

FPT POLYTECHNIC



Bài 5: MÔ HÌNH DOM VÀ CSS

- Mô hình lập trình
- Mô hình lập trình hướng đối tượng
- Thuộc tính, phương thức và lớp
- Tổ chức
- Thêm thuộc tính và phương thức vào đối tượng
- Tổ hợp, tổ chức đối tượng tập
- Các thao tác với đối tượng trong lập
- Browser Object Model

- Giới thiệu về mô hình Document Object Model
- Giới thiệu về HTML DOM
- Cấu trúc DOM
- Thuộc tính của node
- Phương thức của node
 - Truy cập đến node
 - Thêm node
 - Xóa node
- Truy cập và thay đổi style của các element
- Ví dụ về các trình duyệt khác nhau

- DOM là m t chu n c nh ngh a b i W3C (World Wide Web Consortium) có th truy c p và thao tác v i các tài li u nh html hay xml b ng các ngôn ng l p trình nh Javascript, VB...

The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents (nh ngh a b i W3C)

- DOM c chia thành 3 m c
 - Core DOM: Tiêu chu n cho b t k tài li u có c u trúc nào
 - XML DOM: Tiêu chu n cho tài li u XML
 - HTML DOM: Tiêu chu n cho tài li u HTML

- HTML DOM **nh** nghĩa là các **l**ưu **l**ưu và thu **l**ưu tính **l**ưu **l**ưu các thành **l**ưu HTML và ph **l**ưu **l**ưu truy **l**ưu **l**ưu chúng
- Hay: HTML DOM là chu **l**ưu **l**ưu, thay **l**ưu, thêm, xóa các thành **l**ưu HTML

Truy **lưu trang web <http://www.w3schools.com/html/dom/default.asp/> tìm hi **l**ưu thêm v **l**ưu HTML DOM**

- Tất cả các thành phần trong tài liệu HTML đều có biểu diễn bằng một node
 - Toàn bộ tài liệu là document node
 - Tất cả các thành phần của HTML đều là element node
 - Văn bản trong thành phần HTML là text node
 - Tất cả các thuộc tính là attribute node
 - Chú thích là comment node

```
<html >
  <head>
    <title> Hi</title>
  </head>
  <body>
    <p> Hello</p>
    <a href = "http://google.com">Link</a>
  </body>
</html>
```

- Node gốc là <html>
- <html> có hai node con là <head> và <body>
- <head> có một node con là <title>
- <body> chứa hai node con là <p> và <a>

- DOM trình bày tài liệu HTML theo cấu trúc hình cây

```
<html >
```

```
<head>
```

```
<title>Hi</title>
```

```
</head>
```

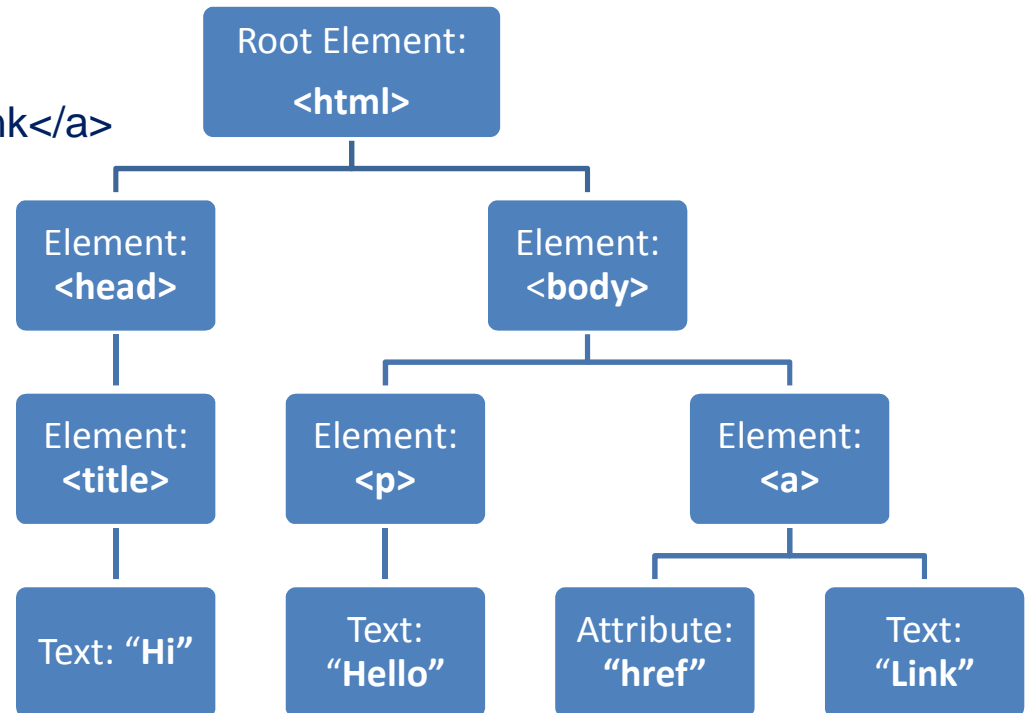
```
<body>
```

```
<p>Hello</p>
```

```
<a href = "http://google.com">Link</a>
```

```
</body>
```

