



FPT POLYTECHNIC



Bài 2: BI N VÀ TOÁN T

- Giới thiệu về JavaScript
- Lớp học JavaScript
- Mục đích của JavaScript
- JavaScript và Browser
- Chương trình JavaScript
- Vị trí mã JavaScript
- Sắp xếp mã JavaScript
- IDE cho JavaScript

- Biến và khai báo biến
- Kiểm thử lối
- Ép kiểu
- Toán tử
- Cấu trúc điều khiển
- Lệnh lachenn
- Debug

BI N, KI U D LI U
& TOÁN T

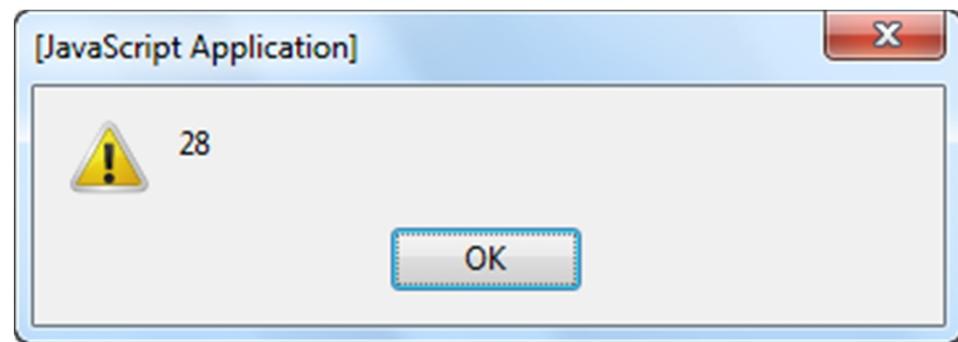
- Biến dùng lục trắc liệu
- Cú pháp **var tenbien;**
- Cách đặt tên biến
 - Tên biến bao gồm cách cái và số, không có bớt dấu số
 - Tên biến không bao gồm dấu cách và dấu câu, ngoại trừ dấu gạch dưới (_)
- Có thể khai báo nhiều biến trên một dòng
 - var x, y, zeta;
- Có thể và khai báo và khởi tạo giá trị cho biến
 - var x = 1;
 - var x = 1, y = "hello";

■ JavaScript h tr nh ng ki u d li u sau

- Ki ú s
- Ki u chu i ký t
- Ki u Boolean
- Ki u m ng
- Object

- JavaScript không chia ra kiểu Double, Integer... như các ngôn ngữ khác.
- Java gộp tất cả thành một kiểu duy nhất là kiểu số
 - var 20
 - var 1.5
 - var 0xd

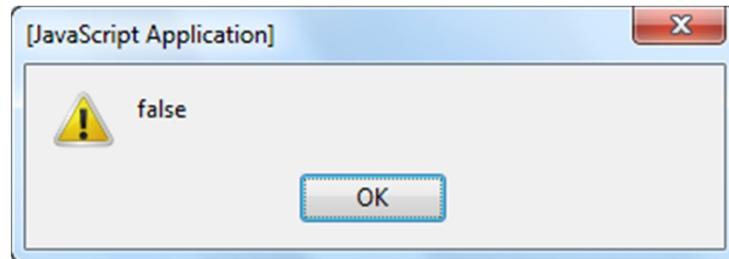
```
<html>
<head>
<title>Hexadecimal Numbers</title>
<script type="text/javascript">
var h = 0xe;
var i = 0x2;
var j = h * i;
alert(j);
</script>
</head>
<body>
<p>Chapter 4 Example</p>
</body>
</html>
```



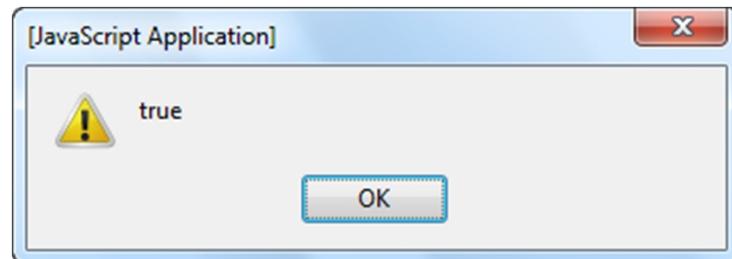
■ isNaN()

- Xác định xem tham số truy cập vào có phải là số hay không
- Nếu là số, trả về False
- Nếu không phải là số, trả về True

■ alert(isNaN("4"))



■ alert(isNaN("four"))



