

Bài 7b

HTML DOM với Javascript

Khoa CNTT – ĐH.KHTN

Nội dung

- Giới thiệu DOM và HTML DOM
- Các đối tượng DOM được hỗ trợ trong Javascript
- Một số thuộc tính và phương thức trong HTML DOM
- HTML DOM và xử lý sự kiện bằng Java script

Giới thiệu về DOM

- **DOM** = Document Object Model
- **DOM** là tổ mô hình cho phép truy xuất đến các thành phần trong một tài liệu có cấu trúc.
- **DOM** được hỗ trợ bởi tất cả các trình duyệt.
- **DOM** trong trình duyệt là một thư viện cung cấp các đối tượng, thuộc tính, phương thức hỗ trợ để truy xuất các tài liệu HTML, XHTML, XML.
- Sử dụng javascript để tương tác với DOM.

DOM - các thuộc tính và phương thức

- Mỗi đối tượng DOM đều có danh sách thuộc tính (**Properties**) và danh sách các phương thức (**Method**) tương ứng.

- **Cách sử dụng:**

objectName.**propertyName** = value

objectName.**methodName**()

- **Ví dụ:**

document.**bgColor** = "blue";

window.**focus**();

HTML DOM

- **HTML DOM** = **HTML** Document Object Model
- **HTML DOM**: là một thành phần của DOM
- **HTML DOM**: Xem trang web như một cây gồm nhiều node
- Mỗi node là một thành phần:
 - tag HTML
 - thuộc tính
 - nội dung của tag

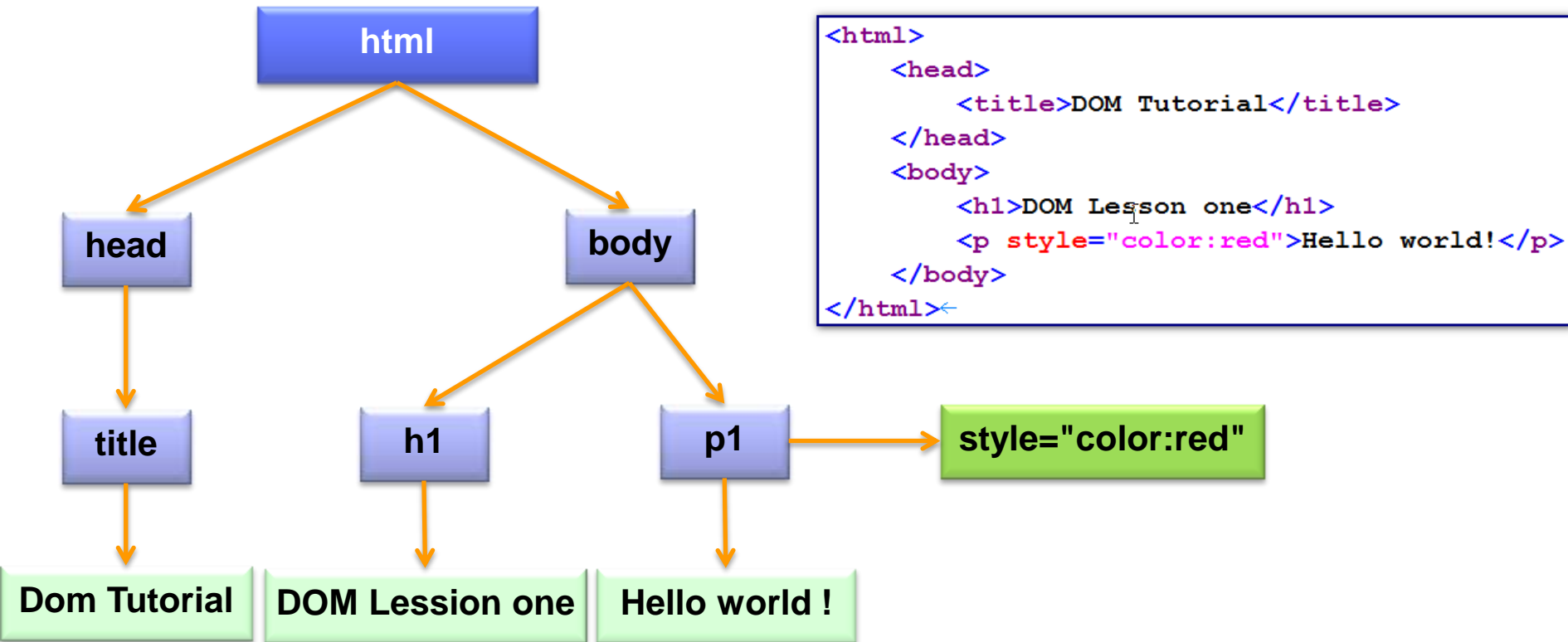
HTML DOM

- HTML DOM định nghĩa một cách để truy xuất và điều khiển các thành phần trong tài liệu HTML
- Là chuẩn của W3C
- **"HTML DOM là một chuẩn cho biết làm thế nào để đọc, thay đổi, thêm, xóa các thành phần của tài liệu HTML"**

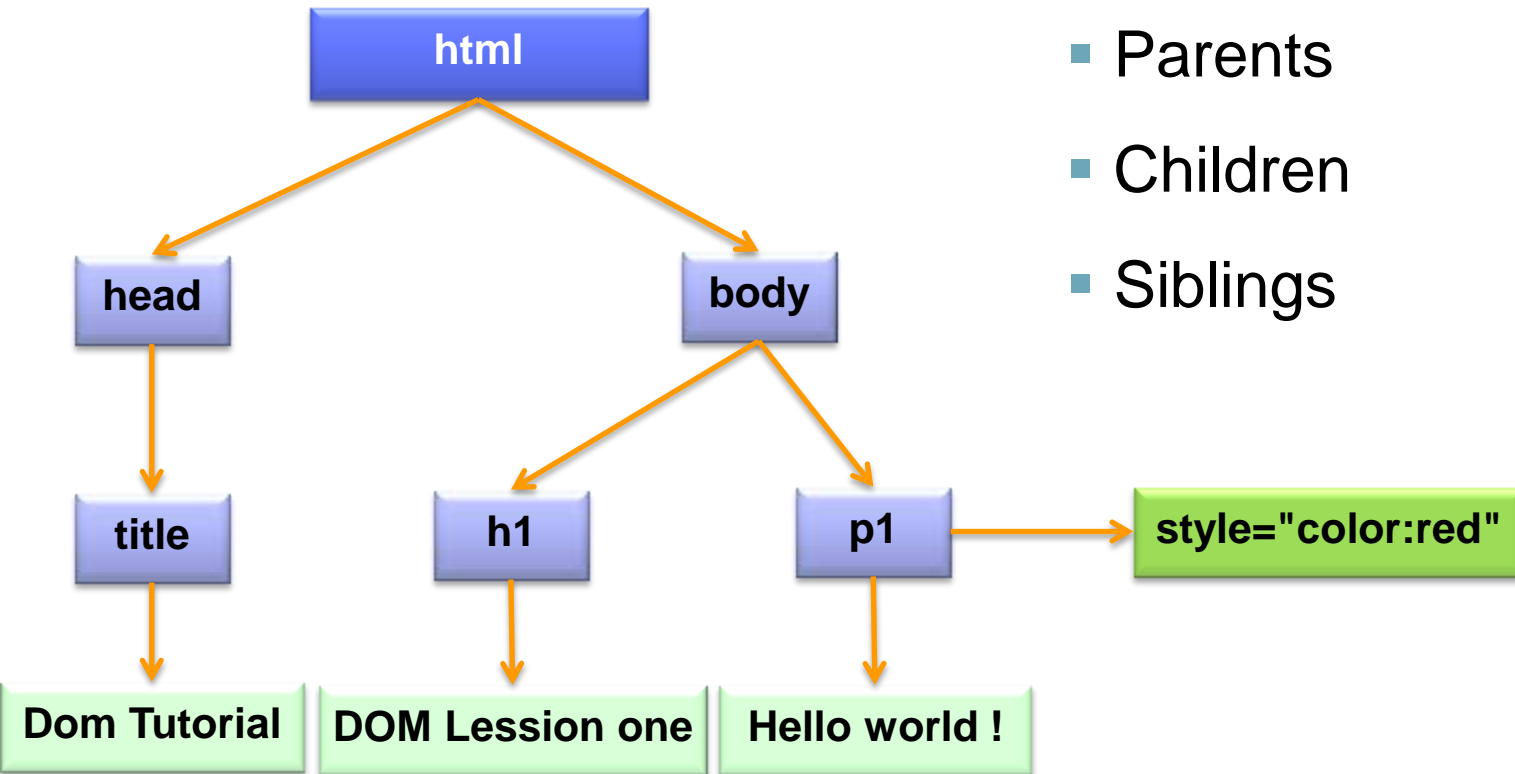
HTML DOM - Các loại node

- **document node**: the entire document
- **element node**: HTML tag
- **text node**: The text in the HTML element
- **attribute node**: HTML attribute
- **comment node**: comments

