



THỰC HỌC – THỰC NGHIỆP

LẬP TRÌNH JAVASCRIPT

Dynamic Element Manipulation using
DOM

HỆ THỐNG BÀI CŨ

- Lập trình hướng đối tượng trong Javascript
 - Phương thức lập trình
 - Phương thức lập trình hướng đối tượng: đối tượng, thuộc tính và phương thức
 - Mối liên hệ giữa object và mảng
- Browser Object Model

MỤC TIÊU BÀI HỌC

- DOM cơ bản
 - Giới thiệu về Document Object Model
 - Giới thiệu về HTML DOM
 - Cấu trúc DOM
- Truy xuất các element trong DOM



PHẦN I

DOM CƠ BẢN

Ôn tập HTML – HTML Element

Thẻ mở	Thẻ đóng	Mô tả
<code><p></code>	<code></p></code>	Sử dụng để tạo paragraph
<code><h1></code>	<code></h1></code>	Sử dụng để tạo header; các header nhỏ hơn từ h2 đến h6
<code></code>	<code></code>	Nội dung trong span nằm cùng dòng với nội dung ngoài (Sử dụng cho mục đích layout)
<code><a></code>	<code></code>	Hyperlink – siêu liên kết
<code><button></code>	<code></button></code>	Tạo nút
<code><table></code>	<code></table></code>	Tạo bảng
<code><tr></code>	<code></tr></code>	Tạo dòng trong bảng
<code><td></code>	<code></td></code>	Tạo ô (Cell) trong dòng

Ôn tập HTML – HTML Element

Thẻ mở	Thẻ đóng	Mô tả
		Tạo danh sách không có thứ tự
		Tạo danh sách có thứ tự
		Tạo mục (item) trong danh sách
<div>	</div>	Tạo khu vực (Section) trong trang HTML
<form>	</form>	Tạo form
<input>	</input>	Tạo vùng điền thông tin cho người dùng. Có thể là textbox, checkbox, button, password, numbers, dropdown, radio, ...
<input/>		
 		Ngắt dòng

Ôn tập HTML – HTML Attributes

Tên thuộc tính	Mô tả	Các element có thể sử dụng
id	Id duy nhất cho 1 element	Tất cả các element
name	Tuỳ chỉnh tên cho 1 element	Input, button, form và còn khá nhiều element khác nhưng chưa tìm thấy
class	Metadata thêm vào 1 element	Tất cả các element trong body
value	Đặt giá trị ban đầu của element	Button, input, li và vài element khác nhưng chưa tìm thấy
style	Layout cụ thể cho phần tử	Tất cả các element

Document Object Model

- DOM là một chuẩn được định nghĩa bởi **W3C** (World Wide Web Consortium) để có thể **truy cập** và **thao tác** với các tài liệu như **html** hay **xml** bằng các **ngôn ngữ lập trình** như 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 (Định nghĩa bởi W3C)