- Javascript cung cấp i t ng Math thao tác v i d li u ki us

Phương thức	Giải thích	Ví dụ	Kết quả
Math.random()	Trả về số ngẫu nhiên trong khoảng từ 0 đến 1		
Math.abs(x)	Trả về giá trị tuyệt đối của x	Math.abs(-2)	2
Math.pow(x,y)	Trả về giá trị của x mũ y	Math.pow(2,3)	8
Math.round(x)	Trả về giá trị làm tròn của x	Math.round(2.6)	3

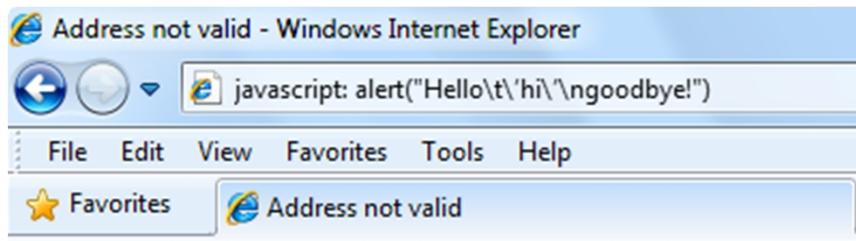
- String c bao quanh bởi cặp dấu nháy kép

- "Hello world"

- Ký tự có biệt

Ký tự	Giải thích
\'	'
\"	"
\b	Dấu cách
\t	Dấu tab
\n	Xuống dòng

- alert("Hello\t\'hi\'\\ngoodbye!")



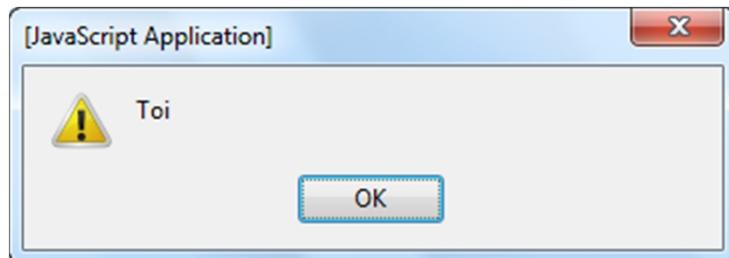
■ Thuộc tính length

- var x = "Tôi là String."; alert(x.length);



■ Phương thức subString

- var x = "Tôi là String."; alert(x.substring(0,3));



- Ph ng th c concat

```
var firstString = " Day la mot xau ky tu ";
var finalString = firstString.concat(" Them mot xau ky tu nua ");
alert(finalString);
```

- Ph ng th c toUpperCase

```
var firstString = " Day la mot xau ky tu ";
var finalString = firstString.toUpperCase();
alert(finalString);
```

- Ph ng th c toLowerCase

```
var firstString = " Day la mot xau ky tu ";
var finalString = firstString.toLowerCase();
alert(finalString);
```

- Kiến thức Boolean có hai giá trị là true và false
- Các biến thuộc Boolean thường được sử dụng trong các cấu trúc điều kiện

```
if (x > 18) {  
    alert("Hi");  
}
```

- M ng là ki u d li u dùng l u m t t p các d li u có ki u gi ng nhau
- Cú pháp khai báo và gán giá tr cho m ng
 - Cách th nh t

```
var convat = new Array();  
convat[0] = "meo";  
convat[1]= "ho";  
convat[2] = "voi";
```

- Cách th hai

```
var convat = new Array("meo", "ho", "voi");
```

■ Giá tr Null

- Null nghĩa là không gì cả
- var x = null; nghĩa là x không tham chiếu nowhere
- var x = null; khác với var x = "";

■ Kiểu Undefined

- Chắc chắn giá trị

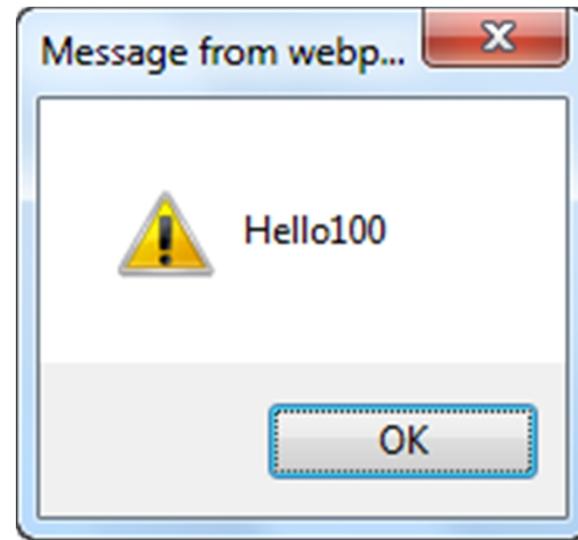
```
var x;  
alert(x);
```



■ Ép kiểu số

- Trình thông dịch tự động chuyển kiểu

```
var x = 100;  
alert("Hello" + x);
```