HTML DOM - Các loại node

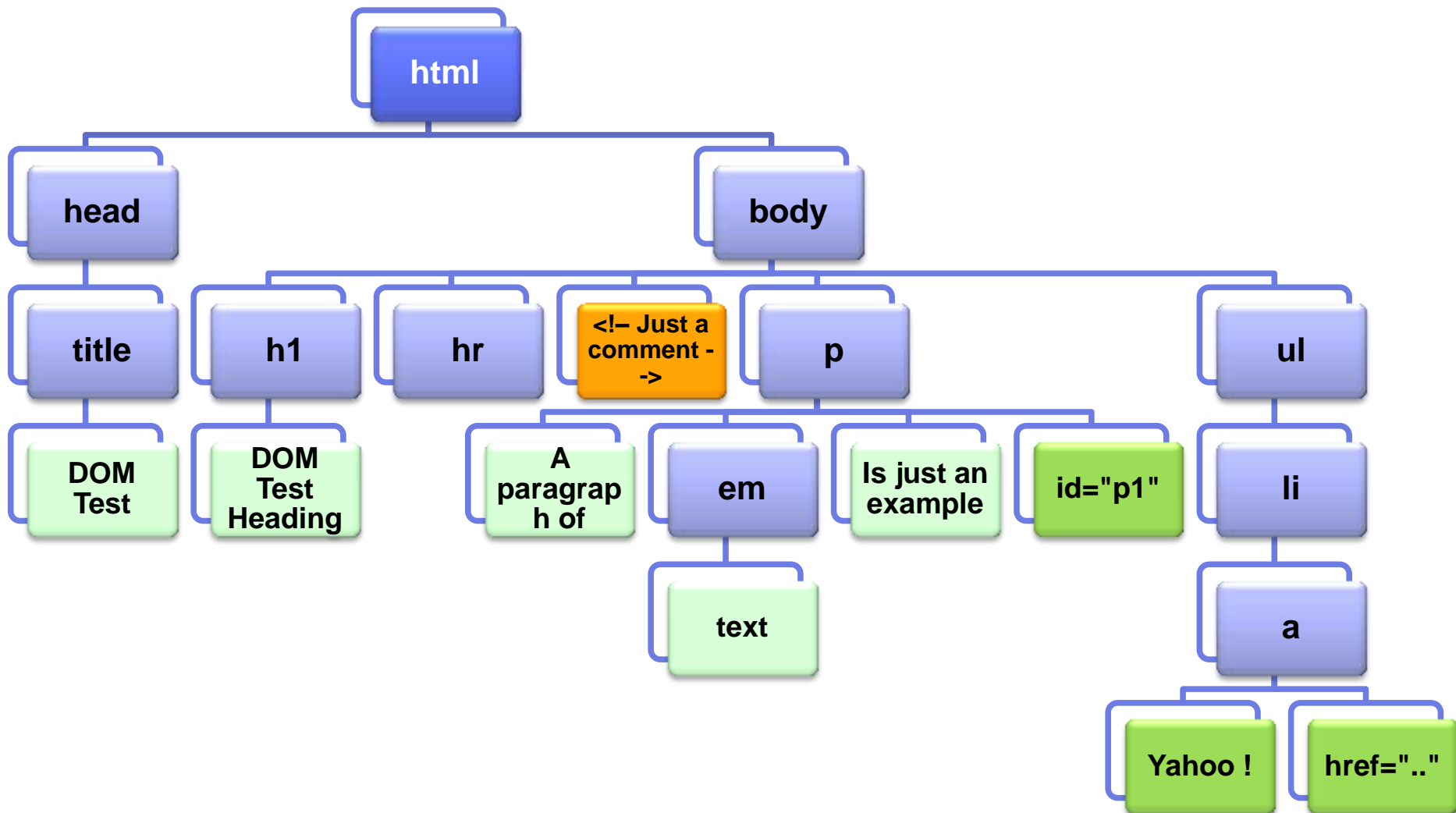


HTML DOM - Quan hệ giữa các node



HTML DOM - Ví dụ

```
<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>
```



HTML DOM - Thuộc tính và phương thức của node

Thuộc tính

- **x.innerHTML**
- **x.parentNode**
- **x.childNodes**
- **x.attributes**

- **x.nodeName**
- **x.nodeValue**
- **x.nodeType**

Phương thức

- **x.getElementById(*id*)**
- **x.getElementsByTagName(*name*)**
- **x.appendChild (*node*)**
- **x.removeChild (*node*)**

- **x.firstChild**
- **x.lastChild**

Truy cập đến các node - **document node**

- **nodeName**: **#document**
- **nodeType** = 9
- **Cách truy cập**:
 - Sử dụng: **document** hoặc **window.document**
- Một số thuộc tính và phương thức:
 - document.**documentElement**: node gốc của tài liệu HTML (thẻ html)
 - document.**body**: node body (thẻ body)
 - document.**getElementById**("id_of_element_node")
 - document.**getElementsByTagName**("tagname_of_node")

Truy cập đến các node - **element node**

- **nodeName**: tên thẻ HTML
- **nodeType** = 1
- **nodeValue** = undefined
- **Cách truy cập:**
 1. Sử dụng phương thức **getElementById ()**
 2. Sử dụng phương thức **getElementsByTagName ()**
 3. Sử dụng phương thức **getElementsByName ()**
 4. Sử dụng cấu trúc của cây để định hướng, dựa trên các quan hệ giữa các node để truy xuất.

Truy cập đến các node - **element node**

- Nhận kết quả trả về
 - DOM element
 - DOM Node List
 - chỉ số bắt đầu là **0**
 - **length**: chiều dài

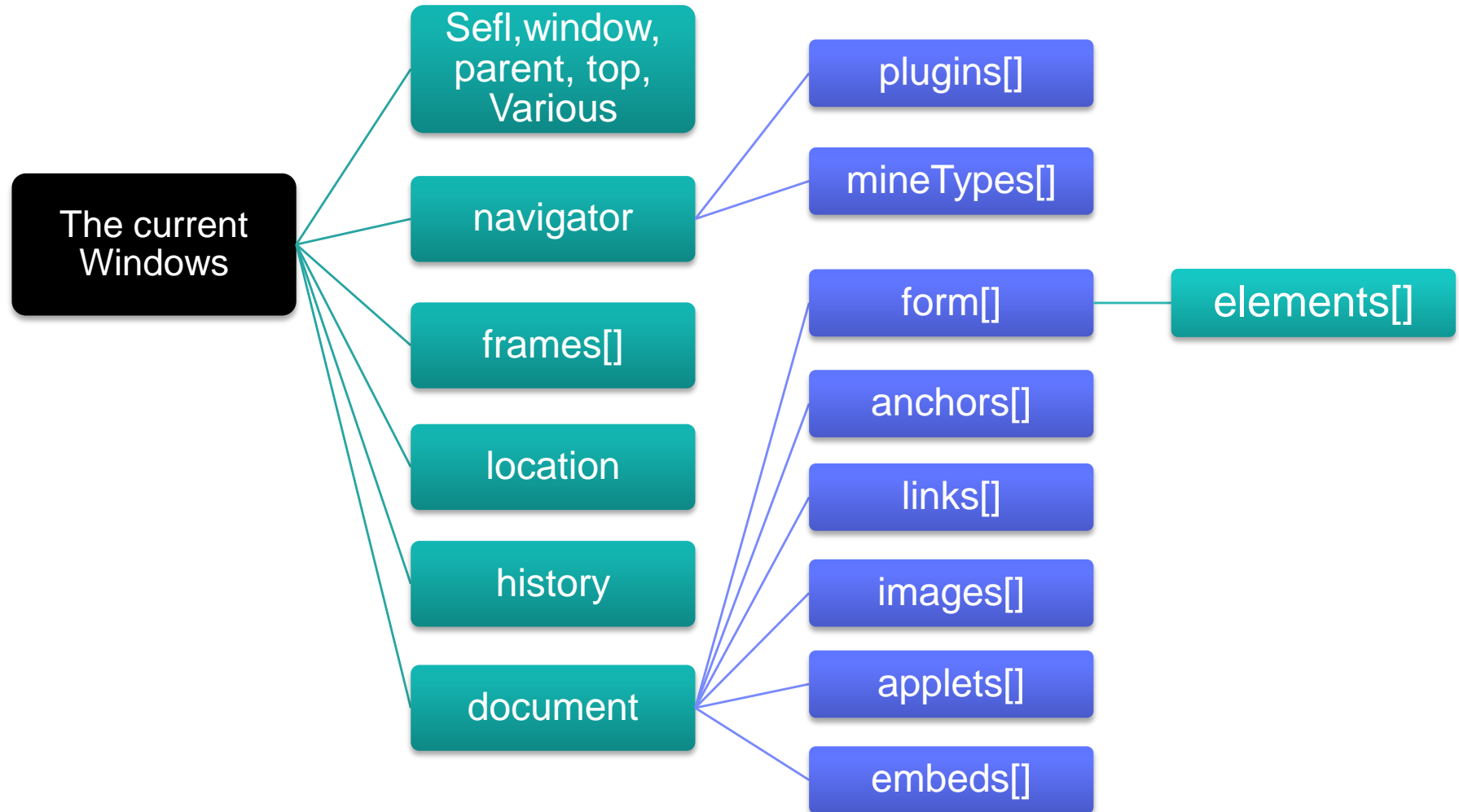
Truy cập đến các node - **attributes node**

- attributes node là con của các element node hoặc document node.
- **nodeName**: tên thuộc tính
- **nodeValue**: giá trị của thuộc tính
- **nodeType** = 2
- **Cách truy cập**: sử dụng thuộc tính **attributes** của một **element node**

Truy cập đến các node - **text node**

- **nodeName** của text node luôn là **#text**
- **nodeValue** của text node là một chuỗi.
- **nodeType** = 3
- **Cách truy cập**: Sử dụng cấu trúc của cây để định hướng, dựa trên các quan hệ giữa các node để truy xuất.

Các đối tượng DOM trong javascript



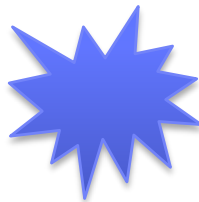
DOM - **window** object

- 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** window
- **Điều khiển các sự kiện** xảy ra trong window
- Nếu tài liệu định nghĩa nhiều frame, browser tạo 1 window object cha và các window object con cho từng frame

DOM – **window** object (tt)

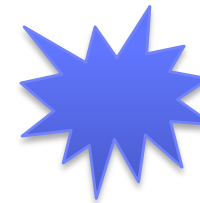
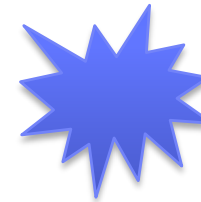
Thuộc tính

- document
- history
- location
- parent
- frames[]
- name
- status
- event
- screen
- ...



Phương thức

- alert
- confirm
- prompt
- blur
- focus
- open
- close
- setTimeout
- setInterval
- ...



DOM – **navigator** Object

- Cung cấp thông tin về trình duyệt Browser

Thuộc tính

- appName
- appVersion
- appCodeName
- **cookieEnabled**
- online
- platform
- ...



Phương thức

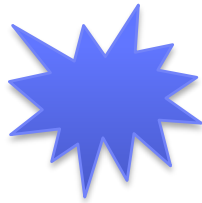
- javaEnabled()
- ...