DOM

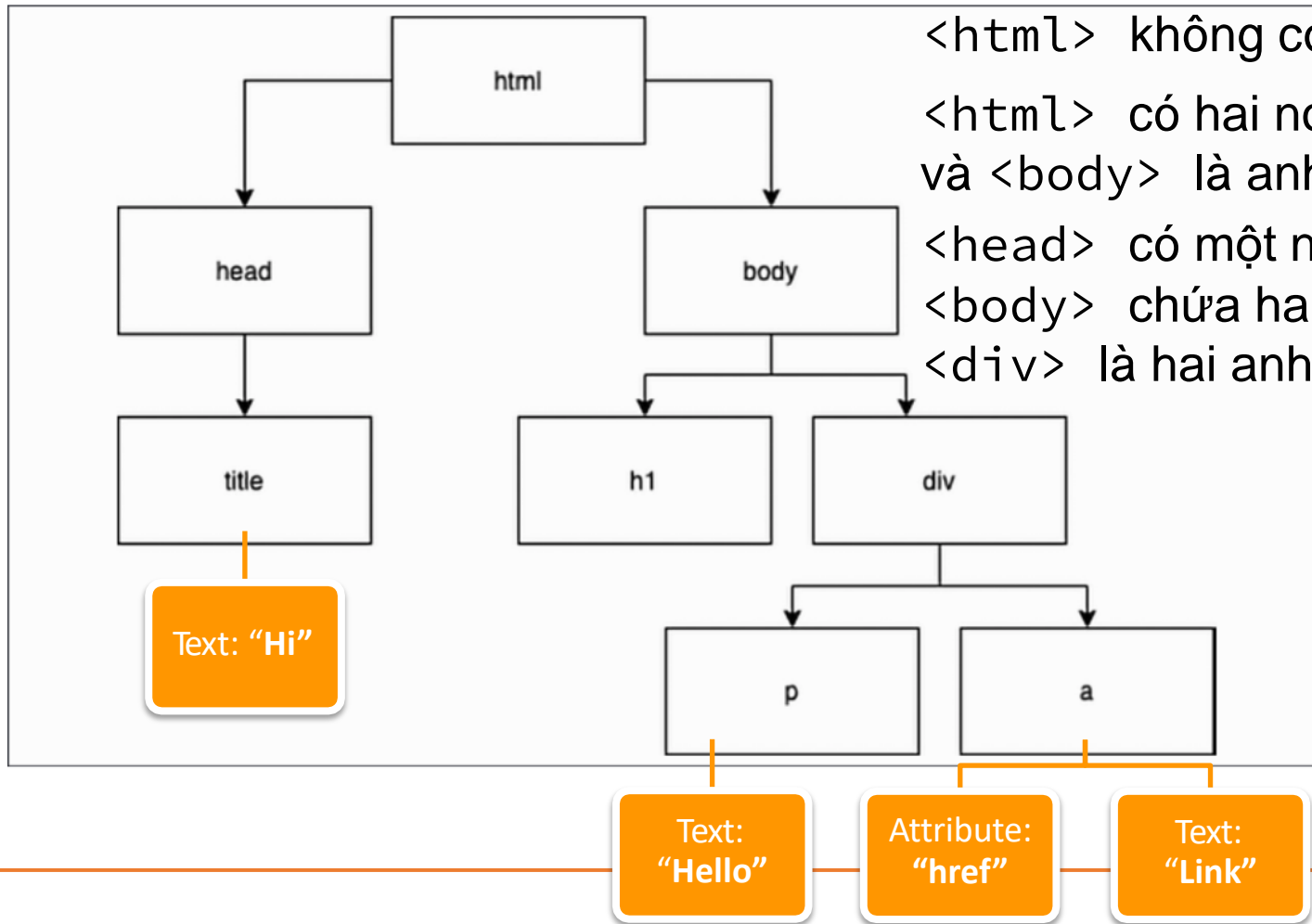
- DOM được chia thành 3 mức
 - Core DOM: Tiêu chuẩn cho bất kỳ tài liệu có cấu trúc nào
 - XML DOM: Tiêu chuẩn cho tài liệu XML
 - HTML DOM: Tiêu chuẩn cho tài liệu HTML
- HTML DOM định nghĩa các **đối tượng** và **thuộc tính** của tất cả các thành phần HTML và **phương thức** để truy cập đến chúng.
- Hay **HTML DOM là chuẩn để lấy, thay đổi, thêm, xoá các thành phần HTML**

Node trong DOM

- Tất cả các thành phần trong tài liệu HTML đều được biểu diễn bằng đối tượng 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**

Cấu trúc hình cây DOM

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



`<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>`

`<body>` chứa hai node con là `<h1>` và `<div>`, `<h1>` và `<div>` là hai anh em

`<title>` có một node con là text node “Hi”