- Ép kiểu tự động minh

- Ép kiểu số thành chuỗi

```
var x = String(100);  
alert(typeof(x));
```

- Ép kiểu chuỗi thành số

```
var x = “100”;  
var y = Number(x);  
alert(typeof(y));
```

■ Toán tử số học

Toán tử	Giải thích
+	Cộng
-	Trừ
*	Nhân
/	Chia
%	Chia lấy dư

■ Toán tử logic

Toán tử	Giải thích
&	Và
	Hoặc
^	XOR
!	NOT

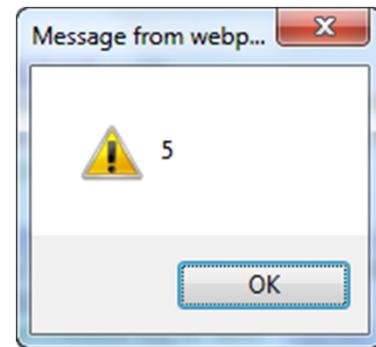
■ Toán tử quan hệ và toán tử b

Toán tử	Giải thích
>	Lớn hơn
<	Bé hơn
\geq	Lớn hơn hoặc bằng
\leq	Bé hơn hoặc bằng
$=$	Bằng
\neq	Khác

■ Toán tử m t ngôi

Toán tử	Giải thích
+	Chuyển toán hạng sang số dương
-	Chuyển toán hạng sang số âm
++	Tăng
--	Trừ

```
var x = 4;  
x++;  
alert (x);
```



- Số khác nhau giữa $++x$ và $x++$

```
var x = 4;
```

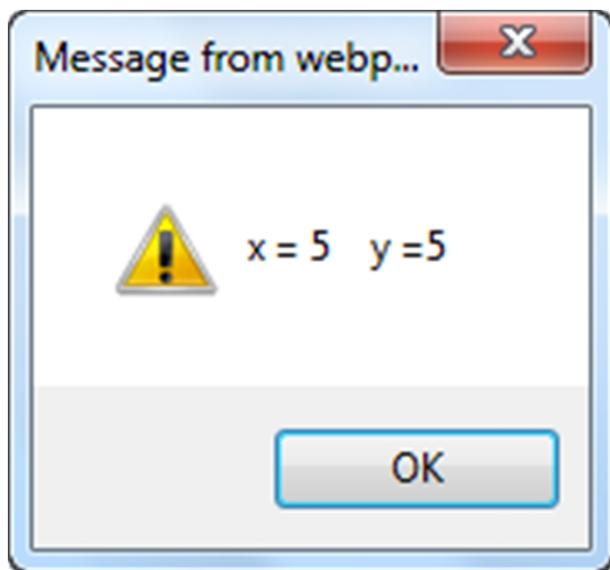
```
var y = ++x;
```

```
alert ("x = "+x+" y =" +y);
```

```
var x = 4;
```

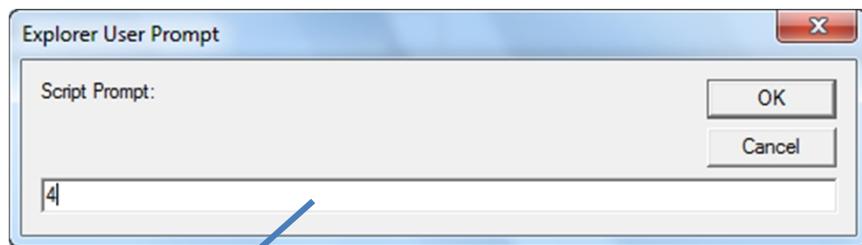
```
var y = x++;
```

```
alert ("x = "+x+" y =" +y);
```

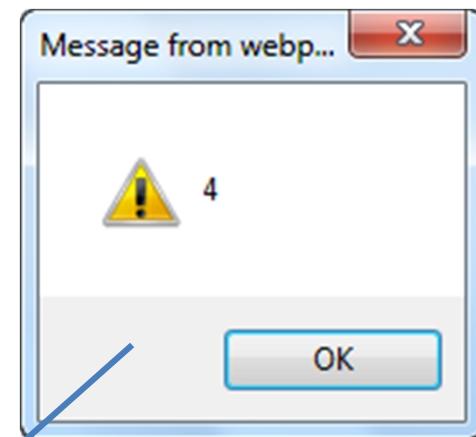


- Dùng I y d li u do ng i dùng nh p vào
- Hàm tr v giá tr mà ng i dùng ā nh p
- o n mā sau s I y d li u do ng i dùng nh p vào và hi n th k t qu trong h p tho i:

```
var x = prompt();  
alert (x);
```



