PHẦN II. JAVASCRIPT

Giảng Viên: Ths. Phạm Đào Minh Vũ

Email: vupdm@itc.edu.vn

JAVASCRIPT CƠ BẢN

MÔ HÌNH DOM

PHẦN 1. JAVASCRIPT CƠ BẢN

- Giới thiệu Javascript
- Cú pháp
- Event

Giới thiệu Javascript

- ✓ Javascript là 1 ngôn ngữ hướng đối tượng dùng để xử lý các thành phần HTML trong 1 trang web
- ✓ Javascript chạy trên phía client (trên trình duyệt – IE,FF,Opera,Chrome ...)
- ✓ Javascript được tạo ra năm 1995 bởi Brendan Eich của Netscape (Mozilla hiện tại) dưới tên Mocha, sau đó đổi thành Livescript rồi Javascript.

Đặc điểm javascript

- ✓ Là ngôn ngữ hướng đối tượng. Ta có thể tạo, sử dụng các đối tượng
- ✓ Javascript sử dụng trình biên dịch được tích hợp với trình duyệt, được hỗ trợ bởi hầu hết các trình duyệt kể cả trên mobile
- ✓ Cú pháp Javascript tương đồng với C/C++ và Java nên dễ viết, dễ học

Cách viết Javascript

- Có 2 cách viết Javascript :
  - Cách 1 : Nhúng đoạn javascript vào trong file HTML
  - Cách 2 : Viết javascript thành 1 file riêng có đuôi .js và liên kết với file HTML

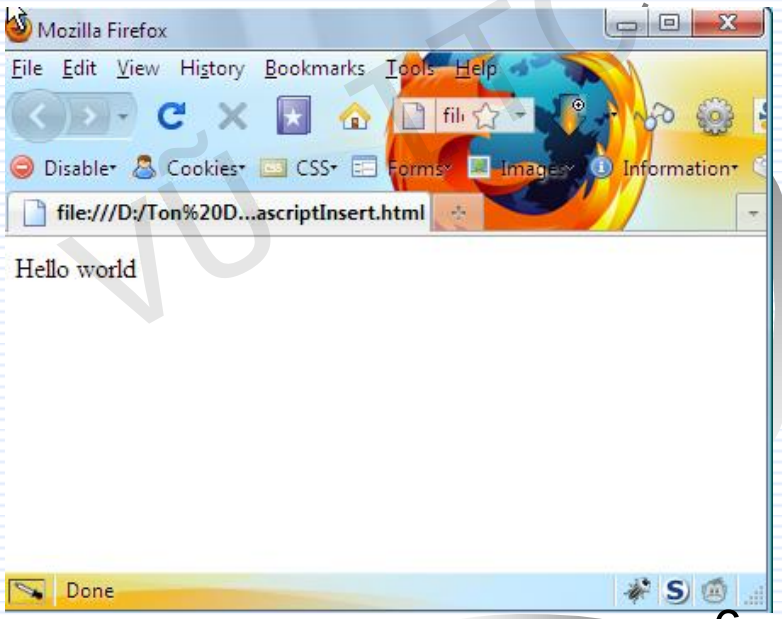
Nhúng javascript vào HTML

- Ta sử dụng thẻ <script> có như sau để chèn đoạn mã Javascript vào bất kỳ nơi nào trong file HTML (<head> và <body>) :

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

- Ví dụ :

```
1  <html>
2  <head>
3  <script lanuage="javascript">
4      document.write("Hello world");
5  </script>
6  </head>
7  <body>
8  </body>
9  </html>
```



Tạo file .js

- ✓ Ta sử dụng thẻ <script> với thuộc tính src để liên kết 1 file javascript vào HTML :

```
<script language="javascript"
  src="myscript.js">
</script>
```

```
1  <html>
2  <head>
3  <script language="javascript" src="myscript.js">
4  </script>
5  </head>
6  <body>
7  </body>
8  </html>
```

ImageSprite.html | ImageOpactity.html | LinkJsFile.html | myscript.js

```
1  document.write("Hello world");
```

Nội dung

- Giới thiệu Javascript
- Cú pháp
- Hàm

Cú pháp Javascript

- ✓ Lệnh đơn : mỗi lệnh đơn kết thúc bằng ;
- ✓ Khối lệnh : được bao bằng { }
- ✓ Chú thích : // và /* .. */
- ✓ Cấu trúc điều khiển :
  - Rẽ nhánh : if, else, switch
  - Lặp : for, while, do... while, for ... in

Biến trong javascript

- ✓ Javascript không cần khai báo biến vẫn có thể sử dụng được
- ✓ Tên biến phân biệt hoa thường, phải bắt đầu bằng kí tự hoặc gạch dưới (_)
- ✓ Biến nếu được khai báo thì không cần khai báo kiểu :
  - **var a;**
  - **a = 10;**
- ✓ Một biến có thể chứa bất kỳ giá trị nào (nguyên, thực , chuỗi ...)

Tầm vực của biến

- Tầm vực là tầm ảnh hưởng của biến :
  - Biến toàn cục : được khai báo ngoài các hàm.Biến có tác dụng từ vị trí khai báo cho đến cuối chương trình
  - Biến cục bộ : được khai báo trong hàm. Biến chỉ có tác dụng trong hàm được khai báo.
- Nếu trong hàm, biến cục bộ trùng tên với biến toàn cục thì biến cục bộ sẽ được sử dụng

Kiểu dữ liệu

- ✓ Biến trong javascript không cần khai báo kiểu dữ liệu
- ✓ Khai báo biến kiểu số :
  - `a = 1.4; b = 2`
- ✓ Khai báo biến kiểu chuỗi :
  - `str = 'Chuỗi dùng dấu nháy đơn' ; str2 = "Dấu nháy kép"`
- ✓ Khai báo biến boolean
  - `var dung = true,sai = fase;`
- ✓ Khai báo biến null
  - `obj = null`

Phép toán

Tóan Tử	Chức Năng	Ví dụ	Kết quả
+	cộng	x=2; x+2	4
-	Trừ	x=2; 5-x	3
*	Nhân	x=4; x*5	20
/	Chia	5/2	2.5
%	Chia lấy phần dư	5%2	1
++	Tăng 1	x=5; x++	6
--	Giảm 1	x=5; x--	x=4

Phép gán

Tóan Tử	Ví dụ	Tương đương
=	x = y	x= y
+=	x += y	x = x+y
-=	x -= y	x = x-y
*=	x *= y	x = x*y
/=	x /= y	x= x/y
%=	x%=y	x = x%y

Phép so sánh

Tóan Tử	Chức Năng	Ví dụ
= =	bằng	5==8 returns false
!=	Không bằng	5!=8 returns true
>	lớn hơn	5>8 returns false
<	nhỏ hơn	5<8 returns true
>=	lớn hợn hoặc bằng	5>=8 returns false
<=	nhỏ hơn hoặc bằng	5<=8 returns true

Phép toán logic

Tóan Tử	Chức Năng	Ví dụ
&&	Và	x =6; y =3 ; (x < 10 && y > 1) returns true
||	hoặc	x = 6 ; y =3 (x==5 || y==5) returns false
!	not	x=6; y =3; !(x==y) returns true