Node cha của node `<head>` và `<body>` là node `<html>`

Node cha của text node “Hello” là node `<p>`

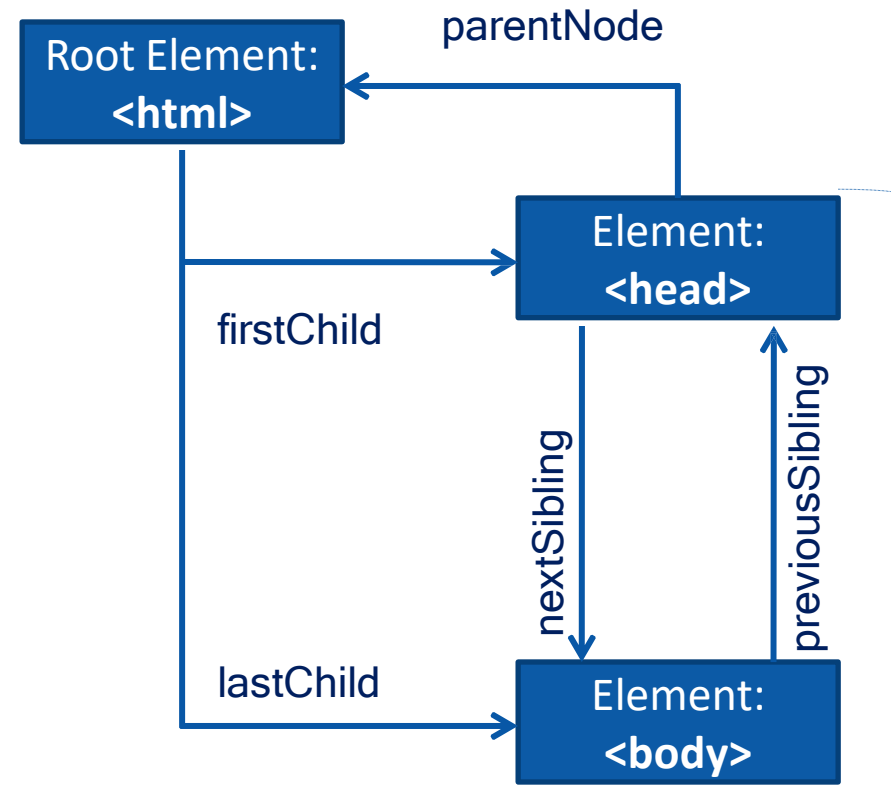
Text node “Hi”, “Hello”, “Link” và attribute node “href” đều là node lá. 11

QUAN HỆ GIỮA CÁC NODE

- 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 đầu tiên được 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

CON CẢ, CON ÚT

- `<head>` và `<body>` là anh em, trong đó `<head>` là con cả của `<html>` còn `<body>` là con út của `<html>`
- `<h1>` và `<div>` là con cả của `<body>`, trong đó `<h1>` là con cả của `<body>`, còn `<div>` là con út của `<body>`



Là các con của `<html>` và anh em với nhau



PHẦN II

TRUY XUẤT CÁC ELEMENT TRONG DOM

Thuộc tính và phương thức của node

- DOM định nghĩa các thuộc tính và các phương thức cho các node để hỗ trợ cho việc lập trình
- Thuộc tính định nghĩa các đặc tính cho node
- Phương thức để thực hiện các thao tác với node
 - Truy cập đến node
 - Thêm node con cho node
 - Xóa node con

Nhắc lại về CSS

- CSS định nghĩa màu sắc, font, layout...cho trang web
- CSS bao gồm một tập các thuộc tính, mỗi thuộc tính có một tập 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;}`

Các thuộc tính của node

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

Truy xuất các element trong DOM

- Duyệt DOM bằng cách sử dụng document (document chứa tất cả các HTML và là đại diện cho trang web)
- Có thể truy cập đến các element (node) bằng cách sau
 - Truy xuất các element (node) **bằng ID: `getElementById()`**
 - Truy xuất các element (node) **bằng tên thẻ (Tag name): `getElementsByTagName()`**
 - Truy xuất các element (node) **bằng tên class (Class name): `getElementsByClassName()`**
 - Truy xuất các element (node) với **CSS selector: `querySelector()`, `querySelectorAll()`**