DOM – **location** Object

- Chứa thông tin về URL hiện tại

Thuộc tính

- hash
- host
- **hostname**
- **href**
- pathname
- port
- protocol
- search



Phương thức

- **assign(url)**
- reload()
- **replace(url)**

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
```

```
<body>
```

```
  <script type="text/javascript">
```

```
    document.write("appName: " + navigator.appName);
```

```
    document.write("<br/>appVersion : " + navigator.appVersion);
```

```
    document.write("<br/>appCodeName : " + navigator.appCodeName);
```

```
    document.write("<br/>cookieEnabled : " + navigator.cookieEnabled);
```

```
    document.write("<br/>hostname : " + location.hostname);
```

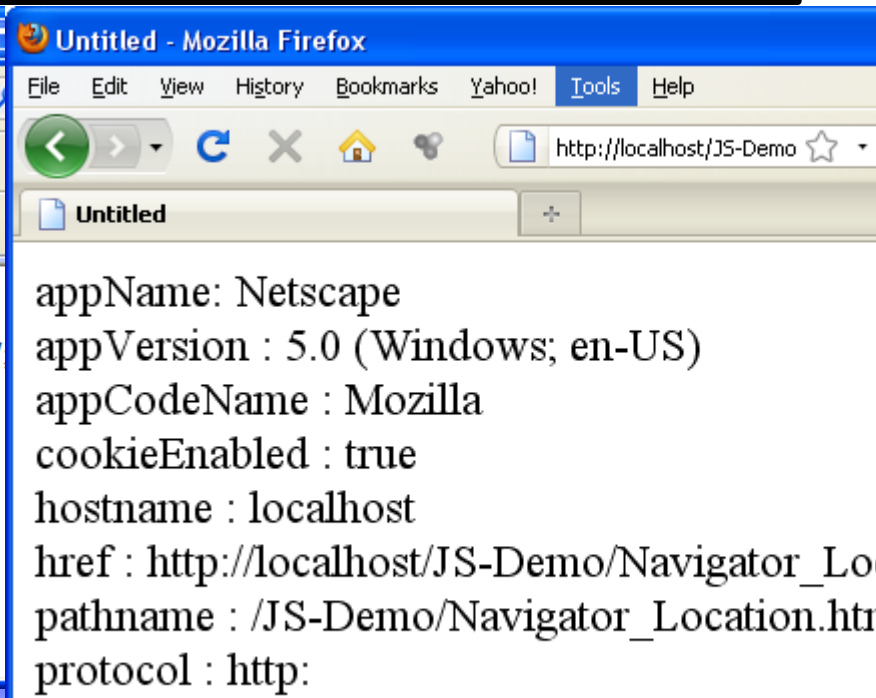
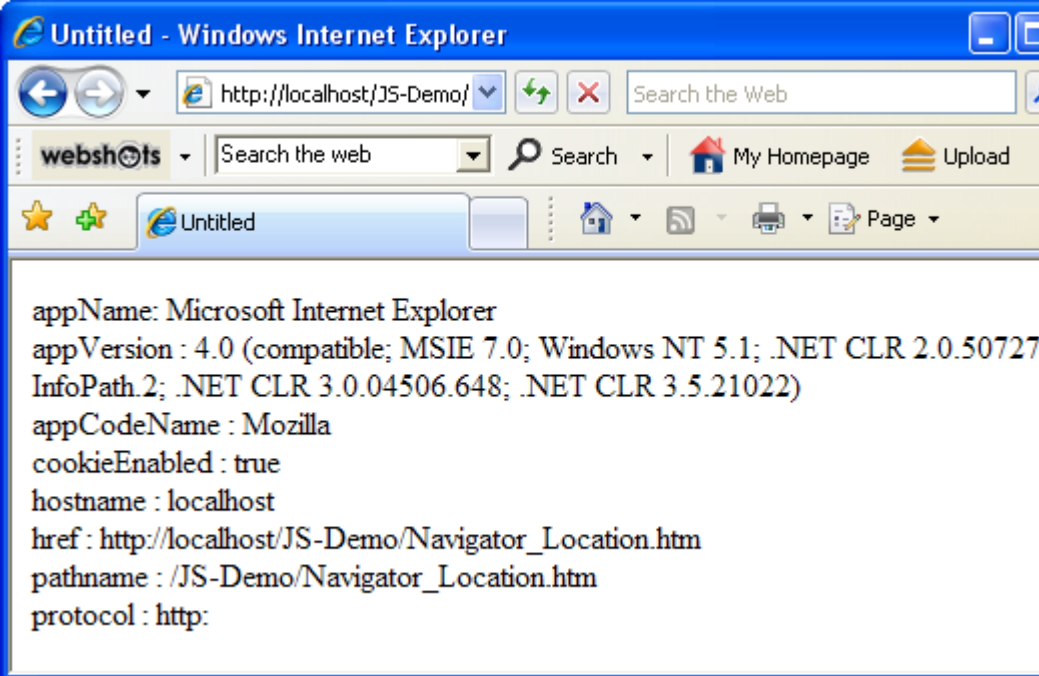
```
    document.write("<br/>href : " + location.href);
```

```
    document.write("<br/>pathname : " + location.pathname);
```

```
    document.write("<br/>protocol : " + location.protocol);
```

```
  </script>
```

```
</body>
```



DOM – **history** Object

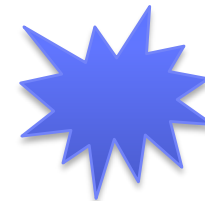
- Cung cấp danh sách các URL đã được duyệt bởi người dùng

Thuộc tính

- length

Phương thức

- **back()**
- **go(url)**
- **forward()**



DOM – **document** Object

- Biểu diễn toàn bộ **các thành phần** trong 1 tài liệu HTML
- Dùng để lấy **thông tin** về tài liệu, các thành phần HTML và xử lý sự kiện.
- Theo cấu trúc HTML DOM



HTML DOM – **document** Object

Thuộc tính

- **bgColor**
- **fgColor**
- **aLinkColor**
- **linkColor**
- **vlinkColor**
- **lastModified**
- **location**
- **referrer**
- **title**



Phương thức

- **clear()**
- **close()**
- **open(...,...)**
- **write(text)**
- **writeln(text)**

- **getElementById("id")**
- **getElementByName("Name")**
- **getElementsByTagName("tagName")**

- **createTextNode(" text ")**
- **createElement("HTMLtag")**

Tập hợp

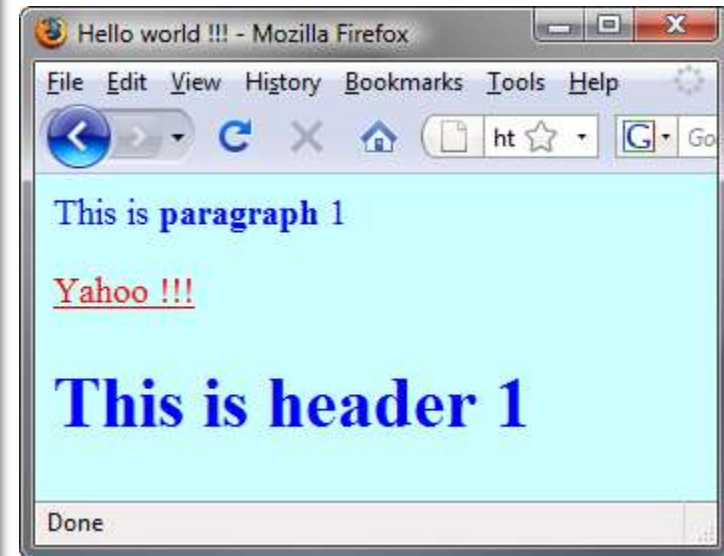
- anchors []**
- forms []**
- frames []**
- images []**
- links []**

HTML DOM – **document** Object

```

<html>
<head>
  <title>document Object</title>
</head>
<body>
  <p id="p1"> This is <b> paragraph </b> 1 </p>
  <a href="http://www.yahoo.com">Yahoo !!!</a>
  <script type="text/javascript">
    document.write("<h1>This is header 1</h1>");
    document.bgColor = "#CCFFFF";
    document.fgColor = "blue";
    document.title = "Hello world !!!";
    document.alinkColor = "green";
    document.vlinkColor = "red";
  </script>
</body>
</html>*

```



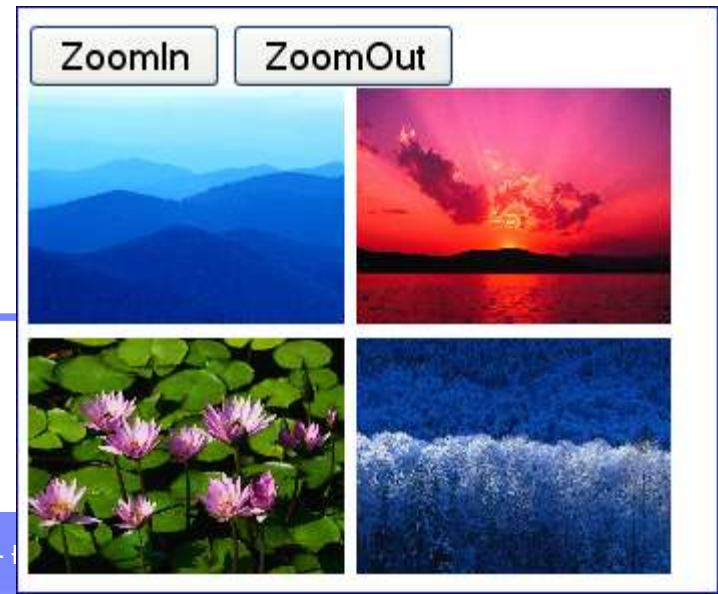
HTML DOM – Image Object

- Truy xuất đến tag `` trên trang web
- Có thể truy xuất thông qua :
 - `document.images[index]`
 - `document.images["ImageName"]`
 - `document.ImageName`
- Một số thuộc tính của Image Obj :
 - `alt`, `border`, `classname`, `title`,
 - `width`, `height`, `hspace`, `vspace`,
 - `id`, `name`, `src`, `lowsrc`, `longDesc`,
 - `isMap`, `complete`

```

<html>
<head>
  <title>Untitled</title>
  <script type="text/javascript">
    function ZoomIn(){
      for (var i=0; i < document.images.length; i++)
        document.images[i].width += 10;
    }
    function ZoomOut(){
      for (var i=0; i < document.images.length; i++)
        document.images[i].width -= 10;
    }
  </script>
</head>
<body>
  <input type="button" onclick="ZoomIn();" value="ZoomIn"/>
  <input type="button" onclick="ZoomOut();" value="ZoomOut"/>
  <br/>
  
  
  
  
</body>
</html>

```



HTML DOM – **form** Object

- Dùng để truy xuất đến tag **<form>** trên trang web
- Có thể truy xuất thông qua
 - `document.forms[index]`
 - `document.forms["FormName"]`
 - `document.FormName`
- Một số thuộc tính :
 - **action**, **method**, **id**, **name**, **target**
 - **classname**, **title**, **language**, **dir**
 - **elements[]**
- Một số phương thức :
 - `reset()`, **`submit()`**

HTML DOM – **element** Object

- Truy cập đến các thành phần của form (form field).

document.formName.controlName

- Ví dụ:

```
<form name="searchForm" action="cgi-bin/search.pl">  
  <input type="text" name="entry">  
  <input type="submit" name="sender" value="Search">  
</form>
```

có thể truy xuất textbox **entry** bằng các lệnh sau:

document.searchForm.entry

document.searchForm.elements[0]

document.forms["searchForm"].elements["entry"]

document.forms["searchForm"].entry

DOM – `document.getElementById ()`

Sử dụng: `getElementById ("id_of_element")`

Kết quả trả về: node đầu tiên có giá trị thuộc tính **id = id_of_element**

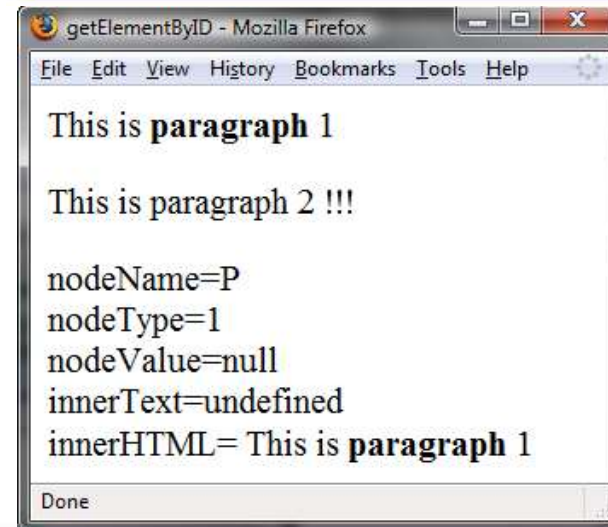
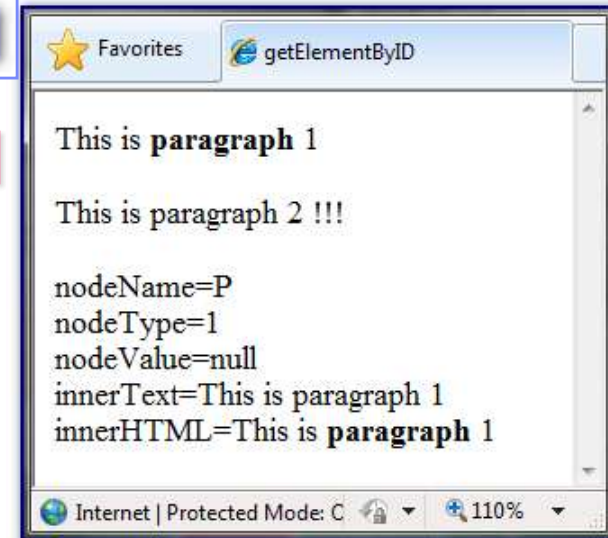
DOM – document.getElementById

```
<html>
<head>
  <title>Untitled</title>
</head>
<body>

<p id="p1"> This is <b> paragraph </b> 1 </p>
<p id="p2"> This is paragraph 2 !!!</p>

<script type="text/javascript">
  var node= document.getElementById("p1");
  document.write("nodeName=" + node.nodeName + "<br/>");
  document.write("nodeType=" + node.nodeType + "<br/>");
  document.write("nodeValue=" + node.nodeValue + "<br/>");
  document.write("innerText=" + node.innerText + "<br/>");
  document.write("innerHTML=" + node.innerHTML + "<br/>");
</script>

</body>
</html>
```



Method or property	IE 5.5	IE 6	IE 7	IE8rc1 as IE7	IE8rc1 as IE8	FF 2.0	FF 3.0	FF 3.1b2	Saf 3.0 Win	Saf 3.1 Win	iPhone 3G	Chrome 1.0	Opera 9.62	Opera 10a	Konqueror 3.5.7
innerText		yes				no			yes			yes	yes	yes	yes

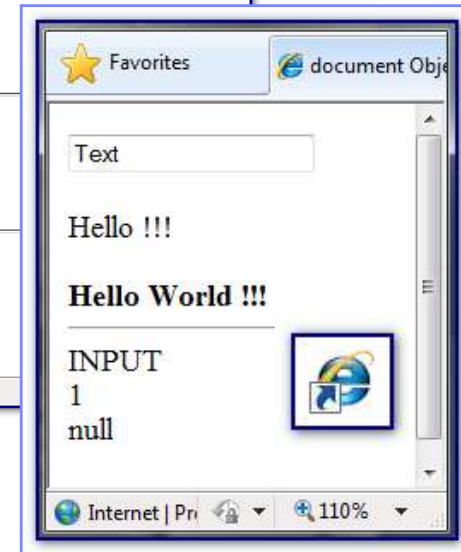
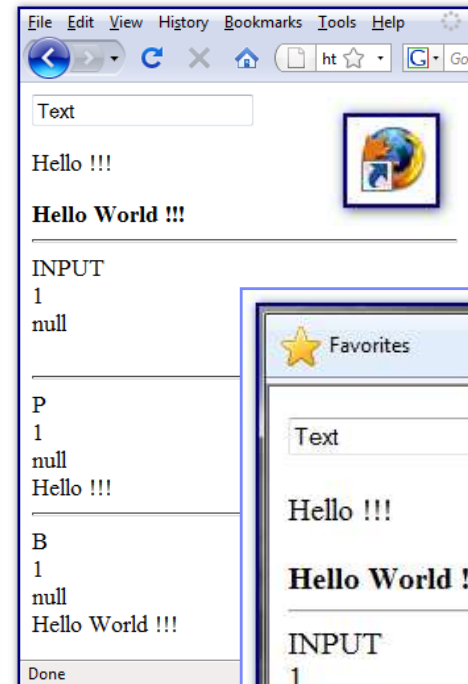
DOM – `document.getElementsByName()`

Cú pháp: `getElementsByName ("node_name")`

Kết quả: Trả về danh sách node có `name = node_name`

```
<body>
  <input type="text" name="abc" value="Text" />
  <p name="abc">
    Hello !!!
  </p>
  <b name="abc"> Hello World !!!</b>

  <script type="text/javascript">
    var nodeList= document.getElementsByName("abc");
    for(var i=0; i<nodeList.length; i++)
    {
      document.write("<hr/>");
      document.write(nodeList[i].nodeName + "<br/>");
      document.write(nodeList[i].nodeType + "<br/>");
      document.write(nodeList[i].nodeValue + "<br/>");
      document.write(nodeList[i].innerHTML + "<br/>");
    }
  </script>
</body>
```

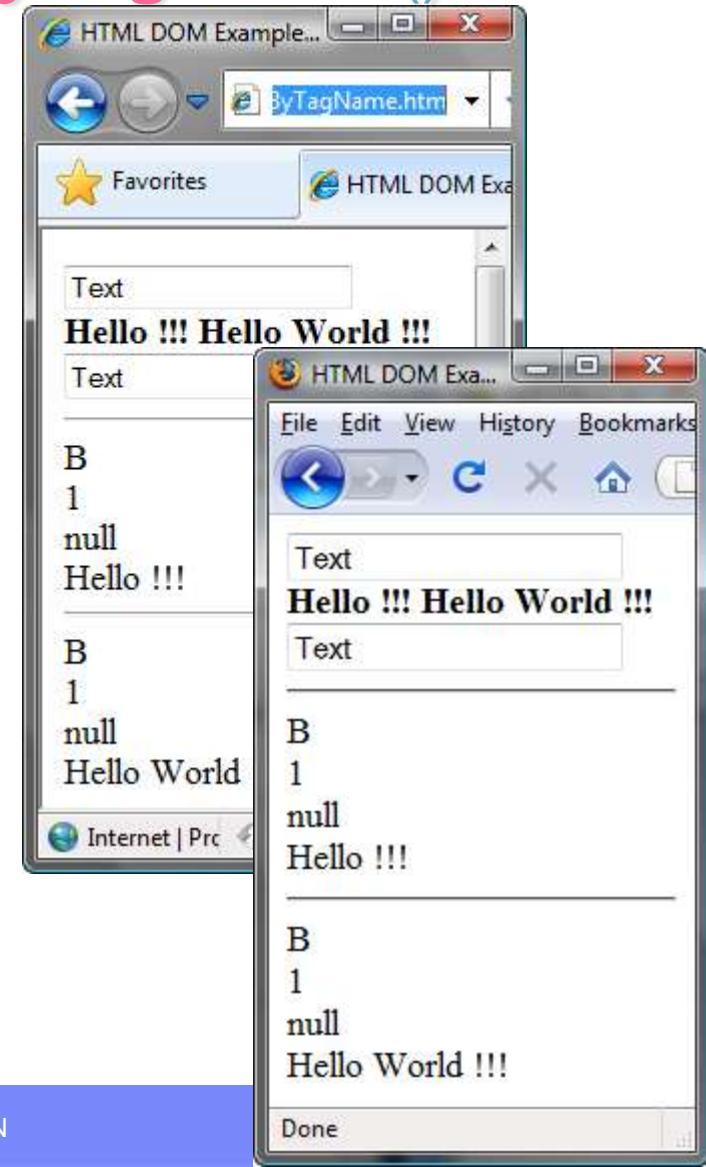


DOM – `document.getElementsByTagName()`

Cú pháp: `getElementsByTagName ("tag_name")`

Kết quả: Trả về danh sách node có `tag = tag_name`

```
<body>
  <input type="text" name="abc" value="Text" />
  <b name="b1">
    Hello !!!
  </b>
  <b name="b2"> Hello World !!!</b>
  <input type="text" name="abc" value="Text" />
  <script type="text/javascript">
    var nodeList= document.getElementsByTagName("b");
    for(var i=0; i<nodeList.length; i++)
    {
      document.write("<hr/>");
      document.write(nodeList[i].nodeName + "<br/>");
      document.write(nodeList[i].nodeType + "<br/>");
      document.write(nodeList[i].nodeValue + "<br/>");
      document.write(nodeList[i].innerHTML + "<br/>");
    }
  </script>
</body>
```



DOM – createElement() , createTextNode() , appendChild()

Cú pháp:

document.createElement ("nodeName") : tạo một element node (thẻ HTML)

```
var pnode = document.createElement("p");
```

document.createTextNode ("content") : tạo một text node

```
var normal_text = document.createTextNode("This is normal text");
```

node.appendChild (child_node): thêm **child_node** vào cuối danh sách node con của **node**

```
pnode.appendChild(normal_text);
```

```
document.body.appendChild(pnode);
```

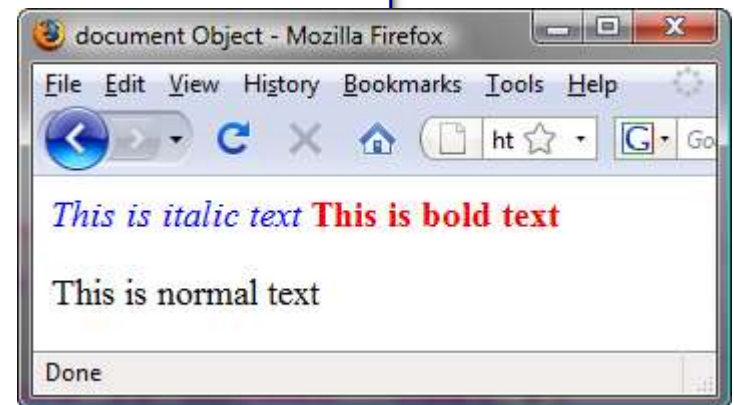
DOM – createElement() , createTextNode() , appendChild()

```

<html>
<head>
  <title>document Object</title>
</head>
<body>
  <i id="i1" style="color:blue"> </i>
  <b id="b1" style="color:red"> </b>
  <script type="text/javascript">
    var italic_text = document.createTextNode("This is italic text");
    var bold_text = document.createTextNode("This is bold text");
    var normal_text = document.createTextNode("This is normal text");

    var inode = document.getElementById("i1");
    var bnode = document.getElementById("b1");
    var pnode = document.createElement("p");
    pnode.appendChild(normal_text);
    bnode.appendChild(bold_text);
    inode.appendChild(italic_text);
    document.body.appendChild(pnode)
  </script>
</body>
</html>

```



DOM – insertBefore() , removeChild() , replaceChild()

node.insertBefore(newChild, childRef) : Chèn node **newChild** vào trước node **childRef** trong danh sách các node con của **node**

```
var normal_text = document.createTextNode("This is new line !");
var pnode = document.createElement("p");
pnode.appendChild(normal_text);
document.body.insertBefore(pnode, document.body.firstChild);
```

node.removeChild(child) : Xóa node **child** ra khỏi danh sách các node con của **node**, giá trị trả về là node bị xóa.