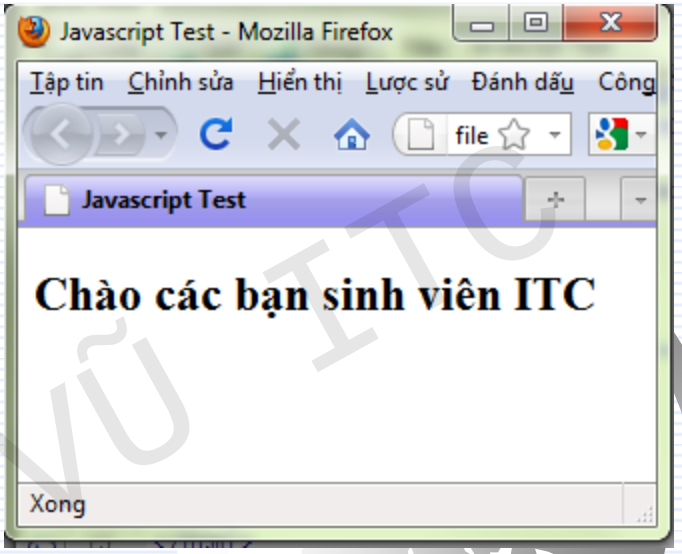
Phép toán +

- ✓ Phép + trên 1 chuỗi sẽ cho ra chuỗi.
- ✓ Ví dụ : s = "Chào các bạn" + "sinh viên ITC"

java1.html*

Code | Split | Design | Title: Javascript Test

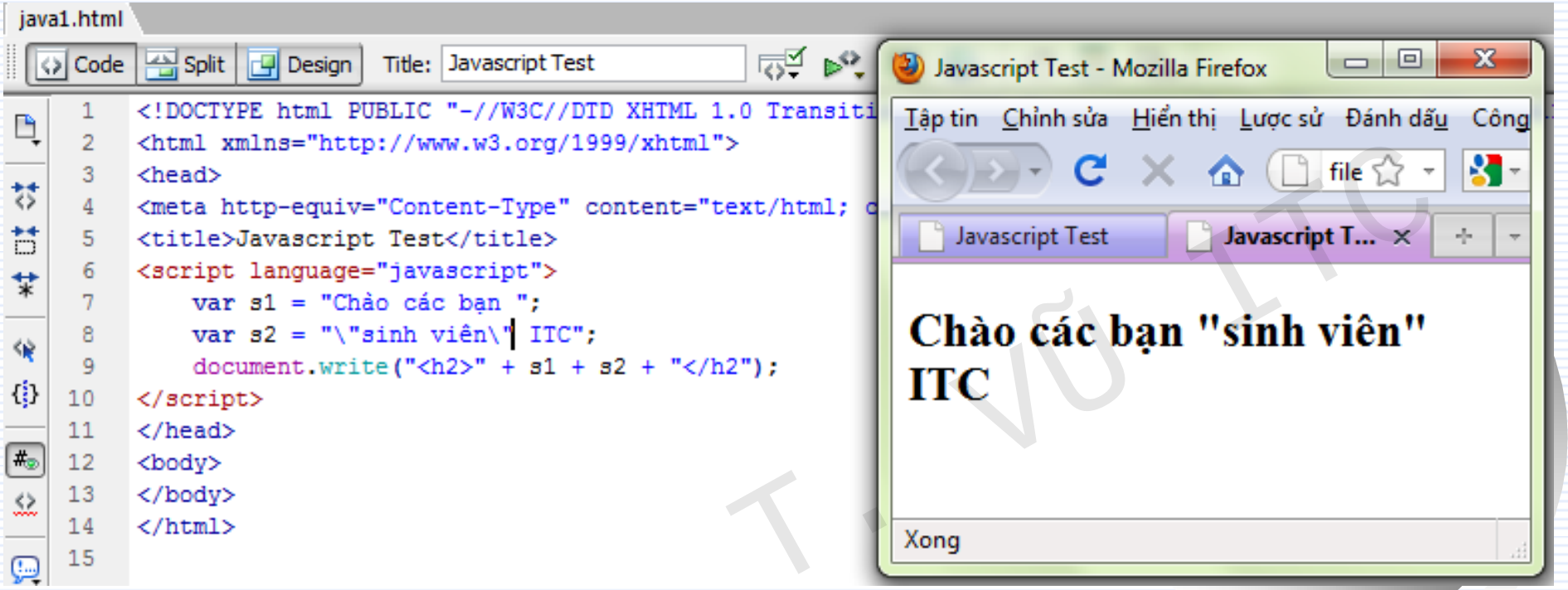
```
1  <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transi
2  <html xmlns="http://www.w3.org/1999/xhtml">
3  <head>
4  <meta http-equiv="Content-Type" content="text/html;
5  <title>Javascript Test</title>
6  <script language="javascript">
7      var s1 = "Chào các bạn ";
8      var s2 = "sinh viên ITC";
9      document.write("<h2>" + s1 + s2 + "</h2");
10 </script>
11 </head>
12 <body>
13 </body>
14 </html>
```



Kí tự đặc biệt

- Các kí tự đặc biệt muốn xuất hiện trong chuỗi phải escape :
  - \n : new line
  - \t : tab
  - \b : BackSpace
  - \& : dấu &
  - \”: dấu “

Kí tự đặc biệt



Cấu trúc rẽ nhánh

- if , else :

```
if (n % 2 == 0)
  document.write("Chẵn");
else
  document.write("Lẻ");
```

- switch :

```
switch (n)
{
  case 2 : document.write("Thứ 2"); break;
  case 3 : document.write("Thứ 3"); break;
}
```

Cấu trúc lặp

- for

```
var i=0;
for (i=0;i<=5;i++)
{
    document.write("The number is " + i);
    document.write("<br />");
}
```

- while

```
var i=0;
while (i<=5)
{
  document.write("The number is " + i);
  document.write("<br />");
  i++;
 }
```

Cấu trúc lặp

- for ... in

```
var x;
var mycars = new Array();
mycars[0] = "Saab";
mycars[1] = "Volvo";
mycars[2] = "BMW";

for (x in mycars)
{
  document.write(mycars[x] + "<br />");
}
```

Hàm

- Cú pháp :

```
function functionName(var1,var2,...,varX)
{
    //some code
}
```

- Hàm không được thực thi khi trang web được load
- Hàm chỉ thực thi khi được gọi (ví dụ trong sự kiện hoặc trực tiếp)
- Hàm có thể đặt ở trong <head> hoặc <body> nhưng nên đặt trong <head>. Nó sẽ luôn được nạp trước khi gọi

Ví dụ hàm

```
<html>
<head>
<script type="text/javascript">
function displaymessage()
{
alert("Hello World!");
}
</script>
</head>

<body>
<form>
<input type="button" value="Click me!" onclick="displaymessage()" />
</form>
</body>
</html>
```

Hàm trả về giá trị

- Từ khóa return dùng để trả về giá trị cho hàm
- Cú pháp :

```
function functionName(var1,var2,...,
                                varX)
{
    //some code …
    return value;
}
```

Ví dụ hàm trả về giá trị

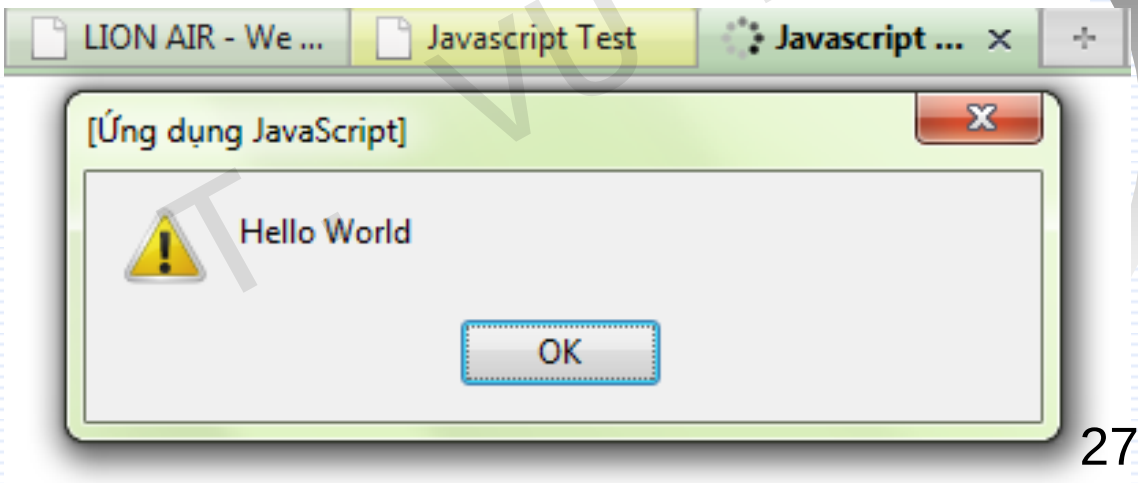
```
<html>
<head>
<script type="text/javascript">
function product(a,b)
{
return a*b;
}
</script>
</head>

<body>
<script type="text/javascript">
document.write(product(4,3));
</script>

</body>
</html>
```

Các hàm thông dụng – hàm alert()

- ✓ alert ("Nội dung thông báo") : hiển thị hộp thoại thông báo có 1 nút OK.