Truy xuất các element trong DOM

- Truy xuất các element (node) **bằng ID**: `getElementById()`

```
<html>
  <body>
    <h1 style="color: pink;">Just an example</h1>
    <div id="one" class="example">Hi!</div>
    <div id="two" class="example">Hi!</div>
    <div id="three" class="something">Hi!</div>
  </body>
  <script>
    console.log(document.getElementById("two"));
  </script>
</html>
```

Truy xuất các element trong DOM

- Truy xuất các element (node) với **Tag name**: `getElementByTagName()`

```
<html>
  <body>
    <h1 style="color: pink;">Just an example</h1>
    <div id="one" class="example">Hi!</div>
    <div id="two" class="example">Hi!</div>
    <div id="three" class="something">Hi!</div>
  </body>
  <script>
    console.log(document.getElementsByTagName("div").item(1));
    console.log(document.getElementsByTagName("div").namedItem("one"));
  </script>
</html>
```

Truy xuất các element trong DOM

- Truy xuất các element (node) với **class name**: `getElementsByClassName()`

```
<html>
  <body>
    <h1 style="color: pink;">Just an example</h1>
    <div id="one" class="example">Hi!</div>
    <div id="two" class="example">Hi!</div>
    <div id="three" class="something">Hi!</div>
  </body>
  <script>
    console.log(document.getElementsByClassName("example"));
  </script>
</html>
```

