagTR.appendChild(tagTD);  
        }
```

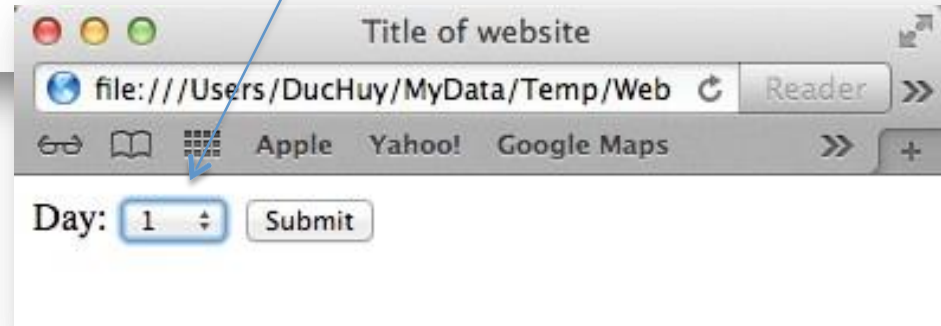
```
        tagTBody.appendChild(tagTR);  
    }
```

```
    divTable.appendChild(tagTable);  
}
```

Ví dụ Tạo ComboBox bằng JavaScript



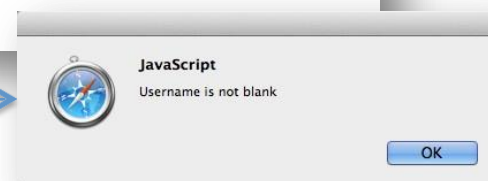
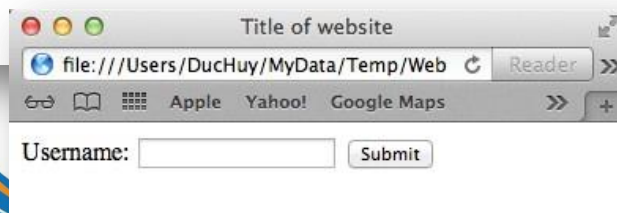
```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Title of website</title>
  </head>
  <body>
    <form name="frmMain" action="#" method="get">
      Day:
      <select name="cmbDay">
        <script type="text/javascript">
          for(var i = 1; i <= 31; i++){
            document.writeln("<option value='" + i + "'>" + i + "</option>");
          }
        </script>
      </select>
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```



Ví dụ Kiểm tra nhập liệu trước khi Submit



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Title of website</title>
    <script type="text/javascript">
      function CheckInput(){
        var us = document.getElementById("txtUS");
        if(us.value == ""){
          alert("Username is not blank");
          us.focus(); //Đưa con trỏ điều khiển đến TextBox Username
          return false;
        }
        return true;
      }
    </script>
  </head>
  <body>
    <form name="frmMain" action="#" method="get" onsubmit="return CheckInput()">
      Username: <input type="text" name="txtUS" id="txtUS" />
      <input type="submit" value="Submit" />
    </form>
  </body>
</html>
```





BOM

JS Browser



Browser Object Model



- £ Browser Object Model
- £ Window là thể hiện của đối tượng cửa sổ trình duyệt
- £ Tồn tại khi mở 1 tài liệu HTML
- £ Sử dụng để truy cập thông tin của các đối tượng trên cửa sổ trình duyệt (tên trình duyệt, phiên bản trình duyệt, thanh tiêu đề, thanh trạng thái ...)

£ Properties

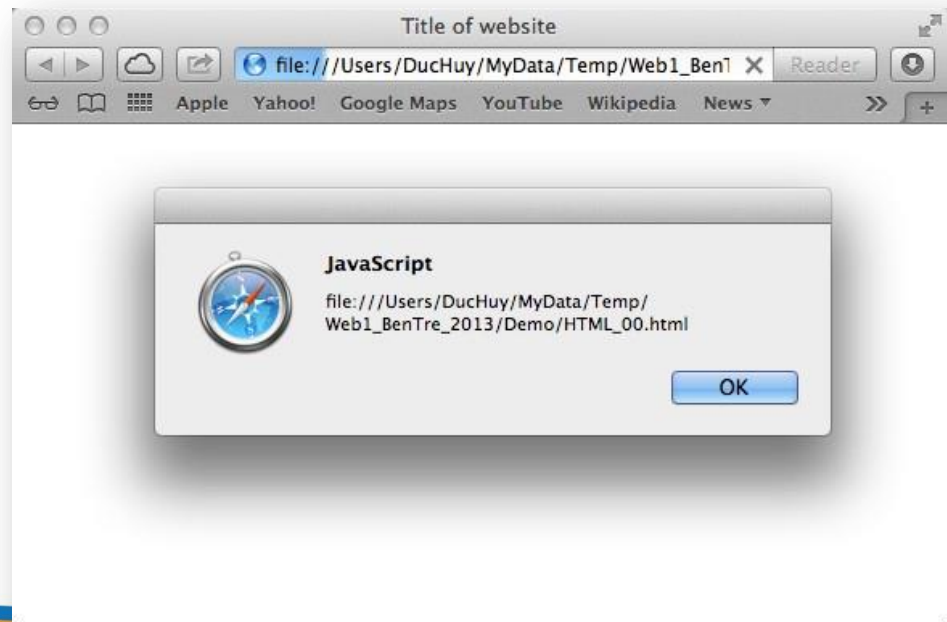
- ☐ **document**
- ☐ event
- ☐ history
- ☐ location
- ☐ name
- ☐ navigator
- ☐ screen
- ☐ status

● Methods

- alert
- confirm
- prompt
- blur
- close
- focus
- open
- moveTo
- resizeTo
- setInterval()
- setTimeout()

£ Ví dụ:

```
<html>
  <body>
    <script type="text/javascript">
      var curURL = window.location;
      window.alert(curURL);
    </script>
  </body>
</html>
```



Window Size



£ IE, Chrome, Firefox, Opera, và Safari:

- `window.innerHeight`

- `window.innerWidth`

£ IE 8, 7, 6, 5:

- `document.documentElement.clientHeight`

- `document.documentElement.clientWidth`

Hoặc

- `document.body.clientHeight`

- `document.body.clientWidth`

£ `Screen.width / screen.height`

□ Trả về chiều rộng/cao pixel của màn hình người dùng

£ `Screen.availWidth / screen.availHeight`

□ Trả về chiều rộng/cao pixel của màn hình người dùng trừ vùng window taskbar

£ `Screen.colorDepth`

□ 24bits : true color

□ 32bits : deep color



Window Screen – Ví dụ



```
<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
document.getElementById("demo").innerHTML =
"Screen (width x height) is " + screen.width + " x " + screen.height + "<br/>"
+ "Available screen width is " + screen.availWidth + " x " + screen.availHeight + "<br/>"
+ "Color Depth is " + screen.colorDepth;
</script>

</body>
</html>
```

£ Trả về thông tin đường dẫn của tài liệu hiện hành

- ☐ `window.location.href`
- ☐ `window.location.hostname`
- ☐ `window.location.pathname`
- ☐ `window.location.protocol`
- ☐ `window.location.assign(url)`

- £ window.history.goBack()
- £ window.history.goForward()

```
<html>
<head>
<script>
function goBack() {
    window.history.back()
}
</script>
</head>
<body>

<input type="button" value="Back" onclick="goBack()">

</body>
</html>
```

£ Trả về thông tin về trình duyệt của người dùng

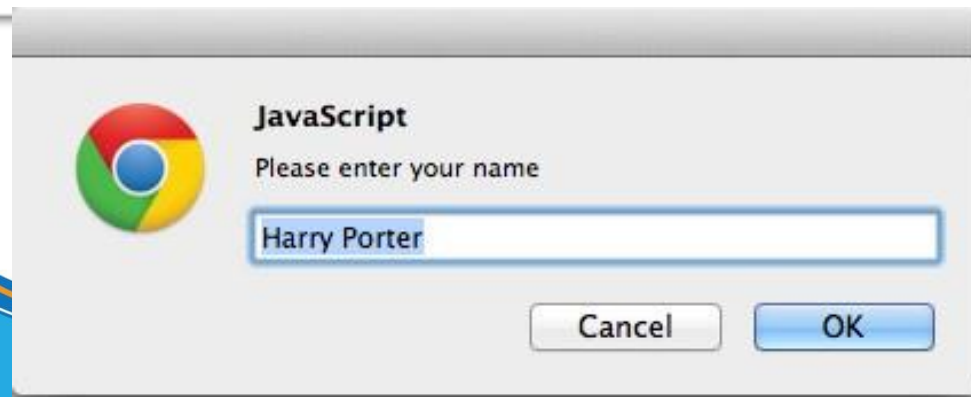
- ☐ navigator.cookieEnabled
- ☐ navigator.javaEnabled()
- ☐ navigator.appName
- ☐ navigator.appCodeName
- ☐ navigator.appVersion
- ☐ navigator.platform
- ☐ navigator.language

Window Popup Alert

```
alert("I am an alert box!");
```

```
var r = confirm("Press a button");
if (r == true) {
    x = "You pressed OK!";
} else {
    x = "You pressed Cancel!";
}
```

```
var person = prompt("Please enter your name", "Harry Potter");
if (person != null) {
    document.getElementById("demo").innerHTML =
        "Hello " + person + "! How are you today?";
}
```



£ Định kỳ gọi hàm thực thi sau 1 khoảng thời gian

□ window.setInterval("function name", milliseconds)

```
var myVar=setInterval(function () {myTimer()}, 1000);

function myTimer() {
    var d = new Date();
    document.getElementById("demo").innerHTML = d.toLocaleTimeString();
}
```

□ window.clearInterval(intervalVariable)

£ Đợi 1 khoảng thời gian trước khi gọi hàm thực thi

- ❑ `window.setTimeout("function name", milliseconds)`
- ❑ `window.clearTimeout(intervalVariable)`

```
<button onclick="myVar=setTimeout(function(){alert('Hello')},3000)">Try it</button>  
<button onclick="clearTimeout(myVar)">Try it</button>
```

£ Document Object Model

- ☐ Document
- ☐ Thay đổi nội dung động
- ☐ Form Validation

£ Browser Object Model

- ☐ Window