MACHome.txt | Bai tap FUNCTION.txt | AddString.html | SpecialChar.html | FunctionExample.html | FunctionExampleReturnValue.html | Alert.html

```
1 <html><head><title>Function</title></head>
2 <body>
3 <script>
4 alert("Hello World")
5 </script>
6 </body></html>
```



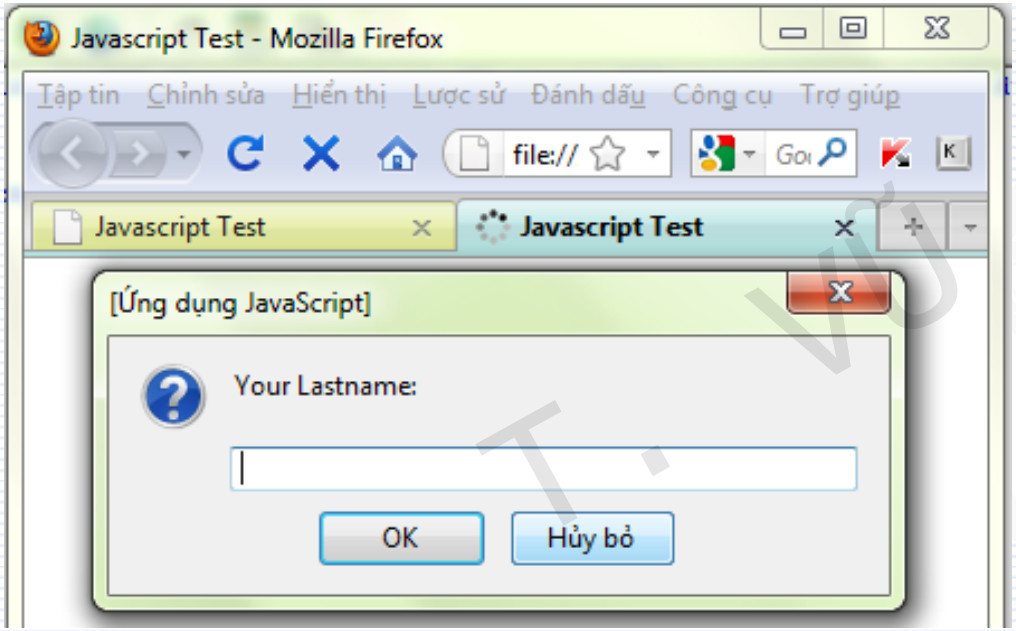
Hàm prompt

- **prompt** (“Thông báo”, Giá_trị_mặc_định) : hiển thị hộp thoại với câu thông báo và 2 nút OK, Cancel và một textfield cho phép người dùng nhập vào 1 giá trị.
- Nếu người dùng nhấn OK, hàm prompt() sẽ trả về chuỗi được nhập, ngược lại giá_trị_mặc_định sẽ trả về

Hàm

MACHome.txt | Bai tap FUNCTION.txt | AddString.html | SpecialChar.html | FunctionExample.html | FunctionExampleReturnValue.html | Alert.html | Prompt.html

```
1  <script>
2  a=prompt("Your Lastname:");
3  b=prompt("Your FirstName:");
4  document.write("Your FullName is :"+ a + ' ' + b)
5  </script>
```



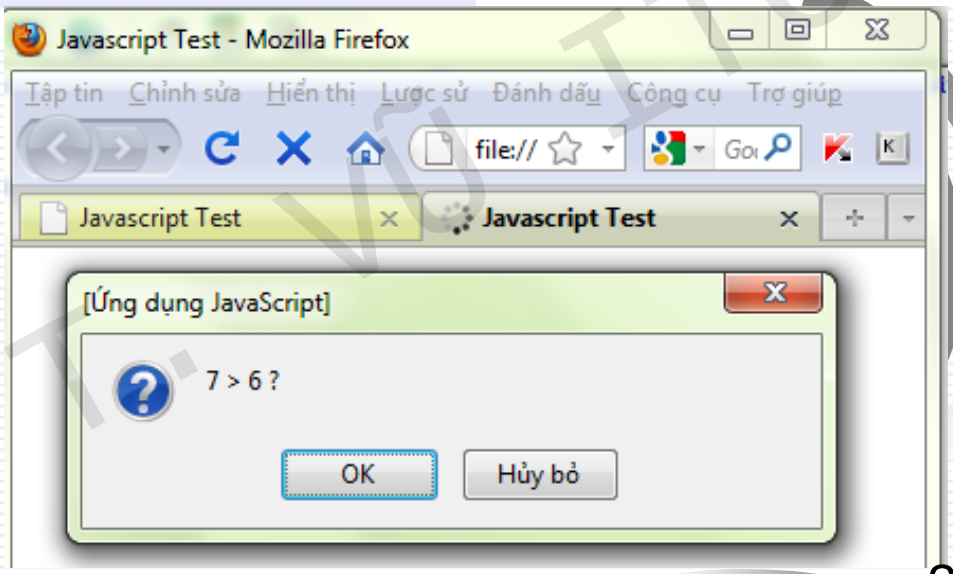
Hàm confirm()

- **confirm**("Thông báo") : là hàm dùng để xác nhận lại thông tin, hiển thị câu thông báo với 2 nút OK, Cancel.
- **confirm**() trả về giá trị **true** nếu OK được nhấn và **false** nếu Cancel được nhấn

Hàm confirm() – ví dụ

Confirm.html

```
1  <script>
2  a=prompt("nhap so a :");
3  b=prompt("nhap so b");
4  c=confirm( a +' lon hon '+ b+'?')
5  if(c == true)
6  document.write( a +" > "+b  )
7  else
8  document.write( a +" < "+b  )
9  </script>
```



Hàm document.write()

- Hàm document.write("Chuỗi") : dùng để ghi 1 chuỗi ra trang HTML
- document.writeln("Chuỗi") : ghi chuỗi ra và kết thúc bằng ký tự xuống dòng. Hàm này nên đi kèm với thẻ <pre> để giữ lại kí tự xuống dòng
- Ví dụ :
  - document.write("Hello world");
  - document.writeln("Hello world");

Hàm eval()

- ✓ Hàm eval("Chuỗi") : Chuyển đổi giá trị chuỗi thành giá trị số
- ✓ Ví dụ

Confirm.html | Eval.html | EvalExample.html

```
1  <script>
2  a = eval(prompt("Nhap so  a:"));
3  b = eval(prompt("Nhap so  b:"));
4  c = a+b ;
5  document.write(c)
6  </script>
```

Hàm parseInt()

- ✓ parseInt("Chuỗi",[radix]) : hàm đổi chuỗi ra số nguyên với cơ số là tham số radix.
- ✓ Nếu Chuỗi gồm các kí số rồi đến kí tự (123abc) thì các kí tự sẽ bị bỏ qua (trả về 123)
- ✓ Nếu Chuỗi bắt đầu không phải kí số thì parseInt sẽ trả về NaN (Not a Number)

Confirm.html | Eval.html | EvalExample.html | ParseInt.js

```
1  var  strNum="123.8" , kq;
2  kq=parseInt(strNum) //kq=123
3  strNum="a123"
4  kq=parseInt(strNum) //kq=NaN
5  strNum="123.8abc"
6  kq=parseInt(strNum) //kq=123
```