Truy xuất các element trong DOM

- Truy xuất các element (node) với **CSS selector**

```
<html>
  <head>
    <title>Hi</title>
  </head>
  <body>
    <div id="hoa">
      <p>Hoa Hong</p>
      <p>Hoa Lan</p>
      <p>Hoa Dao</p>

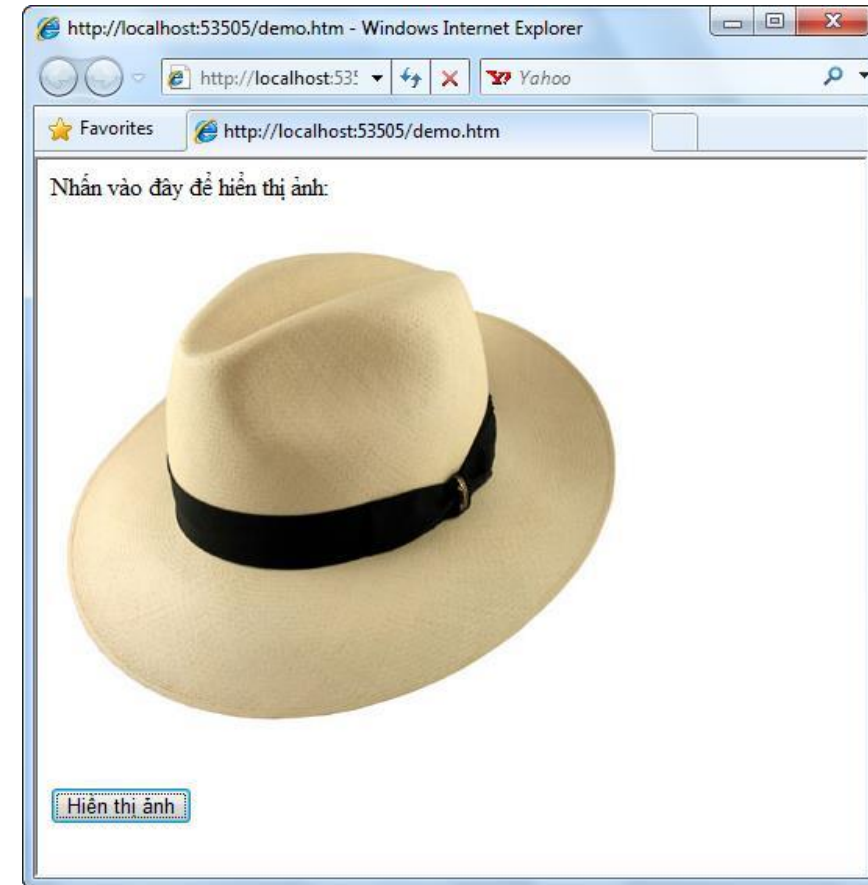
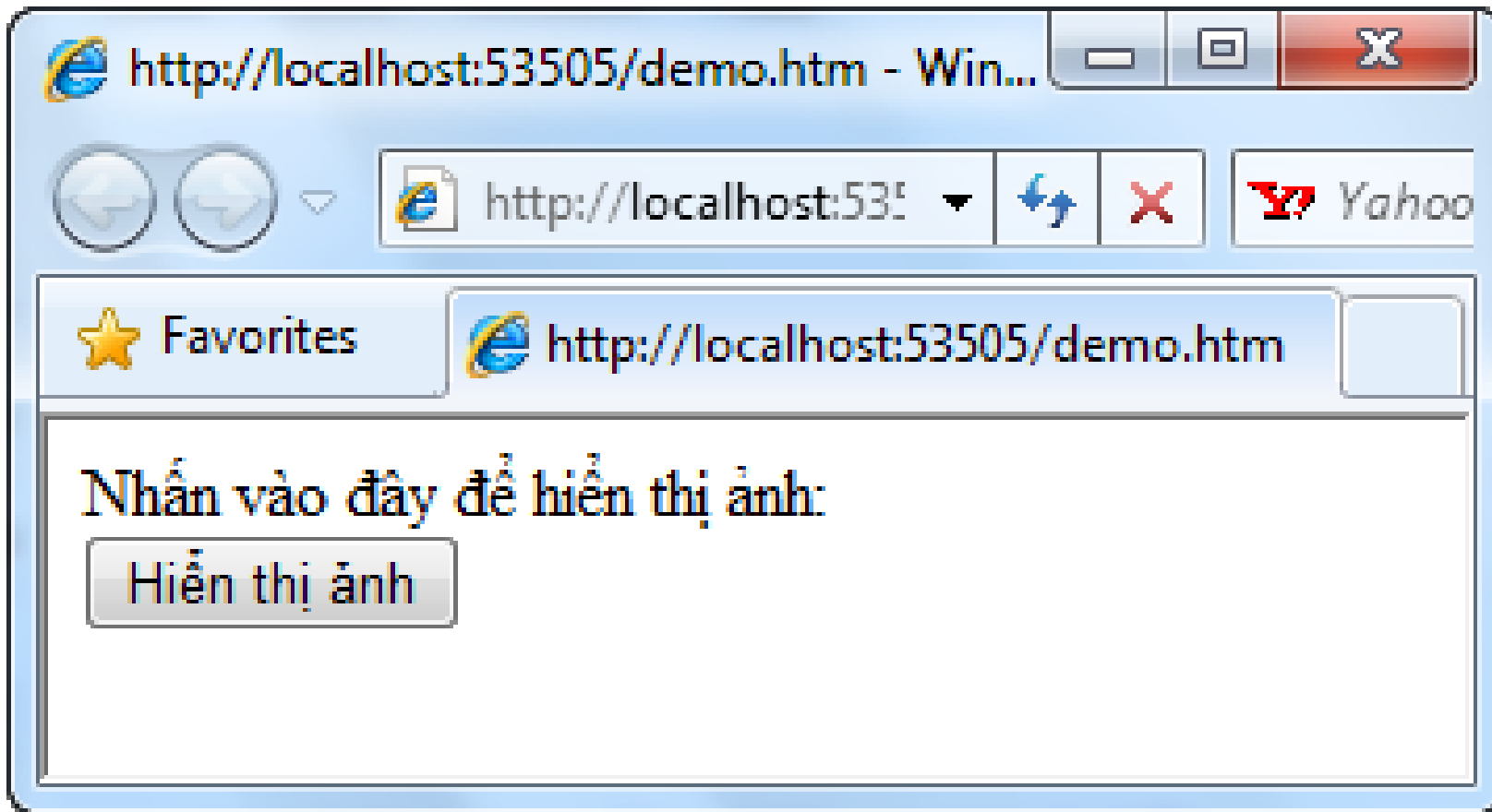
    <div class="example">Hi</div>
    <div class="example">Hello</div>
    </div>
    <div id="three" class="something">Hi!</div>
    <script type="text/javascript">
      console.log(document.querySelector("div"));
      console.log(document.querySelector(".something"));
      console.log(document.querySelectorAll("div.example"));
    </script>
  </body>
</html>
```