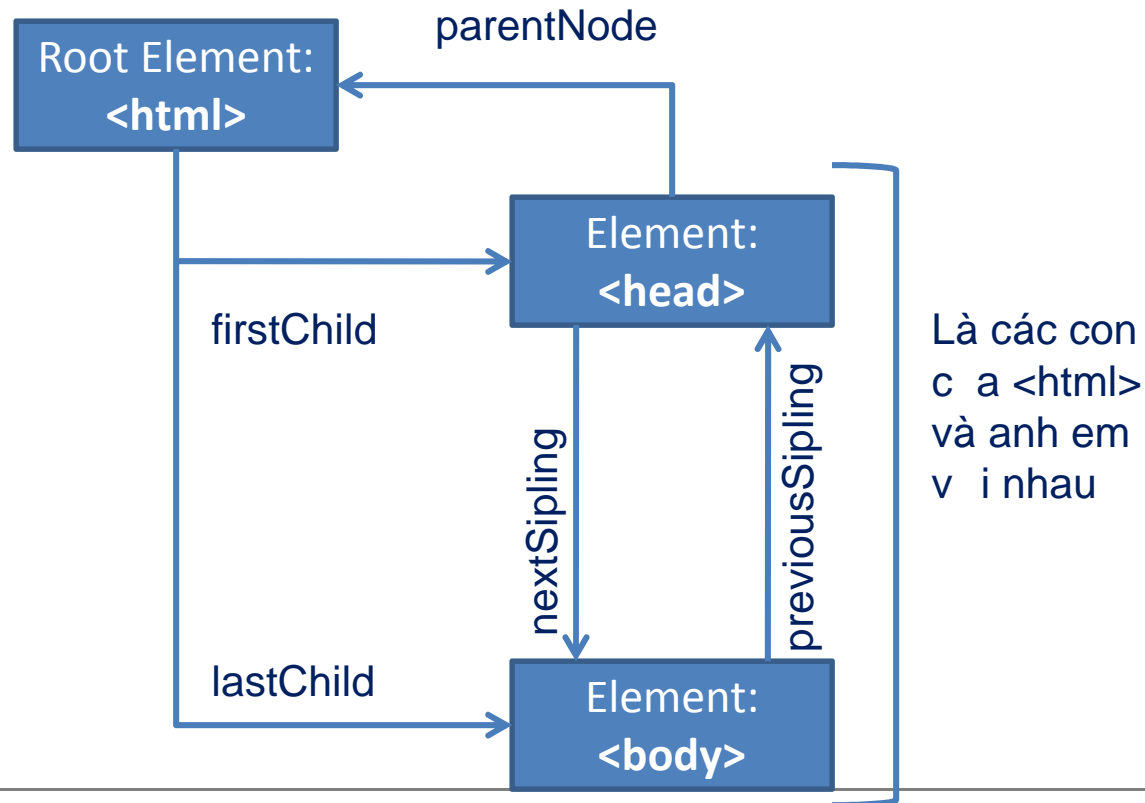
```
</html>
```



- Giữa các node có mối quan hệ cha, con (parent, children) và anh em (siblings)
 - Node cha có các node con, các node con cùng cấp bậc gọi là anh em
- Trên cây, node ưu tiên gọi là root (gốc)
- Tất cả các node, ngoại trừ root chỉ có một node cha
- Một node có thể có nhiều node con hoặc không có node con nào
- Node lá (leaf) là node không có node con
- Những node anh em (siblings) với nhau là node có cùng một node cha

- `<html>` không có node cha, `<html>` là node gốc
- `<html>` có hai node con là `<head>` và `<body>`, `<head>` và `<body>` là anh em
- `<head>` có một node con là `<title>`
- `<title>` có một node con là text node "hi"
- `<body>` có hai node con là `<p>` và `<a>`, `<p>` và `<a>` là anh em
- Node cha của node `<head>` và `<body>` là node `<html>`
- Node cha của text node "Hello" là node `<p>`
- Text node "Hi", "Hello", "Link" và attribute node "href" đều là node lá

- <head> và <body> là anh em, trong đó <head> là con c c a <html> còn <body> là con út c a <html>
- <p> và <a> là anh em, trong đó <p> là con c c a <body> còn <a> là con út c a <body>



- DOM nh ngh a các thu c tính và các ph ng th c cho các node h tr cho vi c l p trình
- Thu c tính nh ngh a các c tính cho node
- Ph ng th c th c hi n các thao tác v i node
 - Truy c p n node
 - Thêm node con cho node
 - Xóa node con