Hàm parseFloat()

- ✓ parseFloat ("Chuỗi") : hàm đổi chuỗi ra số thực
- ✓ Nếu Chuỗi gồm các kí số rồi đến kí tự (123abc) thì các kí tự sẽ bị bỏ qua (trả về 123)
- ✓ Nếu Chuỗi bắt đầu không phải kí số thì parseFloat sẽ trả về NaN (Not a Number)

Confirm.html | Eval.html | EvalExample.html | ParseInt.js | ParseFloat.js

```
1  var  strNum="123.8" , kq;
2  kq=parseFloat(strNum) //kq=123.8
3  strNum="a123.8"
4  kq=parseFloat(strNum) //kq=NaN
5  strNum="123.8abc"
6  kq=parseFloat(strNum) //kq=123.8
```

Hàm isNaN()

- isNaN("Chuỗi") : hàm kiểm tra xem 1 chuỗi có phải là số không ? Nếu là số trả về false, ngược lại true

Confirm.html | Eval.html | EvalExample.html | ParseInt.js | ParseFloat.js | isNaN.js

```
1  var str="123abc", kq;
2  kq=isNaN(str); //kq=true
3  str="123.8"
4  kq=isNaN(str); //kq=false
```

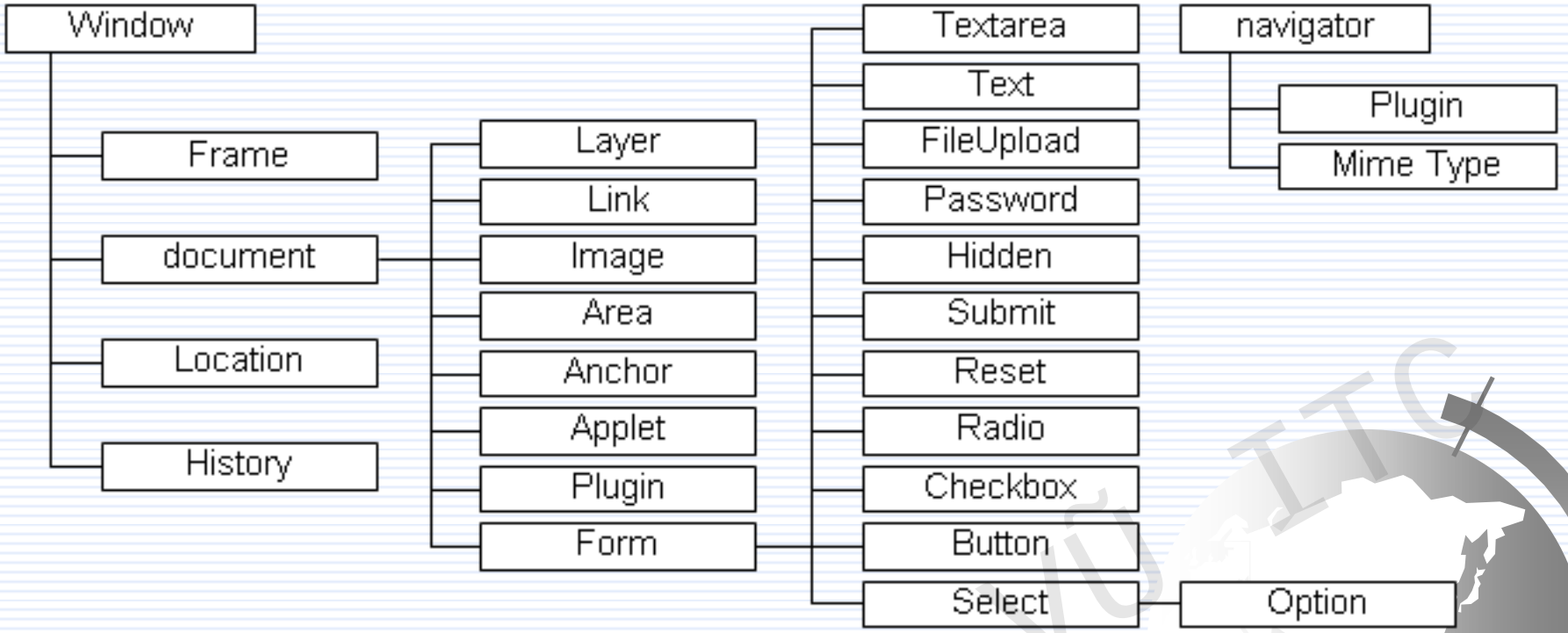
Nội dung

- Giới thiệu Javascript
- Cú pháp
- Event

Event

- ✓ Sử dụng javascript ta có thể tạo các trang web động
- ✓ Event là các sự kiện phát sinh khi người dùng tương tác với các element trong trang web
- ✓ Mỗi element có 1 số các sự kiện riêng có thể kích hoạt javascript
- ✓ Một số các ví dụ :
  - 1 cú click chuột
  - 1 trang web hay 1 hình ảnh đang được nạp
  - Di chuyển chuột lên trên 1 element
  - Chọn 1 ô text field trong HTML
  - Gởi dữ liệu trong form HTML
  - Nhấn phím

Event – Cấu trúc phân cấp



Ví dụ event

- Một số sự kiện tiêu biểu

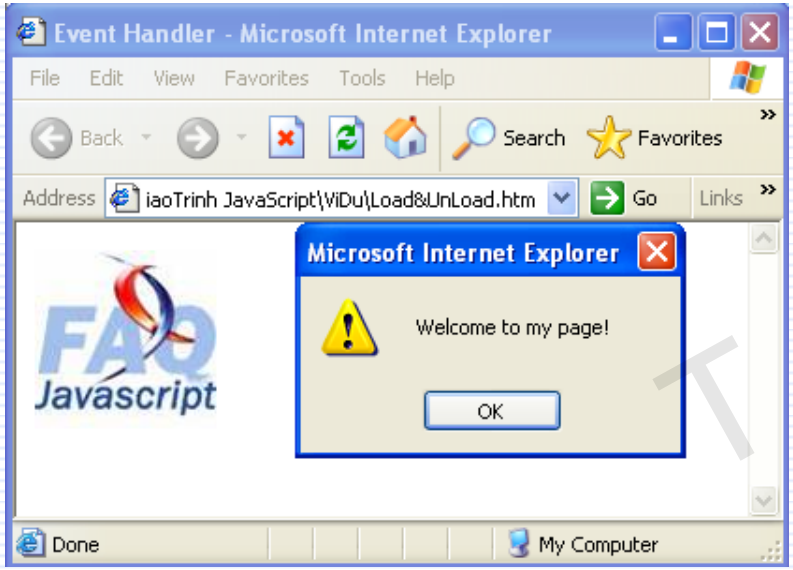
onBlur	Xảy ra khi control mất focus
onClick	Xảy ra khi người dùng kích vào các thành phần hay liên kết của form.
onChange	Xảy ra khi giá trị của thành phần được chọn thay đổi
onFocus	Xảy ra khi thành phần của form được focus.
onLoad	Xảy ra trang Web được tải.
onMouseOver	Xảy ra khi di chuyển chuột lên trên control
onSelect	Xảy ra khi người sử dụng lựa chọn một trường nhập dữ liệu trên form.
onSubmit	Xảy ra khi người dùng chuyển dữ liệu về server (bấm vào nút submit).
onUnLoad	Xảy ra khi người dùng đóng trang

Các sự kiện của một vài đối tượng

Đối tượng	Chương trình xử lý sự kiện có sẵn
Selection list	onBlur, onChange, onFocus
Text	onBlur, onChange, onFocus, onSelect
Textarea	onBlur, onChange, onFocus, onSelect
Button	onClick
Checkbox	onClick
Radio button	onClick
Hypertext link	onClick, onMouseOver, onMouseOut
Clickable Imagemap area	onMouseOver, onMouseOut
Reset button	onClick
Submit button	onClick
Document	onLoad, onUnload, onError
Window	onLoad, onUnload, onBlur, onFocus
Framesets	onBlur, onFocus
Form	onSubmit, onReset
Image	onLoad, onError, onAbort