```
var p1node = document.getElementById("p1");
document.body.removeChild(p1node);
```

node.replaceChild(newChild, oldChild) : thay thế node **oldChild** bởi node **newchild** trong danh sách các node con của **node**.

```
var inode = document.createElement("i");
var i_text = document.createTextNode("This is italic text !");
inode.appendChild(i_text);
var p2node = document.getElementById("p2");
document.body.replaceChild(inode, p2node);
```


DOM – insertBefore() , removeChild() , replaceChild()

```

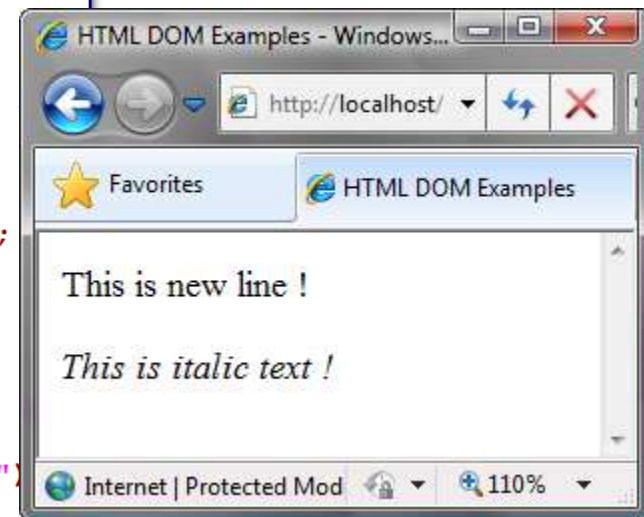
<body>
  <p id="p1" style="color:blue"> This is line 1 </p>
  <p id="p2" style="color:red"> This is line 2</p>

  <script type="text/javascript">
    var normal_text = document.createTextNode("This is new line");
    var pnode = document.createElement("p");
    pnode.appendChild(normal_text);
    document.body.insertBefore(pnode, document.body.firstChild);

    var p1node = document.getElementById("p1");
    document.body.removeChild(p1node);

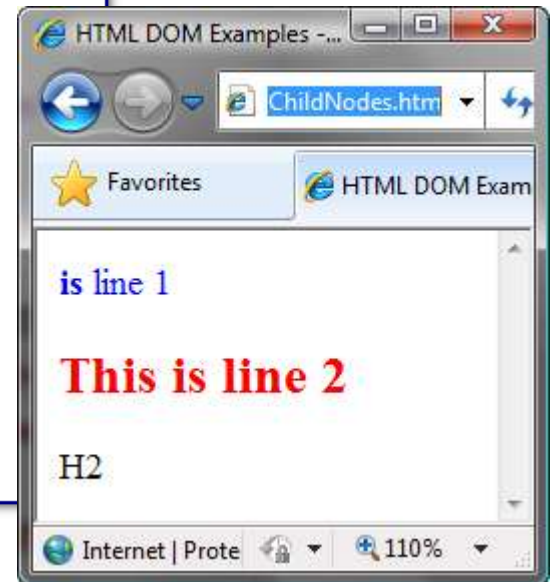
    var inode = document.createElement("i");
    var i_text = document.createTextNode("This is italic text !");
    inode.appendChild(i_text);
    var p2node = document.getElementById("p2");
    document.body.replaceChild(inode, p2node);
  </script>
</body>

```



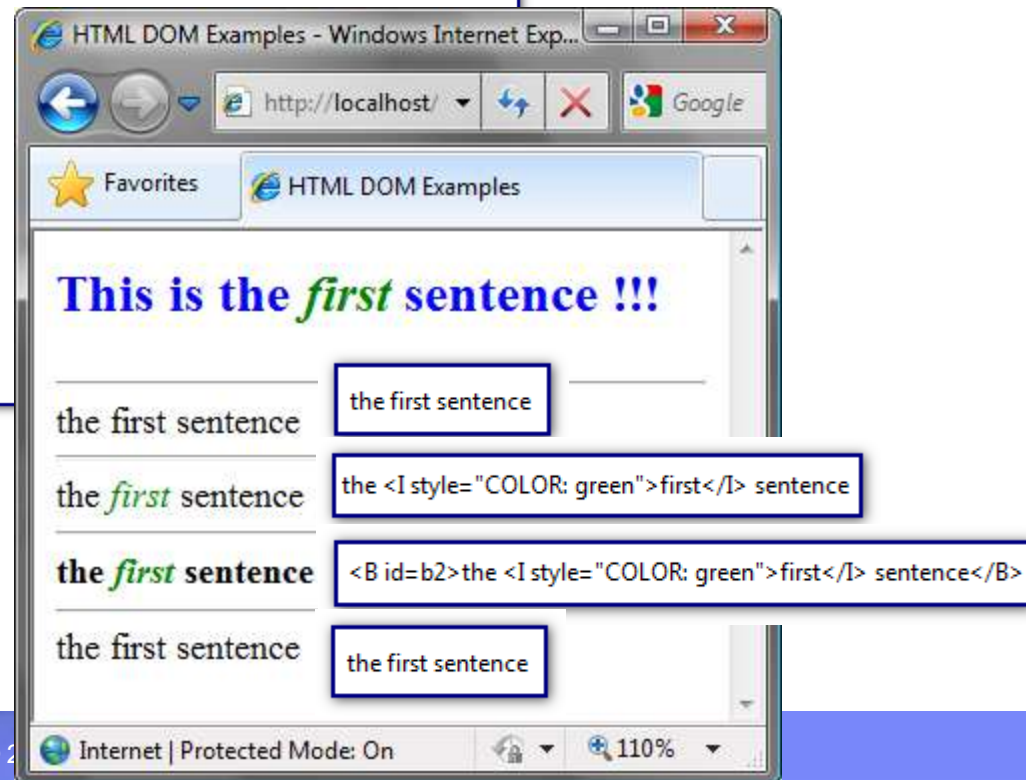
DOM – firstChild , lastChild, nextSibling , hasChildNode()

```
<html>
<head>
  <title>HTML DOM Examples</title>
</head>
<body>
  <p id="p1" style="color:blue"> This <b>is</b> line 1 </p>
  <h2 id="h2" style="color:red">This is line 2</h2>
  <script type="text/javascript">
    var node = document.getElementById("p1");
    if (node.hasChildNodes())
      node.removeChild(node.firstChild);
    else
      document.write("Don't have child !!!");
    var sibling = node.nextSibling;
    document.write(sibling.nodeName);
  </script>
</body>
</html>
```



DOM – innerHTML, innerText, outerText, OuterHTML

```
<body>
  <h2 style="color:blue">
    This is
    <b id="b2"> the <i style="color:green"> first</i> sentence</b>
    !!!
  </h2>
  <script type="text/javascript">
    var node = document.getElementById("b2");
    document.write("<hr/>");
    document.write(node.innerText);
    document.write("<hr/>");
    document.write(node.innerHTML);
    document.write("<hr/>");
    document.write(node.outerHTML);
    document.write("<hr/>");
    document.write(node.outerText);
  </script>
</body>
```



DOM – innerHTML, innerText, outerText, OuterHTML

```

<html>
<head>
  <title>HTML DOM Examples</title>
</head>
<body>
  <h2 id="h2" style="color:blue"></h2>
  <div id="div1"> <div>

  <script type="text/javascript">
    var node1 = document.getElementById("h2");
    node1.innerHTML = "<b>This is <i>header 1</i> !!!</b>";

    var node2 = document.getElementById("div1");
    node2.innerText = "</b>Hello World !!!<b>";
  </script>
</body>
</html>

```



DOM – `node.setAttribute()`, `getAttribute()`, `removeAttribute()`

`node.setAttribute("attributeName", "value")` :

```
var fontNode = document.getElementById("font1");  
fontNode.setAttribute("color", "red");  
fontNode.setAttribute("size", "5");
```

`node.getAttribute("attributeName")`:

```
var font1 = document.getElementById("font1");  
alert(font1.getAttribute("color"));
```

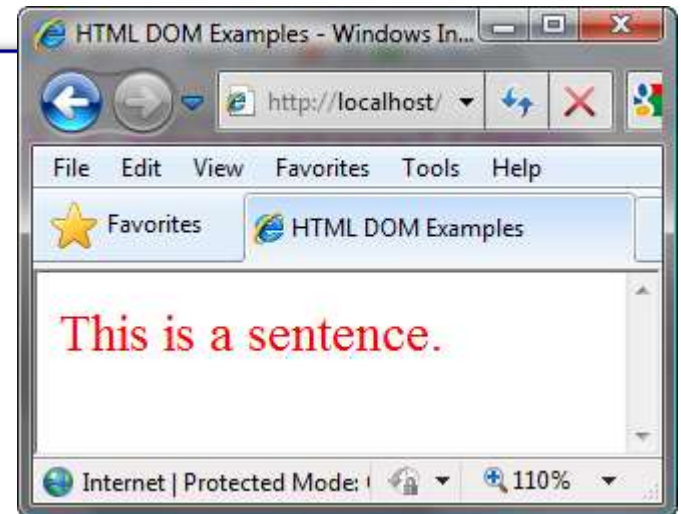
`node.removeAttribute("attributeName")`:

```
var font1 = document.getElementById("font1");  
font1.removeAttribute("color");  
font1.removeAttribute("size");
```

DOM – `node.setAttribute()`, `getAttribute()`, `removeAttribute()`

```
<html>
<head>
  <title>HTML DOM Examples</title>
</head>
<body>
  <font id="font1" >
    This is a sentence.
  </font>

  <script type="text/javascript">
    var fontNode = document.getElementById("font1");
    fontNode.setAttribute("color", "red");
    fontNode.setAttribute("size", "5");
  </script>
</body>
</html>
```

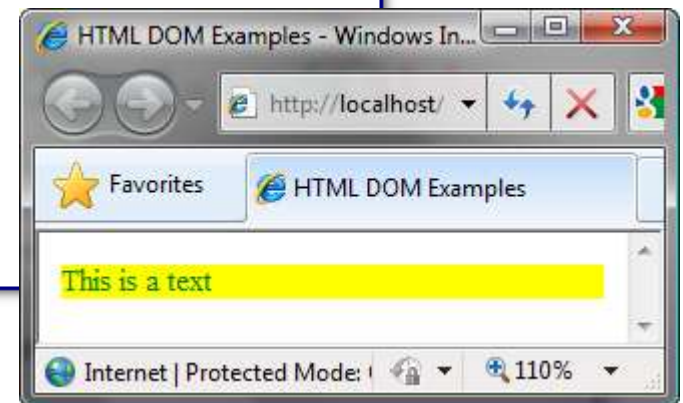


DOM – node.style

node.style Thay đổi định dạng **CSS** của một node