Demo thuộc tính innerHTML

- Demo thuộc tính innerHTML

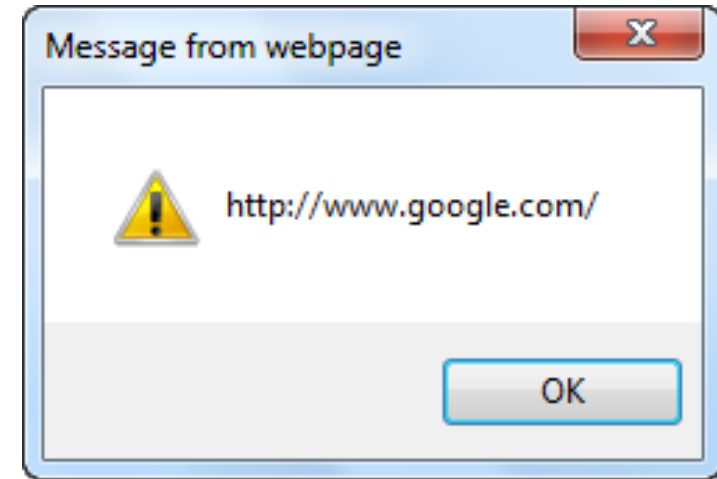
```
<html>
  <head>
    <script type="text/javascript">
      function hienThi() {
        var node = document.getElementById("anhDiv");
        node.innerHTML = "<img src='mu.jpg'>";
      }
    </script>
  </head>
  <body>
    Nhấn vào đây để hiển thị ảnh:
    <div id="anhDiv"></div>
    <input type="button" value="Hiển thị ảnh" onclick="hienThi()"/>
  </body>
</html>
```

Demo thuộc tính innerHTML



Demo lấy attribute của node (element)

```
<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>
```



Demo lấy attribute của node (element)

- Giả sử đoạn mã được viết lại 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 hộp thoại???

Demo lấy attribute của node (element)

- **Trả lời:** Tại vì browser làm việc theo cơ chế thông dịch. Tức là dịch từng dòng một, khi đến lệnh Javascript `document.getElementById("link")` thì chưa có id nào tên là "Link" nên không có node nào trả về cho biến linkNode

Demo điều hướng qua các node

```
<html>
  <head>
    <title>Hi</title>
  </head>
  <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>
```



Thêm node vào element

- Sử dụng phương thức **createElement** và **appendChild** để thêm node element vào tài liệu

```
<html>
  <head>
    <title>Hi</title>
  </head>
  <body>
    <div id="hoa">
      <p>Hoa Hong</p>
      <p>Hoa Lan</p>
      <p>Hoa Dao</p>
    </div>
    <script type="text/javascript">
      var newElement = document.createElement("p");
      document.body.appendChild(newElement);
      var text = document.createTextNode("Hello World");
      newElement.appendChild(text );
    </script>
  </body>
</html>
```

Xoá node element

- Sử dụng phương thức **removeChild(nodeId)** của node để xoá các node element của node

```
<html>
  <head>
    <title>Hi</title>
  </head>
  <body>
    <p id="pHello">Hello</p>
    <p id="pHi">Hi</p>

    <script type="text/javascript">
      document.body.removeChild(pHi);
      //var pHi = document.getElementById("pHi");
      //document.body.removeChild(pHi);
    </script>
  </body>
</html>
```