Ví dụ event

```
<HTML>
<HEAD> <TITLE>Event Handler</TITLE>
</HEAD>
<BODY onLoad="alert('Welcome to my page!');"
onUnload="alert('Goodbye! ');">
<IMG src="logo.gif">
</BODY>
</HTML>
```

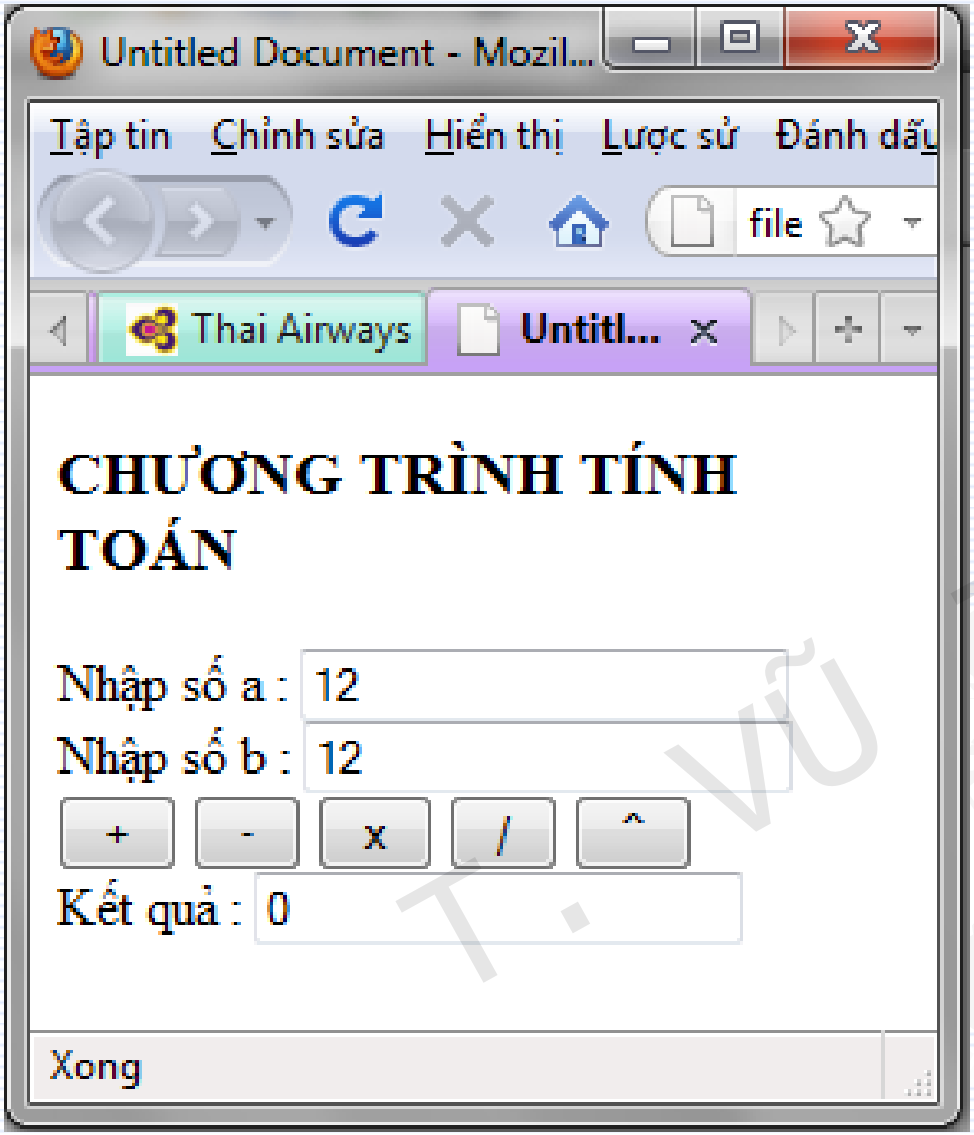


Ví dụ event

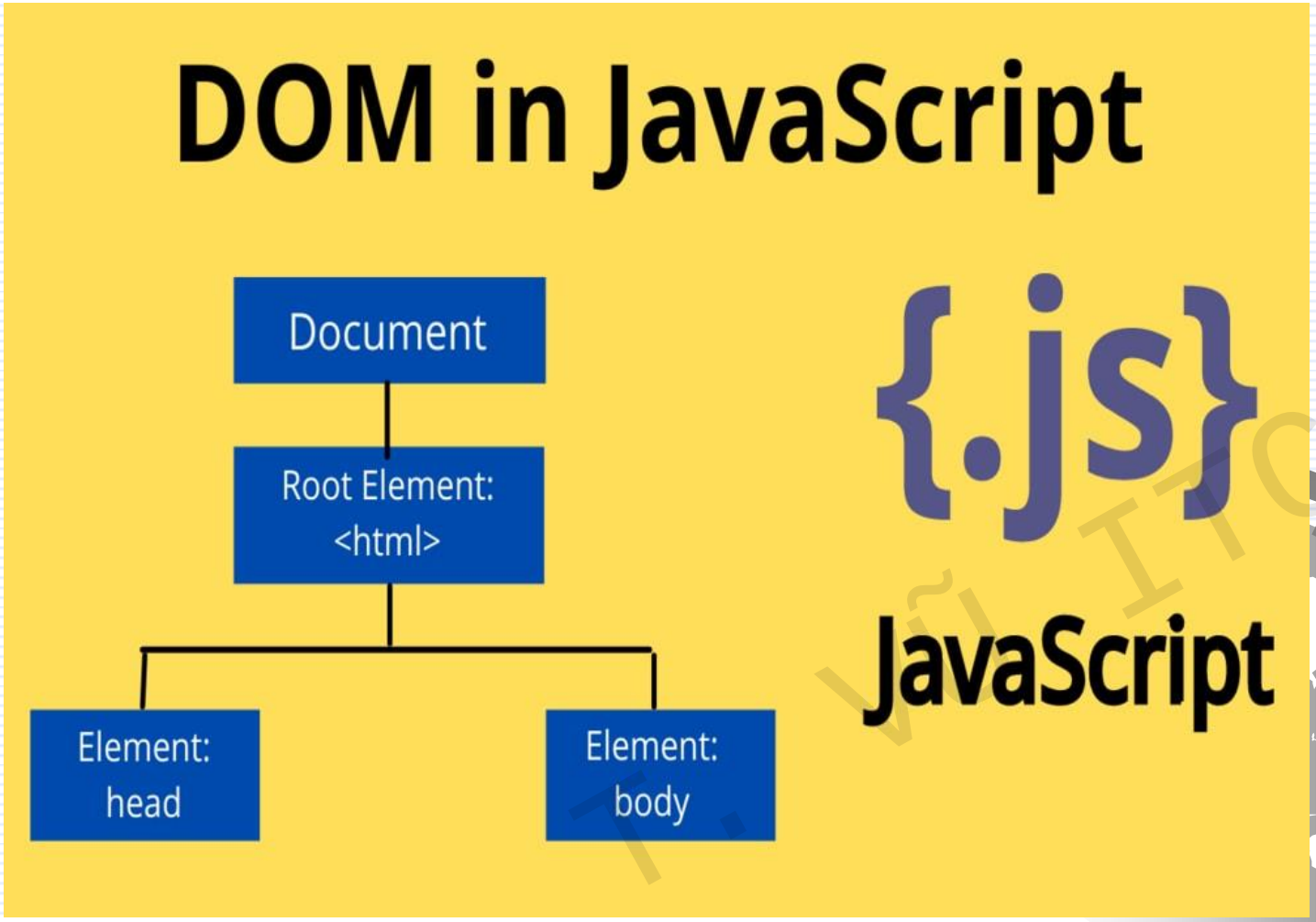
```
<html>
<head>
</head>
<body>
<form>
<input type="text"  onblur="alert('Sự kiện ONBLUR')"
        onclick="alert('Sự kiện ONCLICK');"
                onchange="alert('Sự kiện ONCHANGE');"
                        onkeypress="alert('Sự kiện KEYPRESS');"/>
</form>
</body>
</html>
```

```
<script LANGUAGE="JavaScript">
function a_plus_b(form) {
a=eval(form.a.value)
b=eval(form.b.value)
c=a+b
form.ans.value = c
}
                    function a_minus_b(form) {
                    a=eval(form.a.value)
                    b=eval(form.b.value)
                    c=a-b
                    form.ans.value=c
                    }
<body>
<h3>CHƯƠNG TRÌNH TÍNH TOÁN </h3>
<form name="formx">
Nhập số a : <input type="text" value="12" name="a"> <br/>
Nhập số b : <input type="text" value="12" name="b"> <br/>
<input type="button" value="  +  "   onClick="a_plus_b(this.form)">
<input type="button" value="  -  "   onClick="a_minus_b(this.form)">
<input type="button" value="  x  "   onClick="a_times_b(this.form)">
<input type="button" value="  /  "   onClick="a_div_b(this.form)">
<input type="button" value="  ^  "   onClick="a_pow_b(this.form)"> <br/>
Kết quả : <input  type "text" value="0" name="ans" >
</form>
</body>
```