```
<html>
<head>
  <title>HTML DOM Examples</title>
</head>
<body>
<p id="myParagraph" style="color: red;">This is a text</p>

<script language="javascript" >
  var node = document.getElementById("myParagraph");
  node.style.color = "green";
  node.style.fontSize = "14";
  node.style.backgroundColor = "yellow";
</script>
</body>
</html>
```

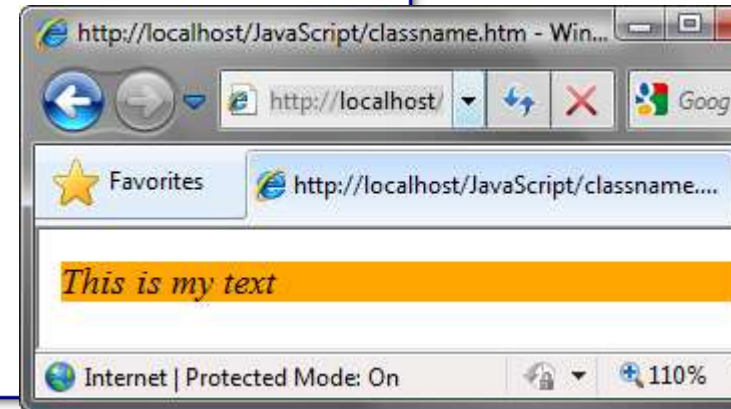


DOM – `node.className`

`node.className`: Thay đổi giá trị của thuộc tính class.

```
<html>
<head>
<style type="text/css">
    .look1{color: black;background-color: yellow; font-style:normal; }
    .look2{background-color: orange; font-style: italic;}
</style>
</head>
<body>
    <p id="p1" class="look1">This is my text</p>

    <script language="javascript" >
        var pNode = document.getElementById("p1");
        pNode.className = "look2";
    </script>
</body>
</html>
```



Trong firefox có thể sử dụng `node.class` thay cho `node.className`

DOM – document Object - node.href

node.href: Thay đổi đường dẫn liên kết.

```
<html>
<head>
  <script language="javascript" >
    function changeSkin()
    {
      document.getElementById("myStyle").href = "style1/style.css";
    }
  </script>
  <link id="myStyle" rel="stylesheet" type="text/css" href="style2/style.css" />
</head>
<body>
  <p class="style1"> Hello world </p>
  <input type="button" onclick="changeSkin()" value="ChangeSkin" />
</body>
</html>
```



DOM trên các trình duyệt

- Đặc điểm:
 - Hỗ trợ và không hỗ trợ
 - Ví dụ:

Method or property	IE 5.5	IE 6	IE 7	IE8rc1 as IE7	IE8rc1 as IE8	FF 2.0	FF 3.0	FF 3.1b2	Saf 3.0 Win	Saf 3.1 Win	iPhone 3G	Chrome 1.0	Opera 9.62	Opera 10a	Konqueror 3.5.7
innerText	yes					no			yes			yes	yes	yes	yes
outerHTML	almost					no			yes			yes	yes	yes	no
outerText	almost					no			yes			yes	yes	yes	no

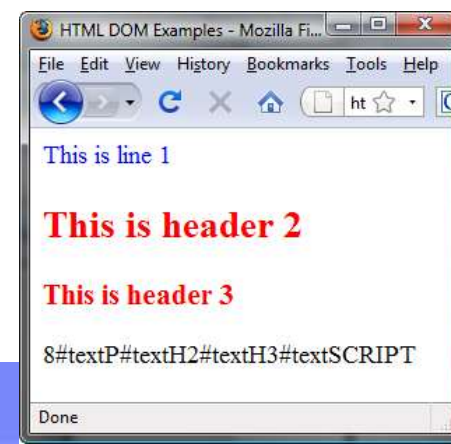
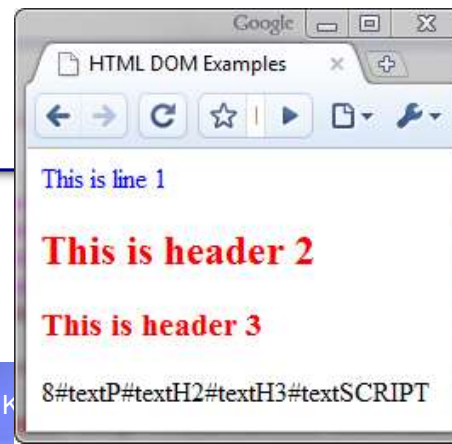
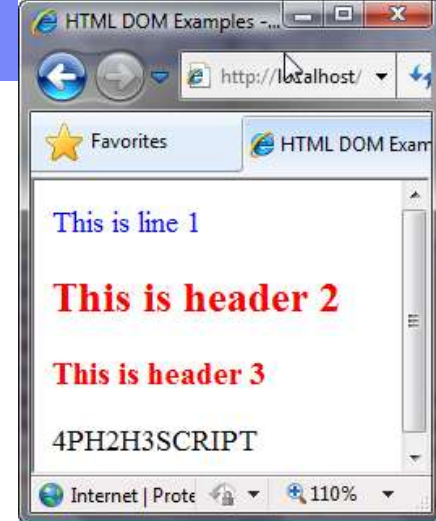
- Hoạt động không giống nhau

DOM trên các trình duyệt

Thuộc tính	IE	Firefox
Thay đổi style cho một đối tượng DOM	<pre>node.style.setAttribute("cssText", styleText);</pre> <p>Ví dụ:</p> <pre>node.style.setAttribute("cssText", "display:none");</pre>	<pre>node.setAttribute("cssText", styleText);</pre> <p>Ví dụ:</p> <pre>node.setAttribute("cssText", "display:none");</pre>
Thay đổi thuộc tính class	<pre>node.setAttribute("className", classname);</pre> <p>Hoặc</p> <pre>node.className = classname;</pre> <p>Ví dụ:</p> <pre>node.setAttribute("className", "tooltip");</pre>	<pre>node.setAttribute("class", classname);</pre> <p>Hoặc</p> <pre>node.class = classname;</pre> <p>Ví dụ:</p> <pre>node.setAttribute("class", "tooltip");</pre>

DOM trên các trình duyệt

```
<html>
<head>
  <title>HTML DOM Examples</title>
</head>
<body>
  <p id="p1" style="color:blue"> This is line 1 </p>
  <h2 id="h2" style="color:red">This is header 2</h2>
  <h3 id="h3" style="color:red">This is header 3</h3>
  <script type="text/javascript">
    var l = document.body.childNodes.length;
    document.write(1);
    for (var i =0; i < l; i++)
    {
      document.write(document.body.childNodes[i].nodeName);
    }
  </script>
</body>
</html>
```

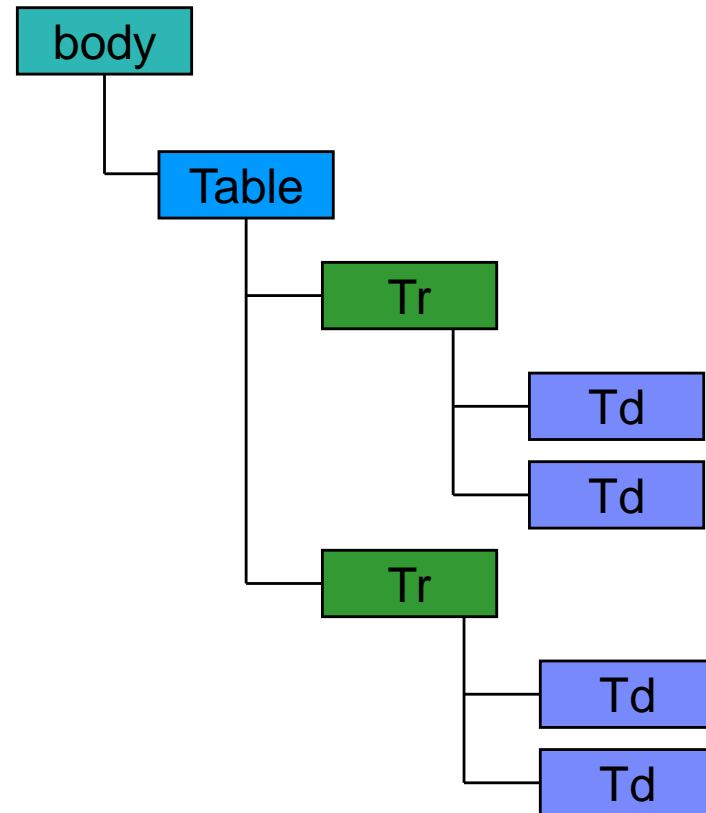


DOM - Table

Cấu trúc của table:

```
<table>
  <tr>
    <td> 12 </td>
    <td> 13 </td>
  </tr>

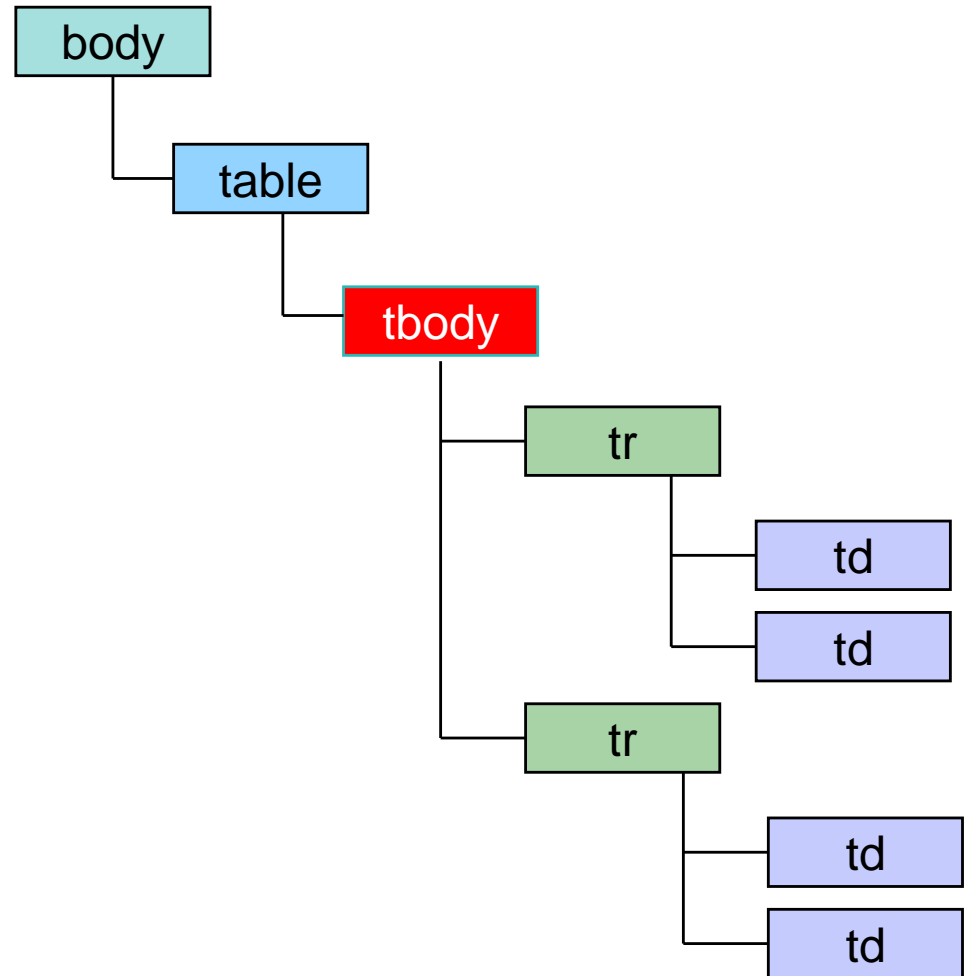
  <tr>
    <td> 21 </td>
    <td> 22 </td>
  </tr>
</table>
```



DOM - Table

- Cấu trúc đầy đủ của table:

```
<table>
<tbody>
  <tr>
    <td> 12 </td>
    <td> 13 </td>
  </tr>
  <tr>
    <td> 21 </td>
    <td> 22 </td>
  </tr>
</tbody>
</table>
```



Khi tạo table bằng DOM thì phải tuân theo cấu trúc đầy đủ.