Thuộc tính	Giải thích
<i>x là đối tượng node</i>	
x.innerHTML	Giá trị văn bản của x
x.nodeName	Tên của x
x.nodeValue	Giá trị của x
x.nodeType	Kiểu của Node
x.parentNode	Node cha của x
x.childNodes	Các node con của x
x.attributes	Các node thuộc tính của x

c thêm

http://www.w3schools.com/html/dom/dom_nodes_info.asp

tra tra cứu thêm về các node

Demo truy cập thuộc tính của Node

- Có thể xem hoc thi t l p giá tr cho các thu c tính b ng cách truy c p n thu c tính

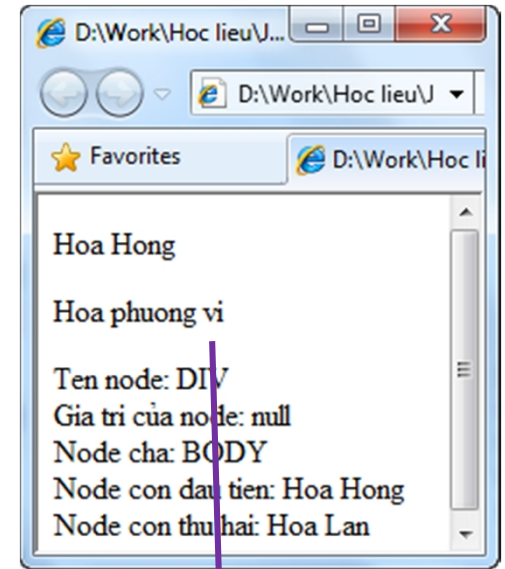
```

<body>
  <div id = "dHoa">
    <p>Hoa Hong</p>
    <p>Hoa Lan</p>
  </div>
  <script type = "text/javascript">
    var pHoa = document.getElementById("dHoa")
    document.write("Ten node: " + pHoa.nodeName + "<br>");
    document.write("Gia tri c a node: " + pHoa.nodeValue + "<br>");
    document.write("Node cha: " + pHoa.parentNode.nodeName + "<br>");
    document.write("Node con dau tien: " + pHoa.firstChild.innerText + "<br>");
    document.write("Node con thu hai: " +
pHoa.childNodes[1].childNodes[0].nodeValue + "<br>");
    pHoa.childNodes[1].innerText = "Hoa phuong vi";
  </script>
</body>

```

Thay i n i dung trong th <p>

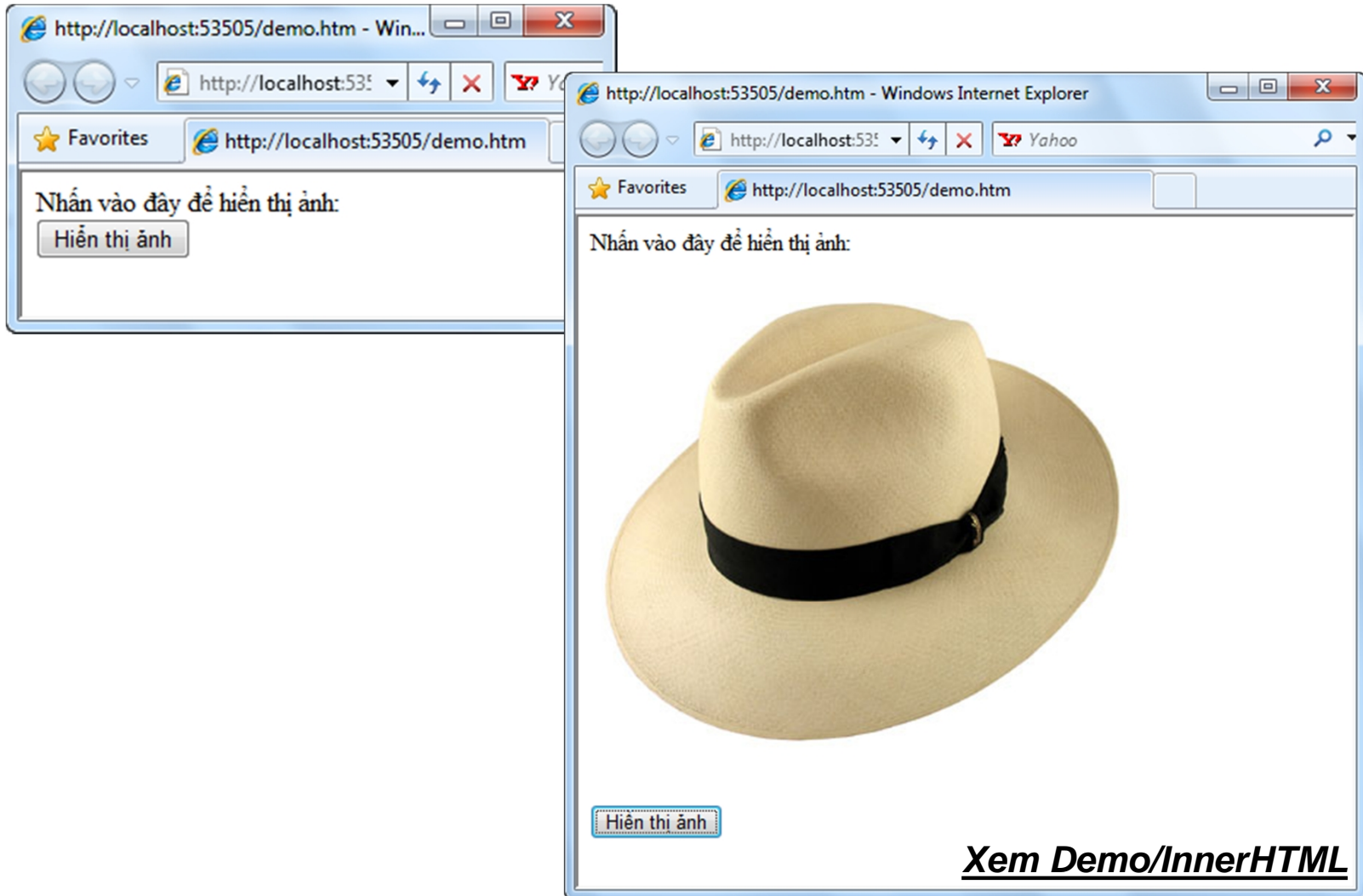
Xem Demo Truy cap cac thuoc tinh c a node