Ví dụ event

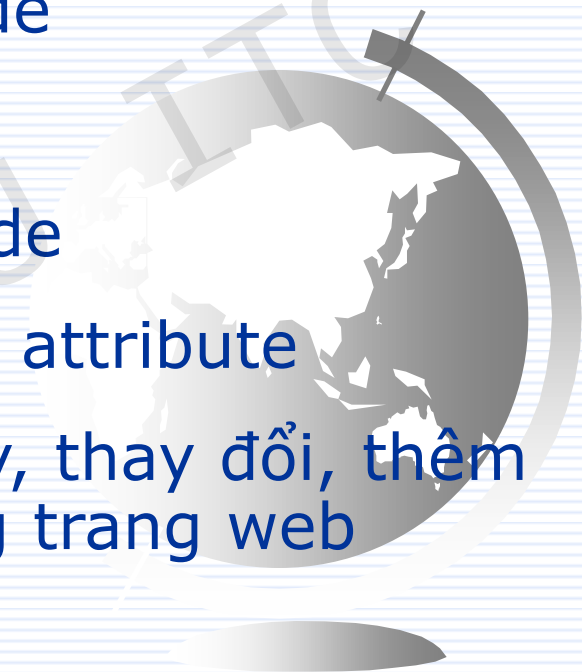


PHẦN 2. MÔ HÌNH DOM



Nội dung

- DOM (Document Object Model) : Là một mô hình chuẩn cho phép ngôn ngữ lập trình có thể truy xuất và thay đổi động nội dung, cấu trúc, định dạng của 1 văn bản.
- HTML DOM : là mô hình chuẩn cho văn bản HTML
  - Toàn bộ trang là document node
  - Mỗi thẻ là 1 HTML node
  - Văn bản trong 1 thẻ là text node
  - Các thuộc tính trong thẻ là các attribute
- Thông qua HTML DOM, ta có thể lấy, thay đổi, thêm hay xóa bất kỳ 1 element nào trong trang web

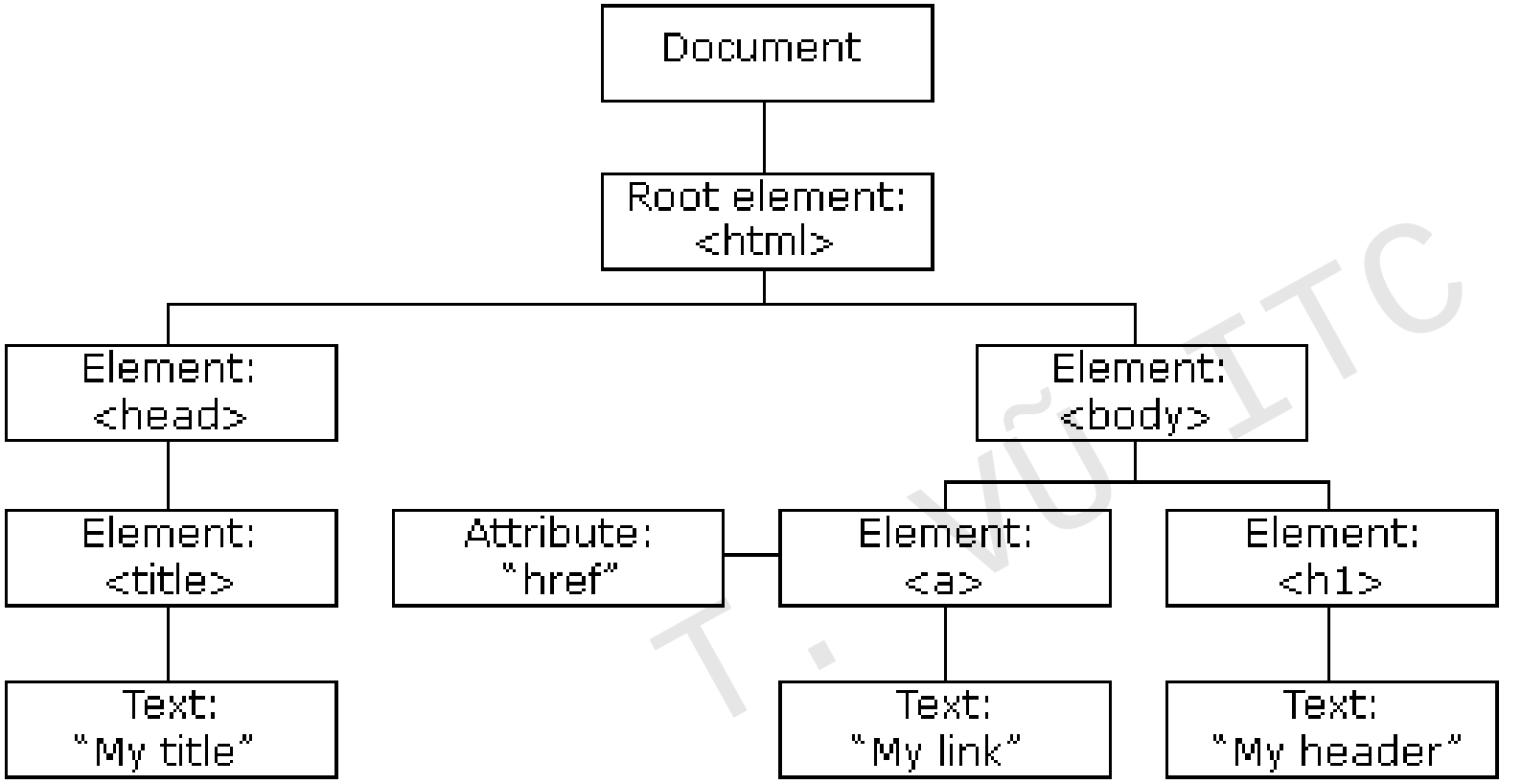


Ví dụ HTML DOM

```
<html>
        <head>
                <title>My Title</title>
        </head>
<body>
        <a href="mypage.htm">my
link</a>
        <h1> My header</h1>
</body>
</html>
```