```

<head>
  <script type="text/javascript" >
    function CreateTable(node, nDong, nCot)
    {
      var tagTable = document.createElement("table");
      tagTable.border = 1;
      var tagTBody = document.createElement("tbody");
      tagTable.appendChild(tagTBody);

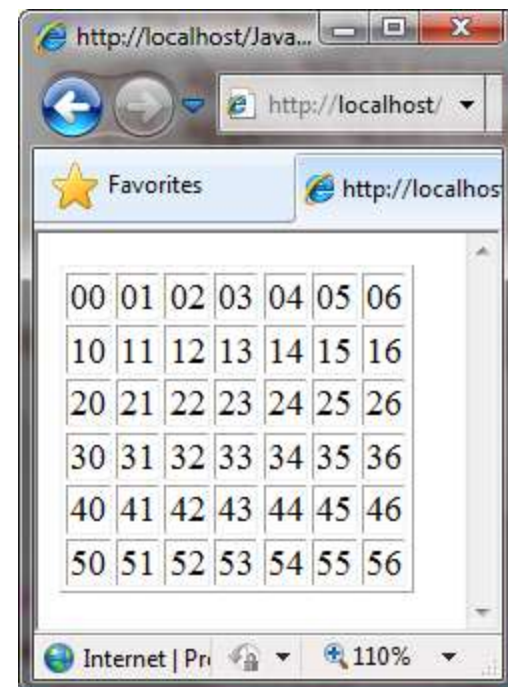
      for (i=0; i<nDong; i++)
      {
        var tagTR = document.createElement("tr");
        for (j=0; j<nCot; j++)
        {
          var tagTD = document.createElement("td");
          var textNode = document.createTextNode(i+" "+j);
          tagTD.appendChild(textNode);
          tagTR.appendChild(tagTD);
        }
        tagTBody.appendChild(tagTR);
      }
      node.appendChild(tagTable);
    }
  </script>
</head>

```

```

<body>
  <div id="p1"> </div>
  <script language="javascript" >
    var node = document.getElementById("p1");
    CreateTable(node, 6, 7);
  </script>
</body>

```



Nội dung

- Giới thiệu DOM và HTML DOM
- Các đối tượng DOM được hỗ trợ trong Javascript
- Một số thuộc tính và phương thức trong HTML DOM
- **HTML DOM và xử lý sự kiện bằng Java script**

Xử lý sự kiện

- Event Object
- Event Handler
- Xử lý sự kiện
- Ví dụ

Xử lý sự kiện – Event Object

- **Events** là các sự kiện xảy ra trên trang Web
- Do người dùng hoặc do hệ thống tạo ra
- Mỗi sự kiện sẽ liên quan đến một **event object**
- Cung cấp thông tin về event
 - Loại event
 - Vị trí con trỏ tại thời điểm xảy ra sự kiện

Xử lý sự kiện

- Event Object
- Event Handler
- Xử lý sự kiện
- Ví dụ

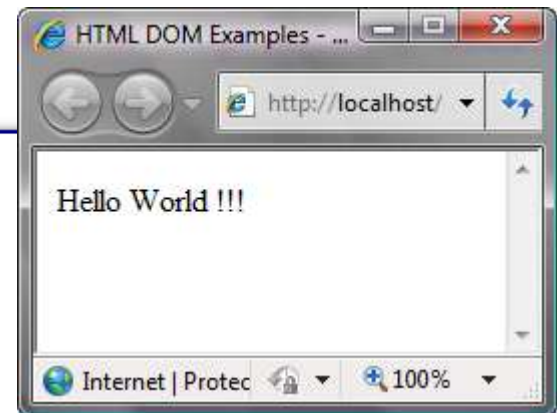
Xử lý sự kiện – Event Handler

- Cú pháp

```
<tag eventName="JavaScript Code">
```

- Ví dụ :

```
<html>
<head>
  <title>HTML DOM Examples</title>
</head>
<body>
  <input type="button" name="abc"
    value="Open new window!!!"
    onclick="window.open ('Hello.htm', 'mywindow', 'status=1,toolbar=1');" />
</body>
</html>
```



Xử lý sự kiện – Event Handler

Danh sách một số Event Handler thường sử dụng

onabort	onload
onblur	onmousedown
onchange	onmousemove
onclick	onmouseout
onerror	onmouseover
onfocus	onmouseup
onkeydown	onreset
onkeyup	onresize
onselect	onsubmit
	onunload

Xử lý sự kiện

- Event Object
- Event Handler
- Xử lý sự kiện
- Ví dụ

Xử lý sự kiện cho các thẻ HTML

- Cú pháp

<TAG **eventHandler = "JavaScript Code">**

- Ví dụ:

```
<body>
```

```
  <INPUT TYPE="button" NAME="Button1"
```

```
    VALUE="OpenSesame!"
```

```
    onClick="window.open('mydoc.html','newWin');">
```

```
</body>
```

- Lưu ý: Dấu "..." và '...'

Xử lý sự kiện

- Xử lý bằng function

```
<SCRIPT LANGUAGE="JavaScript">  
    function greeting() {  
        alert("Welcome to my world");  
    }  
</SCRIPT>  
  
<BODY onLoad="greeting()">  
</BODY>
```

Xử lý sự kiện

- Xử lý bằng thuộc tính :
 - Gán tên hàm xử lý cho 1 object event

`object.eventhandler = functionname;`

```
<SCRIPT LANGUAGE="JavaScript">
    function greeting() {
        alert("Welcome to my world");
    }

    window.onload = greeting;
</SCRIPT>
```

Xử lý sự kiện – Danh sách các sự kiện của Form field

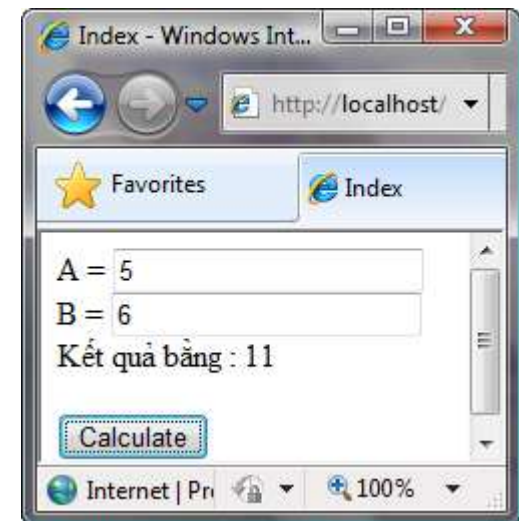
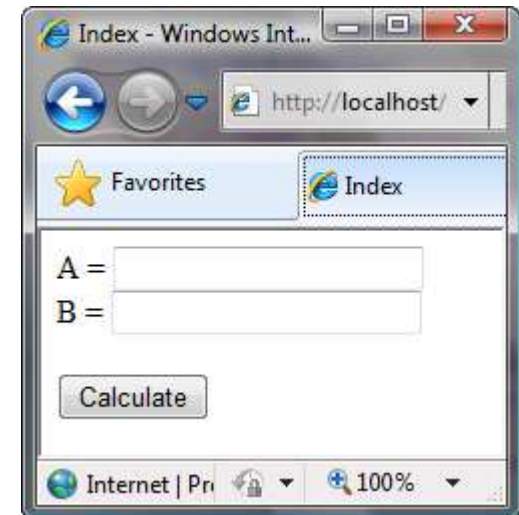
	Blur	Click	Change	Focus	Load	Mouse over	Select	Submit	Unload
Button		x							
Checkbox		x							
Document					x				x
Form								x	
Link		x				x			
Radio		x							
Reset		x							
Selection	x		x	x					
Submit		x							
Text	x		x	x			x		
Textarea	x		x	x			x		

Xử lý sự kiện

- Event Object
- Event Handler
- Xử lý sự kiện
- Ví dụ

Ví dụ: Tính tổng 2 số:

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
<title>Index</title>
<script type="text/javascript" language="javascript">
    function btnTinhTong()
    {
        var a = parseInt(document.getElementsByName("txtSoA")[0].value);
        var b = parseInt(document.getElementsByName("txtSoB")[0].value);
        var tong = a + b;
        document.getElementById("divKetQua").innerHTML = "Kết quả bằng : " + tong;
    }
</script>
</head>
<body>
<form id="form1" name="form1" method="get" action="">
    A = <input type="text" name="txtSoA" id="txtSoA" /> <br/>
    B = <input type="text" name="txtSoB" id="txtSoB" /> <br/>
    <div id="divKetQua" name="divKetQua" > </div> <br/>
    <input type="button" onclick ="btnTinhTong();" value="Calculate" />
</form>
</body>
</html>
```



Ví dụ: Mở một cửa sổ bằng java script

```
<html>
<head>
<script>
var newwindow;
function showWindow()
{
    newwindow = window.open("", "new window", "toolbar=no,width=300,height=25,top=300");
    newwindow.document.open();
    newwindow.document.write("<html><title>New Window</title><body><em>This is a text window</em></body></html>");
    newwindow.document.close();
}
function closeWindow() { newwindow.close(); }
</script>
</head>
<body>
    <input type="button" value="Open" onClick="return showWindow();"/>
    <input type="button" value="Close" onClick="return closeWindow();"/>
</body>
</html>
```