Thuộc tính	Của node	Có giá trị là
nodeName	element	Tên thẻ HTML tương ứng
	attribute	Tên thuộc tính
	text	#text
	document	#document
nodeValue	element	null
	attribute	Giá trị của thuộc tính
	text	Văn bản
nodeType	element	1
	attribute	2
	text	3
	comment	8
	document	9


```
<head>
  <script type="text/javascript">
    function hienThi() {
      var node = document.getElementById("anhDiv");
      node.innerHTML = "<img src='mu.jpg'>";
    }
  </script>
</head>
<body>
  Nh  n vào  ây  hi  n th  nh:
  <div id="anhDiv"></div>
  <input type="button" value="Hi  n th  nh" onclick="hienThi()"/>
</body>
</html>
```

Demo về tính innerHTML

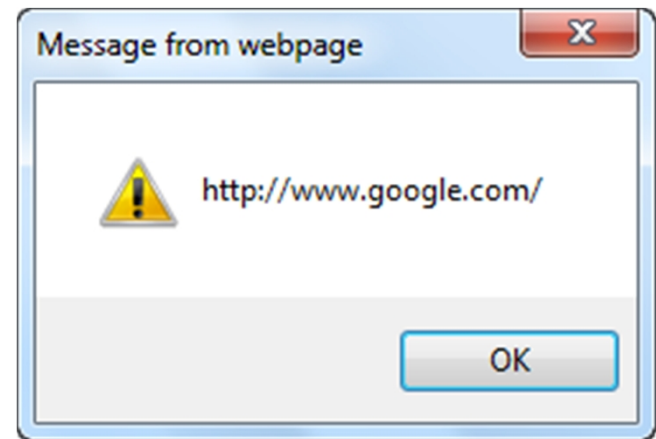


Thuộc tính	Giải thích
<i>x là đối tượng node</i>	
<code>x.getElementById(<i>id</i>)</code>	Trả về thành phần có id xác định
<code>x.getElementsByTagName(<i>name</i>)</code>	Trả về tất cả các thành phần với tên thẻ xác định
<code>x.appendChild(<i>node</i>)</code>	Thêm node con vào node x
<code>x.removeChild(<i>node</i>)</code>	Xóa node con của node x

- Có thể truy cập các node bằng 3 cách sau:
 - Sử dụng phương thức getElementById(id)
 - Sử dụng phương thức getElementsByName(name)
 - Sử dụng phương thức getElementsByTagName(name)
 - Sử dụng mối quan hệ giữa các node với nhau

Demo sử dụng getElementById(id)

```
<html >
  <head>
    <title>Hi</title>
  </head>
  <body>
    <p>Hello</p>
    <a id = "link" href = "http://www.google.com">Link</a>
    <script type = "text/javascript" >
      var linkNode = document.getElementById("link");
      alert(linkNode.href);
    </script>
  </body>
</html>
```



- Giả sử có mã nguồn ví dụ như sau

```
<html >
  <head>
    <title>Hi</title>
  </head>
  <body >
    <p>Hello</p>
    <script type = "text/javascript" >
      var linkNode = document.getElementById("link");
      alert(linkNode.href);
    </script>
    <a id = "link" href = "http://www.google.com">Link</a>
  </body>
</html>
```

- Tại sao không hiển thị hình ảnh như thế này????