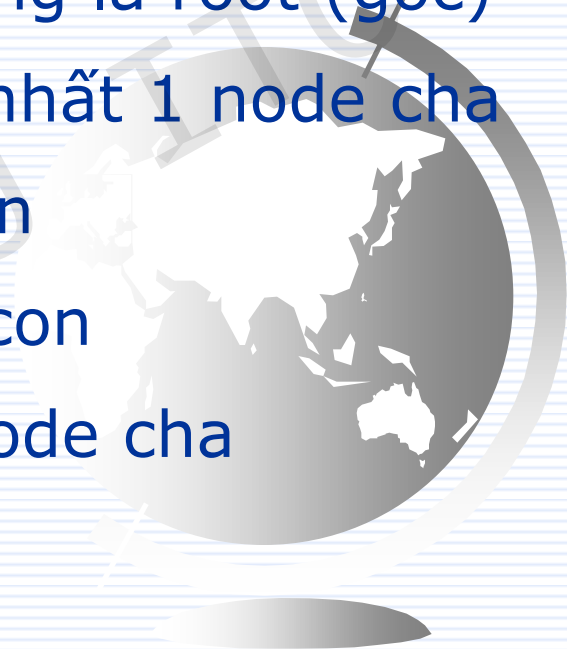
Ví dụ Cây HTML

- HTML DOM coi trang HTML là 1 cây

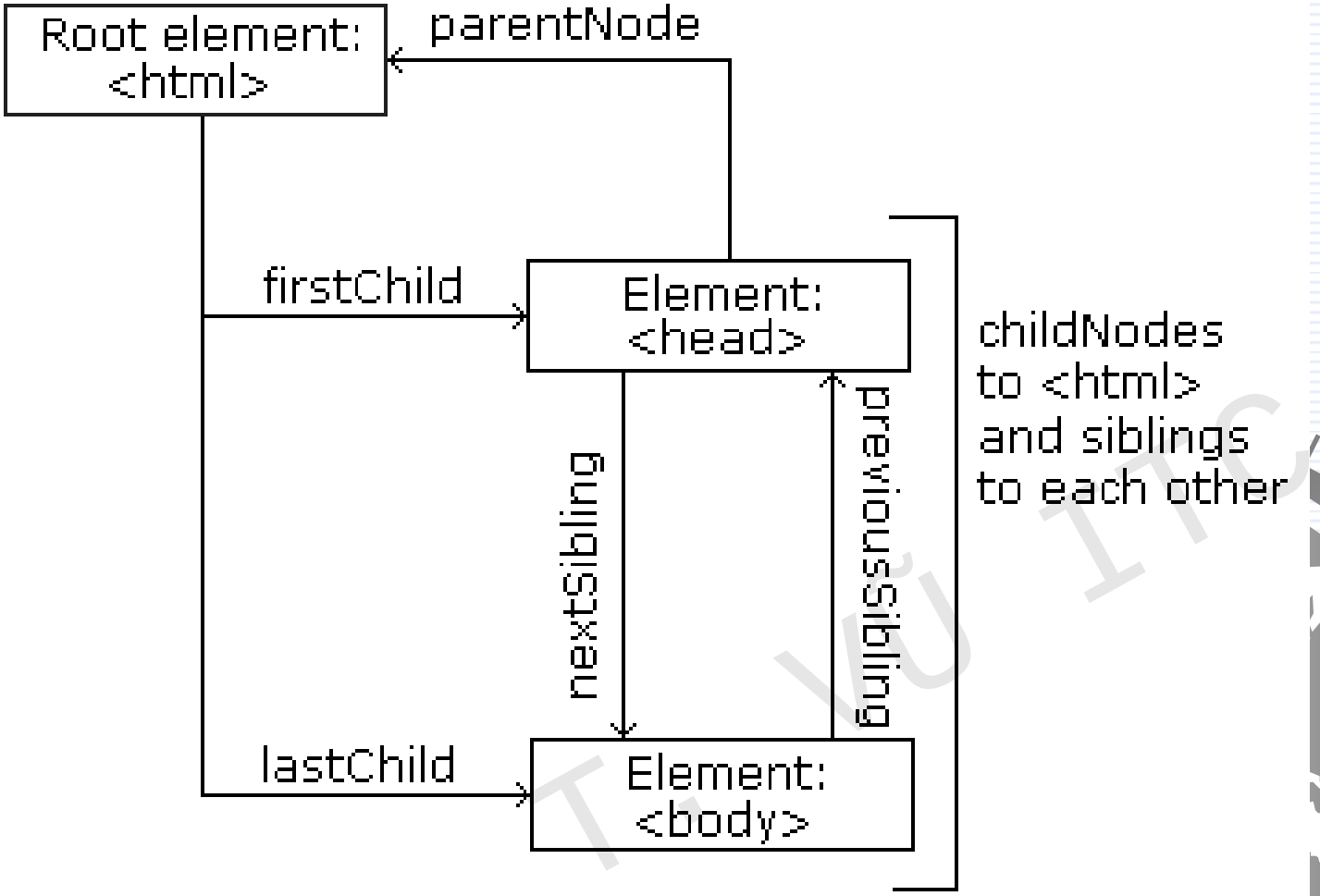


Quan hệ giữa các node

- Các node trong cây HTML có mối quan hệ phân cấp với nhau
- Các từ cha, con, anh em dùng để mô tả các mối quan hệ này. Các node cha có các node con, các node anh em là các node có cùng cấp
- Trong 1 cây HTML, node trên cùng là root (gốc)
- Mọi node (trừ root) đều có duy nhất 1 node cha
- 1 node có thể có nhiều node con
- Node lá là node không có node con
- Node anh em là node có cùng node cha



Mô hình quan hệ giữa các node



Quan hệ giữa các node

```
<html>
  <head>
    <title>DOM Tutorial</title>
  </head>
  <body>
    <h1>DOM Lesson one</h1>
    <p>Hello world!</p>
  </body>
</html>
```

Lấy 1 node

Mỗi node trong HTML là 1 đối tượng. Ta có thể lấy đối tượng bằng cách :

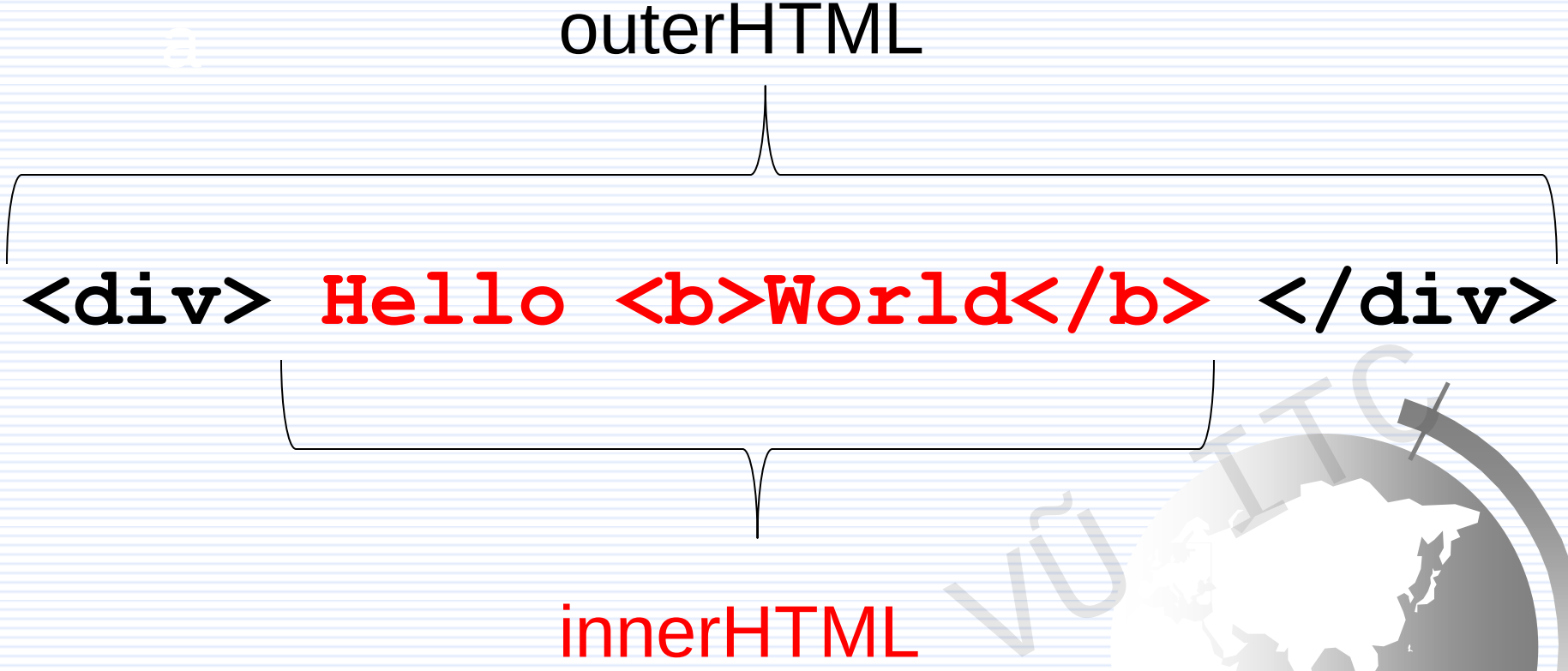
- X.getElementById (id) : lấy element có id cung cấp trong node X
- X.getElementByTagName(name) : lấy danh sách các element có name cung cấp trong node X