Ví dụ: Mở một cửa sổ và nhận giá trị trả về

Mainwindow.html

```
<html>
<head>
<script language="javascript">
    var newwindow;
    function showWindow()
    {
        newwindow = window.open("popup_window.html", "newwindow", "toolbar=no
    }
    function update(content)
    {
        window.document.getElementById("txtContent").value = content;
    }
</script>
</head>
<body>
    <input id="txtContent" type="text"/>
    <input type="button" value="Show" onClick="showWindow();"/>
</body>
</html>
```

Ví dụ: Mở một cửa sổ và nhận giá trị trả về (tt)

popup_window.html

```
<html>
<head>
<script language="javascript">
    function closeWindow() { window.close(); }
    function OK(txtTextBox)
    {
        var content = txtTextBox.value;
        window.opener.update(content);
        window.close();
    }
</script>
</head>
<body>
    <input id="txtValue" type="text"/>
    <input type="button" value="OK" onClick="OK( this );"/>
    <input type="button" value="Close" onClick="closeWindow();"/>
</body>
</html>
```

Ví dụ: Java script và CSS

```
<html>
<head>
<script language="javascript">
    function setStyle( object, styleText )
    {
        if( object.style.setAttribute )
            object.style.setAttribute("cssText", styleText );
        else
            object.setAttribute("style", styleText );
    }

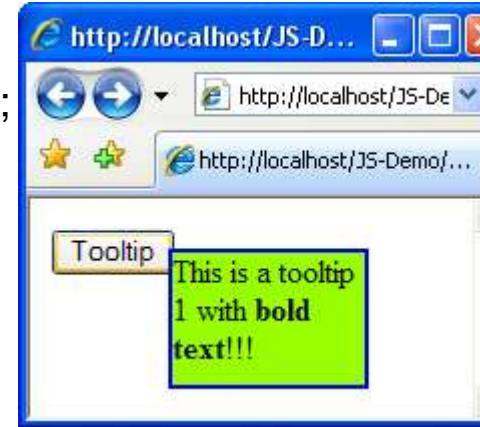
    function showTooltip()
    {
        setStyle (document.getElementById("divTooltip"), "display:visible;");
    }
    function hideTooltip()
    {
        setStyle (document.getElementById("divTooltip"), "display:none;");
    }
</script>
</head>
<body>
    <input type="button" value="Tooltip" onMouseOver="showTooltip(this);" onMouseOut="hideTooltip(this);" />
    <div id="divTooltip" class="tooltip">
        This is a tooltip !!!
    </div>
</body>
</html>
```

Ví dụ: Java script và CSS (2)

```
<html>
<head>
<script language="javascript">
    function setClass( object, classname)
    {
        object.className = classname; //work in IE
        object.class = classname; //work in FF
    }
    function showTooltip()
    {
        setClass (document.getElementById("divTooltip2"), "mouseover");
    }
    function hideTooltip()
    {
        setClass (document.getElementById("divTooltip2"), "mouseout");
    }
</script>
<style>
    .mouseout {background-color:#FF0;}
    .mouseover {background-color:#06C;}
</style>
</head>
<body>
    <div id="divTooltip1" >
        This is a tooltip 2!!!
    </div>
</body>
</html>
```

Ví dụ: Tooltip với java script

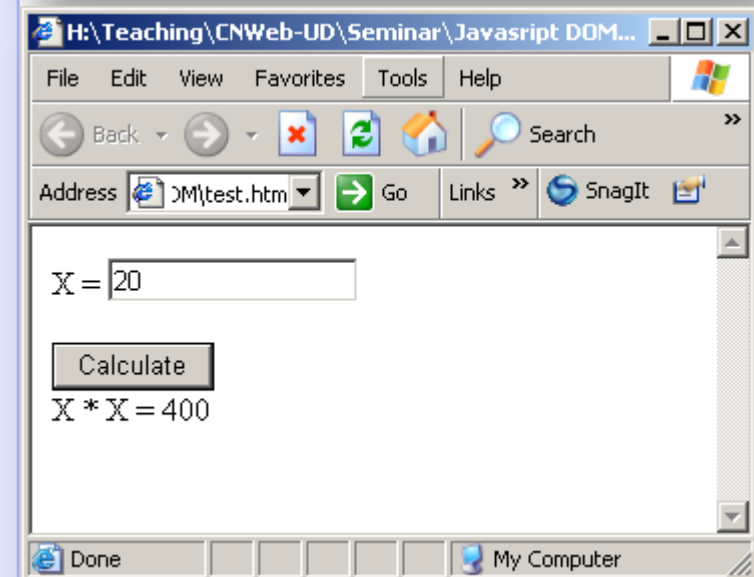
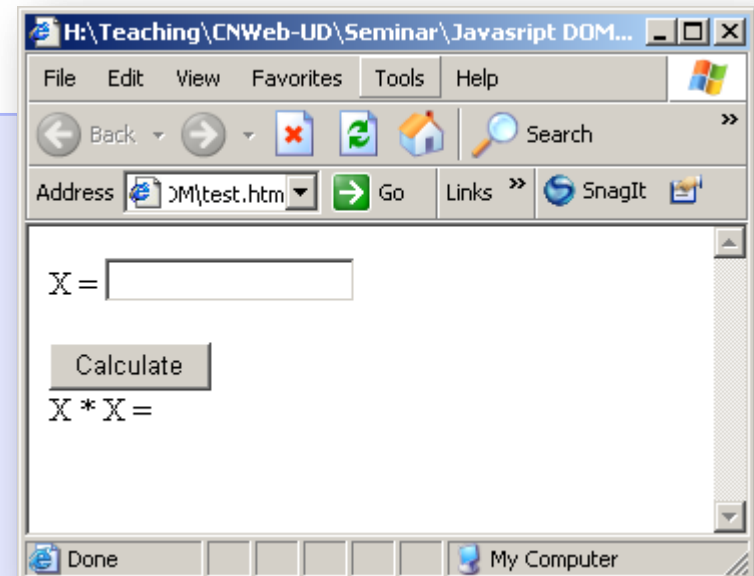
```
<html> <head>
<script language="javascript">
    function setStyle( object, styleText ) {
        if( object.style.setAttribute )
            object.style.setAttribute("cssText", styleText );
        else
            object.setAttribute("style", styleText );
    }
    function showTooltip(e) {
        var evt = e || window.event;
        var mouseX = evt.clientX + document.body.scrollLeft;
        var mouseY = evt.clientY + document.body.scrollTop;;
        setStyle (document.getElementById("divTooltip"), "display:visible;left:" + mouseX
    }
    function hideTooltip() { setStyle (document.getElementById("divTooltip"), "display:none;");
</script>
<style> div.tooltip { position:absolute; height:70px; width:100px; background-color:#9F0; border:#0
</head>
<body>
    <input id="btn1" type="button" value="Tooltip" onmouseout="hideTooltip();" />
    <div id="divTooltip" class="tooltip" style="display:none;">
        This is a tooltip 1 with <b>bold text</b>!!!
    </div>
    <script language="javascript">
        document.getElementById("btn1").onmousemove = showTooltip; // need to assign
    </script>
</body> </html>
```



Ví dụ: onclick Event

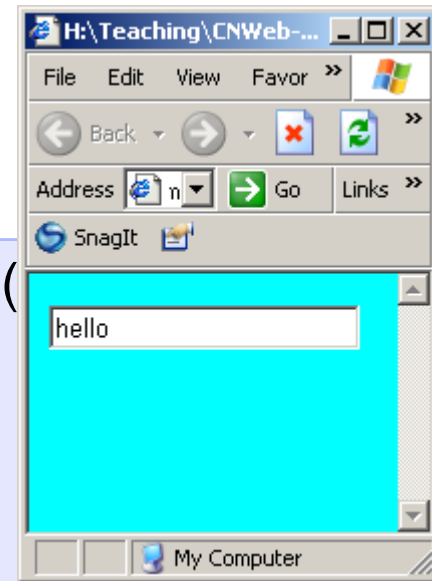
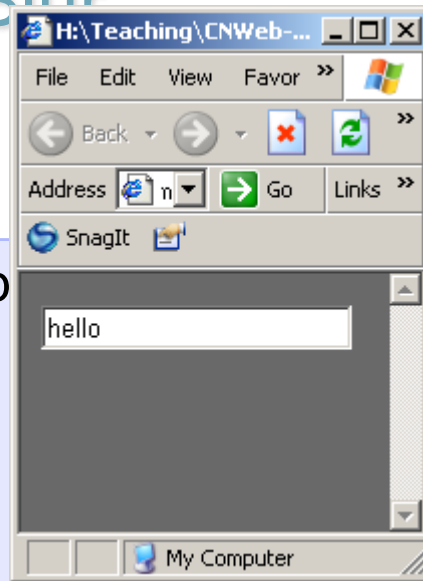
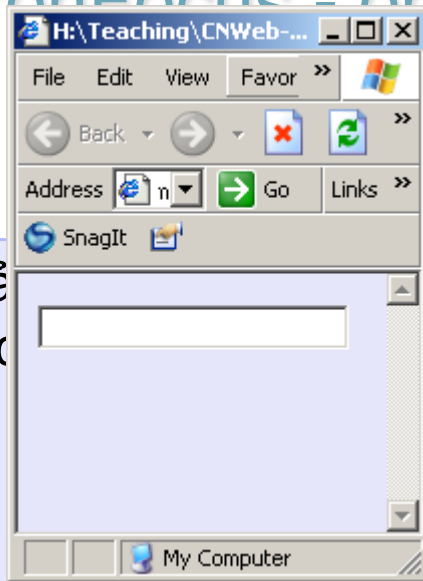
```
<SCRIPT LANGUAGE="JavaScript">
    function compute(frm){
        var x = frm.expr.value;
        result.innerHTML = x*x;
    }
</SCRIPT>

<FORM>
    X = <INPUT TYPE="text" NAME="expr" SIZE=15>
    <BR><BR>
    <INPUT TYPE="button" VALUE="Calculate"
        ONCLICK="compute(this.form)">
    <BR>
    X * X = <SPAN ID="result"></SPAN>
</FORM>
```



Ví dụ: onFocus - onBlur

- Xử lý onFocus - onBlur
- Ví dụ



```
<BODY BGCOLOR="lavender">
```

```
<FORM>
```

```
<INPUT type="text" name="myTextbox"
```

```
onblur="(document.bgColor='aqua')"
```

```
onfocus="(document.bgColor='dimgray')">
```

```
</FORM>
```

```
</BODY>
```

Ví dụ: onMouseOver - onMouseOut

```
<script language = "javascript">
```

```
function showLink(num) {
```

```
    if (num==1)
```

```
        document.forms[0].elements[0].value=
```

```
            "You have selected Aptech";
```

```
    } else {
```

```
        document.forms[0].elements[0].value="";
```

```
    }
```

```
}
```

```
</script>
```

```
<form> <input type=text size=60> </form>
```

```
<a href="#" onMouseOver="showLink(1)" onMouseOut="showLink(0)"
```

```
Aptech</a>
```