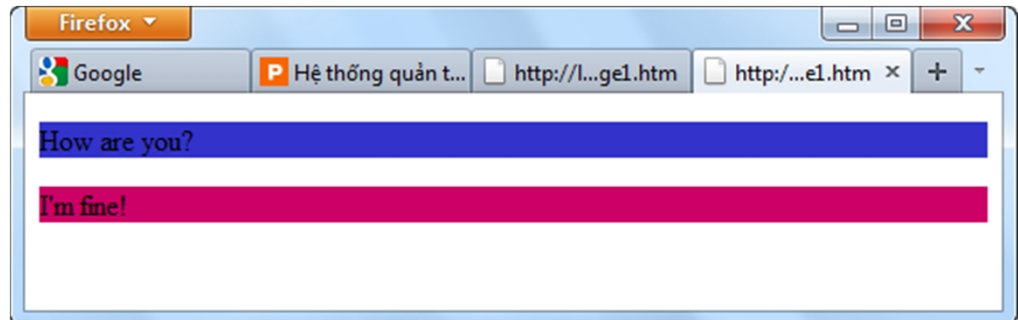
Trắc Nghiệm: Tại vì Browser làm việc theo cách thông dịch. Tác là dịch từng dòng mã, khi nào như JavaScript `document.getElementById("link")` thì chưa có Id nào tên là "Link" nên không có node nào trả về cho biến `linkNode`

- Hãy dự đoán kết quả của mã sau 😊

```
<html >
  <head>
    <title>Hi</title>
    <script type = "text/javascript" >
      function checkHref() {
        var linkNode = document.getElementById("link");
        alert(linkNode.href);
      }
    </script>
  </head>
  <body onload = "checkHref()">
    <p>Hello</p>
    <a id = "link" href = "http://www.google.com">Link</a>
  </body>
</html>
```

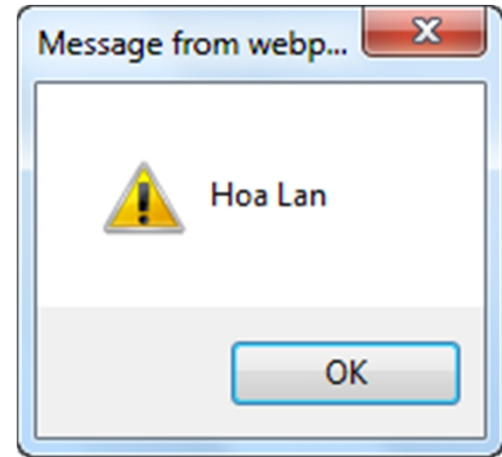


```
<html>
<head>
  <title></title>
</head>
<body>
  <p>How are you?</p>
  <p>I'm fine!</p>
  <script type = "text/javascript">
    var p = document.getElementsByTagName("p");
    p[0].style.background = "#3333CC";
    p[1].style.background = "#CC0066";
  </script>
</body>
</html>
```



■ Mã cho

```
<html >
<body>
  <div id = "hoa">
    <p>Hoa Hong</p>
    <p>Hoa Lan</p>
    <p>Hoa Dao</p>
  </div>
  <script type = "text/javascript">
    var divHoa = document.getElementById("hoa");
    var pHoaHong = divHoa.firstChild;
    alert(pHoaHong.nextSibling.childNodes[0].nodeValue);
  </script>
</body>
</html>
```



- pHoa trên IE có 3 node con

+	behaviorUrns	{...}	DispHTML
...	canHaveChildren	true	Boolean
...	canHaveHTML	true	Boolean
-	childNodes	{...}	DispDOM
+	[Methods]		
-	[Expandos]		
+	[0]	{...}	DispHTML
+	[1]	{...}	DispHTML
+	[2]	{...}	DispHTML
...	constructor	{...}	Object
...	length	3	Number