Element click handler (Xử lý sự kiện)

- HTML elements có thể làm điều gì đó khi bạn click chuột

- Ví dụ

```
<!DOCTYPE html>
<html>
  <body>
    <div id="one" onclick="alert('Ouch! Stop it!')">Don't click here!</div>
  </body>
</html>
```

- Hoặc

```
<!DOCTYPE html>
<html>
  <body>
    <script>
      function stop(){
        alert("Ouch! Stop it!");
      }
    </script>
    <div id="one" onclick="stop()">Don't click here!</div>
  </body>
</html>
```

Element click handler (Xử lý sự kiện)

- HTML elements có thể làm điều gì đó khi bạn click chuột
- Sử dụng DOM

```
<!DOCTYPE html>
<html>
  <body>
    <div id="one">Don't click here!</div>
  </body>
  <script type="text/javascript">
    function e_click() {
      alert("Auch! Stop!");
    }
    document.getElementById("one").onclick = eclick();
  </script>
</html>
```


This và DOM

- This keyword có ý nghĩa tương đối và phụ thuộc vào ngữ cảnh sử dụng nó
- Trong DOM, this đề cập đến DOM element mà nó thuộc về

```
<!DOCTYPE html>
<html>
  <body>
    <script>
      function reveal(el){
        console.log(el);
      }
    </script>
    <button onclick="reveal(this)">Click here!</button>
  </body>
</html>
```

Lấy thông tin browser

- Vấn đề nảy sinh:
 - Có rất nhiều trình duyệt
 - Mỗi trình duyệt lại có nhiều phiên bản
 - Mỗi trình duyệt lại hỗ trợ ở các mức khác nhau
- ⇒ Để kiểm tra tất cả các trình duyệt, phiên bản của trình duyệt là điều không thể
- ⇒ Thuộc tính userAgent ([navigator.userAgent](#)) cũng có thể trả về thông tin sai
- Sử dụng cách thử để biết trình duyệt hỗ trợ phương thức gì

```
if (typeof document.body.firstChild != "undefined") {  
    alert("Browser ho tro phuong thuc firstElementChild");  
} else {  
    alert("Browser khong ho tro phuong thuc firstElementChild");  
}
```

Đối mặt với các browser cũ

- Tạo mã Javascript chạy tốt trên tất cả các version của tất cả các trình duyệt là điều không thể
- Thiết lập một giới hạn hợp lý các trình duyệt và version để hỗ trợ
- Giới hạn càng hẹp thì càng ít khách hàng truy cập được vào website
- Đối với browser không hỗ trợ Javascript hoặc bị disable Javascript
 - Dùng thẻ `<noscript></noscript>`
 - Dùng thẻ `<!--> <-->`

TỔNG KẾT BÀI HỌC

- DOM là một chuẩn được định nghĩa bởi W3C (World Wide Web Consortium) để có thể truy cập và thao tác với các tài liệu như html hay xml bằng các ngôn ngữ lập trình như Javascript, VB...
- DOM được chia làm 3 mức CoreDOM, HTML DOM và XML DOM
- HTML DOM định nghĩa các đối tượng và thuộc tính của tất cả các thành phần HTML và phương thức để truy cập đến chúng.
- Tất cả các thành phần trong tài liệu HTML đều được biểu diễn bằng đối tượng node
- DOM trình bày tài liệu HTML theo cấu trúc hình cây
- Thuộc tính định nghĩa các đặc tính node như nodeName, nodeValue, innerText, childNodes, parentNode...

TỔNG KẾT BÀI HỌC

- Phương thức để thực hiện các thao tác với node như truy cập đến node, thêm node con và xoá node con.
- Có thể truy cập đến node với Id, tag name, class name, css selector
- Có rất nhiều trình duyệt, mỗi trình duyệt lại định nghĩa cấu trúc DOM và các thuộc tính, phương thức cho mỗi Node khác nhau => Sử dụng cách thử để biết trình duyệt hỗ trợ phương thức nào.

*Thank
you!*