Thuộc tính 1 node

- X.innerHTML : văn bản trong X
- X.nodeName : tên của X
- X.nodeValue : giá trị của X
- X.parentNode : node cha của X
- X.childNodes : các node con của X
- X.attributes : các thuộc tính của X
- X.firstChild : node con đầu tiên của X
- X.lastChild : node con cuối của X

Trong đó X là 1 node trong HTML DOM

innerHTML vs outerHTML



Ví dụ 1 : innerHTML

```
1  <html>
2  <body>
3
4  <p id="intro">Hello World !</p>
5
6  <script type="text/javascript">
7  txt=document.getElementById("intro");
8  txt.innerHTML = txt.innerHTML + " Thiet ke web 1";
9  </script>
10
11 </body>
12 </html>
```

Thêm, xóa 1 node

- X.appendChild (Y) : thêm node Y vào làm con node X
- X.removeChild (Y) : xóa node Y ra khỏi con node X
- document.createTextNode(“Text”) : tạo 1 node văn bản
- document.createElement(TagName) : tạo 1 node có là thẻ

Làm việc với thuộc tính

Ta có thể thêm, xóa, cập nhật thuộc tính của 1 node như sau :

- X.getAttribute("Attribute name") : Lấy giá trị 1 attribute
- X.setAttribute("Attribute name","Attribute value") : Thêm hay set lại giá trị 1 attribute
- X.removeAttribute("Attribute name") : Xóa 1 attribute

Ví dụ 2 :

```
1  <html>
2  <body>
3  <div id="cont"><p id="intro_0">Hello World 0</p></div>
4  <input type="button" value="+" onclick="add()"/>
5  <input type="button" value="-" onclick="remove()"/>
6  <script type="text/javascript">
7  i = 1;
8  function add()
9  {
10     p = document.createElement("p");
11     p.setAttribute("id","intro_"+i);
12     p.innerHTML = "Hello World " + i;
13     d = document.getElementById("cont");
14     d.appendChild(p);
15     i++;
16 }
17 function remove()
18 {
19     d = document.getElementById("cont");
20     d.removeChild(d.lastChild);
21     i--;
22 }
23 </script>
24 </body>
25 </html>
```

Định dạng node

- Các node biểu diễn 1 thẻ đều có thuộc tính style dùng để định dạng
- Cách thay đổi định dạng 1 node :

```
document.getElementById("id").style.property="value"
```

Thay đổi background

Property	Description	W3C
background	Sets all background properties in one	Yes
backgroundAttachment	Sets whether a background-image is fixed or scrolls with the page	Yes
backgroundColor	Sets the background-color of an element	Yes
backgroundImage	Sets the background-image of an element	Yes
backgroundPosition	Sets the starting position of a background-image	Yes
backgroundPositionX	Sets the x-coordinates of the backgroundPosition property	No
backgroundPositionY	Sets the y-coordinates of the backgroundPosition property	No
backgroundRepeat	Sets if/how a background-image will be repeated	Yes

Ví dụ 3

```
1  <html>
2  <body>
3  <p id="intro">Hello World 0</p>
4  <input type="button" value="Background"
5         onclick="changeBackground()"/>
6  <input type="button" value="Reset"
7         onclick="removeBackground()"/>
8  <script type="text/javascript">
9  function changeBackground()
10 {
11     p = document.getElementById("intro");
12     p.style.background = "#C7c7c7 url(smiley.gif)";
13 }
14 function removeBackground()
15 {
16     p = document.getElementById("intro");
17     p.style.background = "";
18 }
19 </script>
20 </body>
21 </html>
```

Thay đổi border, margin, padding

Property	Description	W3C
border	Sets all properties for the four borders in one	Yes
margin	Sets the margins of an element (can have up to four values)	Yes
padding	Sets the padding of an element (can have up to four values)	Yes

Ví dụ 3

```
<body>
<h1 align="center">MÔ HÌNH DOM - VÍ DỤ 3</h1>

 <p id="intro"> Hello world 0 </p>
<input type="button" value="Increase Border" onclick="increase('border')" />
<input type="button" value="Increase Padding" onclick="increase('padding')" />
<input type="button" value="Increase Margin" onclick="increase('margin')" />
<script type="text/javascript">
border=0;
padding=0;
margin=0;
function increase(type)
{
    d = document.getElementById("intro");
    if(type=='border')
    {
        border ++;
        d.style.border = "solid " + border + "px";
    }
    else if(type=='padding')
    {
        padding = padding + 5;
        d.style.padding = padding + "px";
    }
    else if(type=='margin')
    {
        margin += margin + 5;
        d.style.margin = margin + "px";
    }
}
</script>
```


Thay đổi font

Property	Description	W3C
color	Sets the color of the text	Yes
font	Sets all font properties in one	Yes
fontFamily	Sets the font of an element	Yes
fontSize	Sets the font-size of an element	Yes
fontSizeAdjust	Sets/adjusts the size of a text	Yes
fontStretch	Sets how to condense or stretch a font	Yes
fontStyle	Sets the font-style of an element	Yes
fontVariant	Displays text in a small-caps font	Yes
fontWeight	Sets the boldness of the font	Yes
letterSpacing	Sets the space between characters	Yes
lineHeight	Sets the distance between lines	Yes

Kham khảo

- Javascript Tutorial : http://www.w3schools.com/JS/default.asp
- HTML DOM Tutorial : http://www.w3schools.com/HTMLDOM/default.asp
- Javascript & DOM example : http://www.w3schools.com/JS/js_ex_dom.asp
- HTML DOM Style Object : http://www.w3schools.com/jsref/dom_obj_style.asp

Hàm thiết lập thời gain

- **setTimeout**(“javascript command”, delayTime) : hàm cho phép thực thi 1 lần 1 câu lệnh hay hàm sau 1 khoảng thời gian tính bằng milisecond. Hàm trả về 1 **id** và **id** này dùng cho hàm clearTimeOut() để dừng việc thực thi
- **clearTimeout(id)** : dừng việc thực thi
- **setInterval**(“javascript command”, delayTime) : hàm cho phép thực thi liên tục sau 1 khoảng thời gian
- **clearInterval(id)** : dừng việc thực thi liên tục

Ví dụ

```
<body>
<input type="button" value="Stop hello" onclick="stopHello()"/>
<script type="text/javascript">
id = setTimeout("sayHello()",1000);
function sayHello()
{
    alert("Hello world");
}
function stopHello()
{
    clearTimeout(id);
}
</script>
```

```
<body>
<input type="button" value="Stop hello" onclick="stopHello()"/>
<script type="text/javascript">
id = setInterval("sayHello()",4000);
function sayHello()
{
    alert("Hello world");
}
function stopHello()
{
    clearInterval(id);
}
</script>
</body>
```


QUESTIONS