pHoa có 3 node con

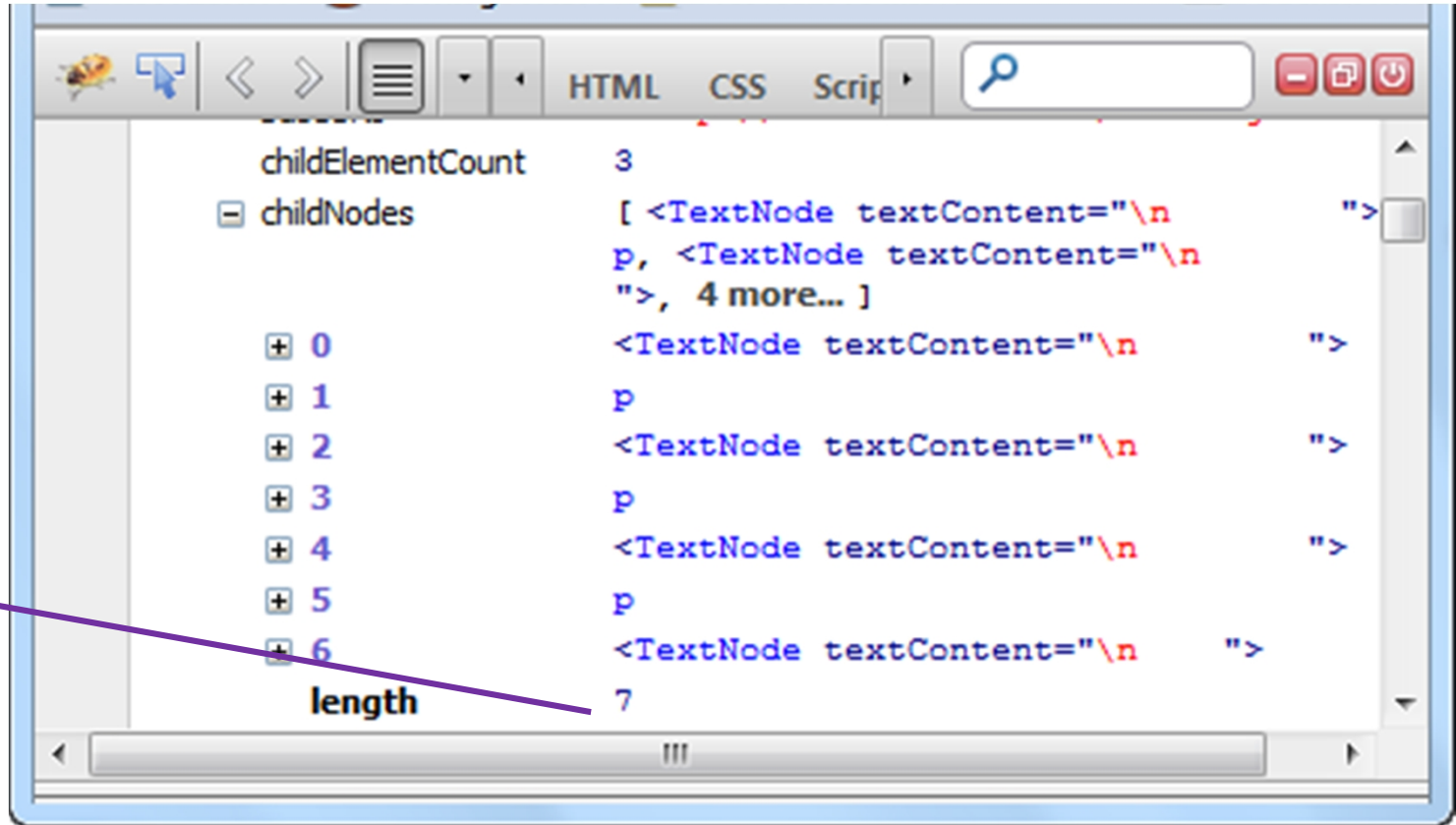
■ Mã cho

```
<html>
<body>
  <div id = "hoa">
    <p>Hoa Hong</p>
    <p>Hoa Lan</p>
    <p>Hoa Dao</p>
  </div>
  <script type = "text/javascript">
    var divHoa = document.getElementById("hoa");
    var pHoaHong = divHoa.firstElementChild;
    alert(pHoaHong.nextSibling.nextSibling.childNodes[0].nodeValue);
  </script>
</body>
</html>
```

Mọi Browser đều nhúng a
phần ng th c riêng
truy c p n các ph n t

Cấu trúc DOM của m i
Browser là khác nhau

- pHoa trên fireFox có 7 node con



pHoa có 7
node con

Mỗi browser có cách phân tích khác nhau theo mô hình DOM khác nhau

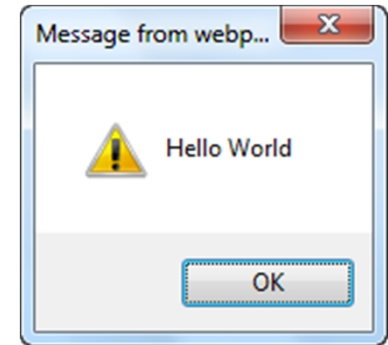
- Sử dụng phương thức `createElement` và `appendChild` thêm các node element vào tài liệu


```
<html >
<head></head>
<body>
<script type="text/javascript">
    var newElement = document.createElement("p");
    document.body.appendChild(newElement);
    newElement.appendChild(document.createTextNode("Hello World"));
</script>
</body>
</html>
```

- Dùng ph ng th c `setAttribute` c a node thi t l p thu c tính cho chính node ó

```
<head></head>
<body id = "bd">
<script type="text/javascript">
    var newElement = document.createElement("p");
    newElement.setAttribute("id", "newElement");
    document.body.appendChild(newElement);
    newElement.appendChild(document.createTextNode("Hello World"));

    var ne = document.getElementById("newElement");
    alert(ne.innerText);
</script></body></html>
```



Chú ý: o n mã này dành cho 

- Sử dụng phương thức `removeChild(nodeId)` của node xóa các node element của node

```
<html >
<head>
</head>
<body >
  <p id="pHello">Hello</p>
  <p id="pHi">Hi</p>
<script type="text/javascript">

document.body.removeChild(pHi);
</script></body></html>
```

o n mã này dành cho



```
<html >
<head>
</head>
<body >
  <p id="pHello">Hello</p>
  <p id="pHi">Hi</p>
<script type="text/javascript">
var pHi = document.getElementById("pHi");
document.body.removeChild(pHi);
</script></body></html>
```

o n mã này dành cho



- CSS định nghĩa màu sắc, font, layout... cho trang web
- CSS bao gồm nhiều tính chất, mỗi tính chất có một giá trị nhất định

font-family:arial;

- Selector cho biết thành phần nào sẽ được áp dụng

selector { property:value;}

- Có 3 loại Selector

- Thẻ `h1 { font-family:arial; }`
- Lớp `.tenClass { font-family:arial; }`
- ID `#tenId { font-family:arial; }`

- Có thể sử dụng JavaScript thay thế style cho trang web
- JavaScript cung cấp khả năng style cho mọi element của trang web thay thế style của trang web
- Sử dụng thuộc tính style của mọi element truy cập và thay đổi style

```
var hTieuDe = document.getElementById("hTieuDe");
hTieuDe.style.fontFamily = "arial"
```

- Khả năng style trong JavaScript có các thuộc tính tương ứng với các thuộc tính của CSS
 - Chú ý: Với các thuộc tính CSS có dấu gạch ngang thì sẽ chuyển thành dấu gạch ngang và viết hoa chữ cái đầu sau dấu gạch ngang (font-family → fontFamily)

- Có th thi t l p style cho các thành ph n b ng
 - ID
 - Th
 - Class

■ Các b ã c th ã c hi ñ

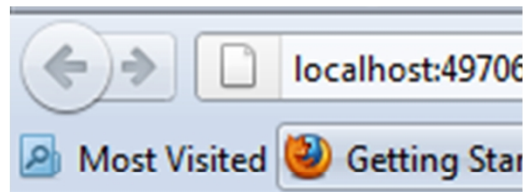
- B ã c 1. Dùng ID truy c p ñ n các element
- B ã c 2. S ã d ñ ng thu c tính style thi t l p style cho element ó

```
var hTieuDe = document.getElementById("hTieuDe");  
hTieuDe.style.fontFamily = "arial"
```



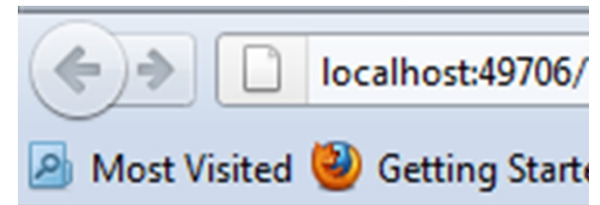
Thiet lap style bang ID

Xanh Do



Thiet lap style bang ID

Xanh Do



Thiet lap style bang ID

Xanh Do

Xem CSS

```

<html >
<head>
  <script type="text/javascript">
    function xanhButtonClick() {
      var pText = document.getElementById("pText");
      pText.style.color = "blue";
    }
    function doButtonClick() {
      var pText = document.getElementById("pText");
      pText.style.color = "red";
    }
  </script>
</head>
<body>
<p id="pText">Thiet lap style bang ID</p>
<button onclick="xanhButtonClick();">Xanh</button>
<button onclick="doButtonClick();">Do</button>
</body>
</html>
  
```

■ V n n y sinh:

- Mu n thay i style cho nhi u ph n t

```
<p id = "pText">Thiet lap style cho nhieu thanh phan</p>  
<p id = "p1">Thanh phan 1</p>  
<p id = "p2">Thanh phan 2</p>  
<p id = "p3">Thanh phan 3</p>
```

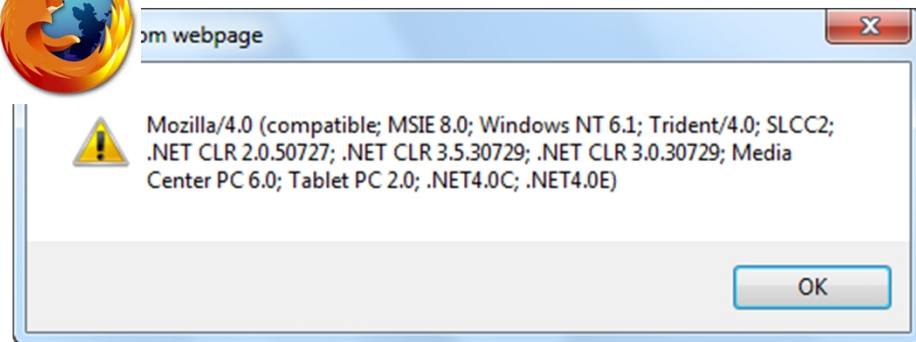
■ Gi i pháp:

- S d ng ph ng th c getElementByName ho c getElementByTagName l y m t nhóm các ph n t
- S d ng vòng For duy t qua các ph n t

```
var pAr = document.getElementsByTagName("p");  
for (var i = 0; i < pAr.length; i++) {  
    pAr[i].style.color = "blue";  
}
```

- Sử dụng thuộc tính userAgent của navigator

`alert(navigator.userAgent)`



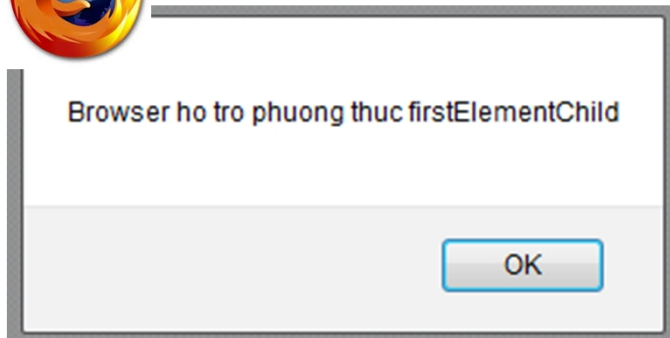
*Tạo ra các phiên bản khác nhau cho các **browser***

■ V ận hành sinh:

- Có r ất nhi ều trình duyệt
- M ỗi trình duyệt t ại có nhi ều phiên b ản
- M ỗi trình duyệt t ại l ập h ệ tr ường các m ạng khác nhau
- ➔ ki ểm tra t ất c ả trình duyệt, phiên b ản c ủa trình duyệt là i ều không th
- ➔ Thu ộc tính userAgent c ũng có th ể tr ả v ề thông tin sai
- ➔ C ần có m ột gi ới pháp khác

- Sử dụng cách th bi t trình duy t h tr ph ng th c gì

```
if (typeof document.body.firstChild != "undefined") {  
    alert("Browser hỗ trợ phương thức firstElementChild");  
} else {  
    alert("Browser không hỗ trợ phương thức firstElementChild");  
}
```



- T o mã JavaScript ch y t t trên t t c các version cũ a t t c các trình duy t là i u không th
- Thi t l p m t gi i h n h p lý các trình duy t và version h tr
- Gi i h n càng h p thì càng ít khách hàng truy c p c vào website
- i v i browser không h tr JavaScript ho c b disable JavaScript.
 - Dùng th <noscript></noscript>
 - Dùng th <!--><-->

- Có th dùng x lý s ki n thay i l i n i dung, c u trúc trang web

```
<html>
<head>
<script type="text/javascript">
    function displayDate() {
        document.getElementById("demo").innerHTML = Date();
    }
</script>
</head>
<body>
    <h1 id ="demo">My First Web Page</h1>
    <button type="button" onclick="displayDate()">Display Date</button>
</body>
</html>
```

c thêm Ch ng 11 bi t thêm v x lý s ki n

- DOM là mô hình chuẩn của nh ngh a b i W3C (World Wide Web Consortium) có th truy c p và thao tác v i các tài li u nh html hay xml b ng các ngôn ng l p trình nh Javascript, VB...
- DOM c chia làm 3 m c CoreDOM, HTML DOM, và XML DOM
- HTML DOM nh ngh a các i t ng và thu c tính c a t t c các thành ph n HTML và ph ng th c truy c p n chúng
- T t c các thành ph n trong tài li u HTML u c bi u di n b ng i t ng node
- DOM trình bày tài li u HTML theo c u trúc hình cây
- Thu c tính nh ngh a các c tính cho node nh nodeName, nodeValue, nodeType, innerText, childNodes, parentNode...

- Phức tạp hơn, thể hiện các thao tác với node như truy cập node, thêm node con cho node và xóa node con
- Có thể truy cập các node bằng phương thức `getElementById(id)`, phương thức `getElementsByTagName(name)` hoặc sử dụng mối quan hệ giữa các node để tìm kiếm
- Có rất nhiều trình duyệt, mỗi trình duyệt lại có cấu trúc DOM và các thuộc tính, phương thức cho mỗi Node khác nhau → Sử dụng cách thức biểu diễn trình duyệt để tìm phương thức nào
- JavaScript cung cấp tính năng style trong mỗi element thay đổi style cho các element

XIN CHÀO!