Cửa sổ cho người dùng
nhập thông tin



Hiển thị giá trị người
dùng đã nhập

C Ư TRÚC | U KHI N

- C u trúc tu n t
- C u trúc l a ch n
- C u trúc l p

- L nh l a ch n n
- L nh l a ch n kép
- L nh a l a ch n

- Cú pháp

```
if (điều_kiện) {  
    //Thực hiện  
}
```

- Các trường hợp

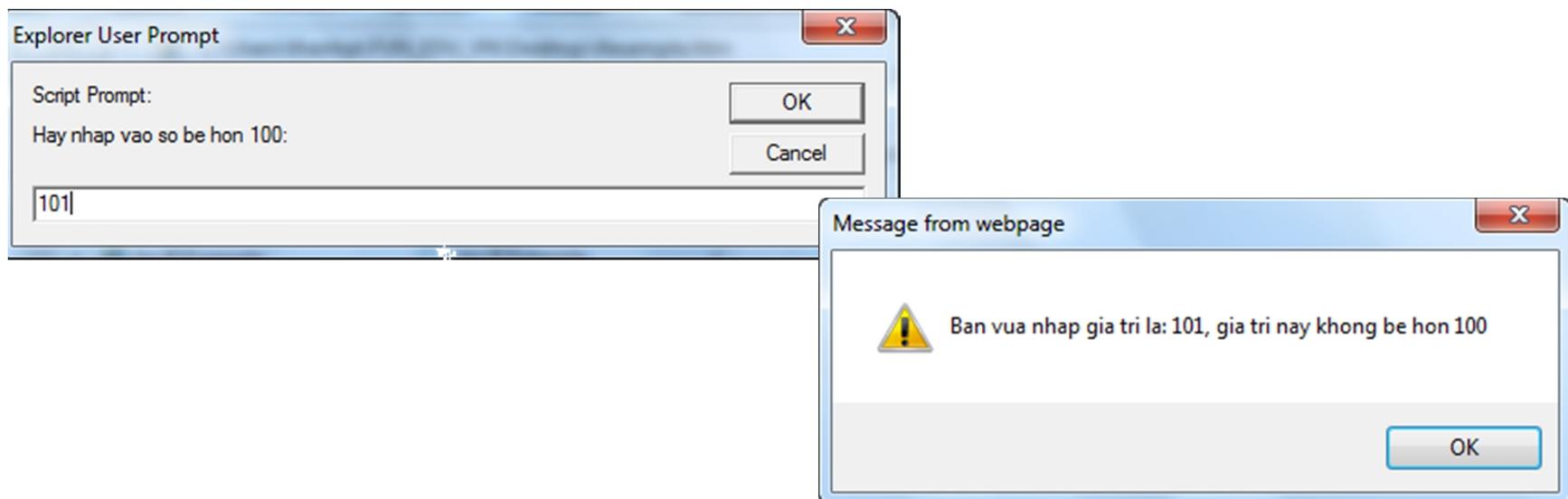
```
if (! điều_kiện) {  
    //Thực hiện  
}
```

```
if (x < 350) {  
    //Thực hiện  
}
```

```
var x = 3;  
var y = 4;  
if (x == y) {  
    //Thực hiện  
}
```

Ví d l nh lach n If...Then

```
var x = prompt("Hay nhap vao so be hon 100:");
if (x >= 100)
{
    alert("Ban vua nhap gia tri la: " + x + ", gia tri nay khong
be hon 100");
}
```

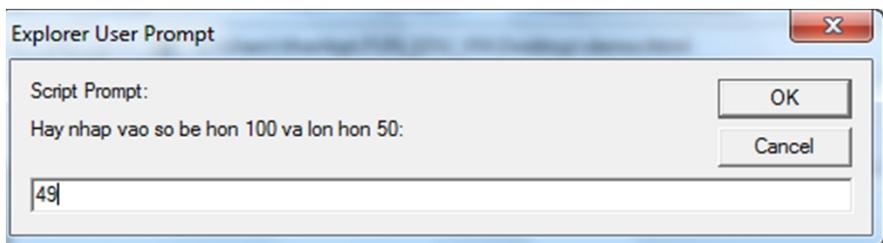


- Dùng toán tử logic để kiểm tra xem nh

thành mним nh ph c t ph n

```
var x = prompt("Hay nhap vao so be hon 100 va lon hon 50:");
```

```
if ((x < 51) || (x > 99)) {  
    document.write("<h1>Ban vua nhap gia tri la: " + x + "</h1>");  
    document.write("<h2>Gia tri nay khong nam trong khoang tu 50  
den 100</h2>");  
}
```



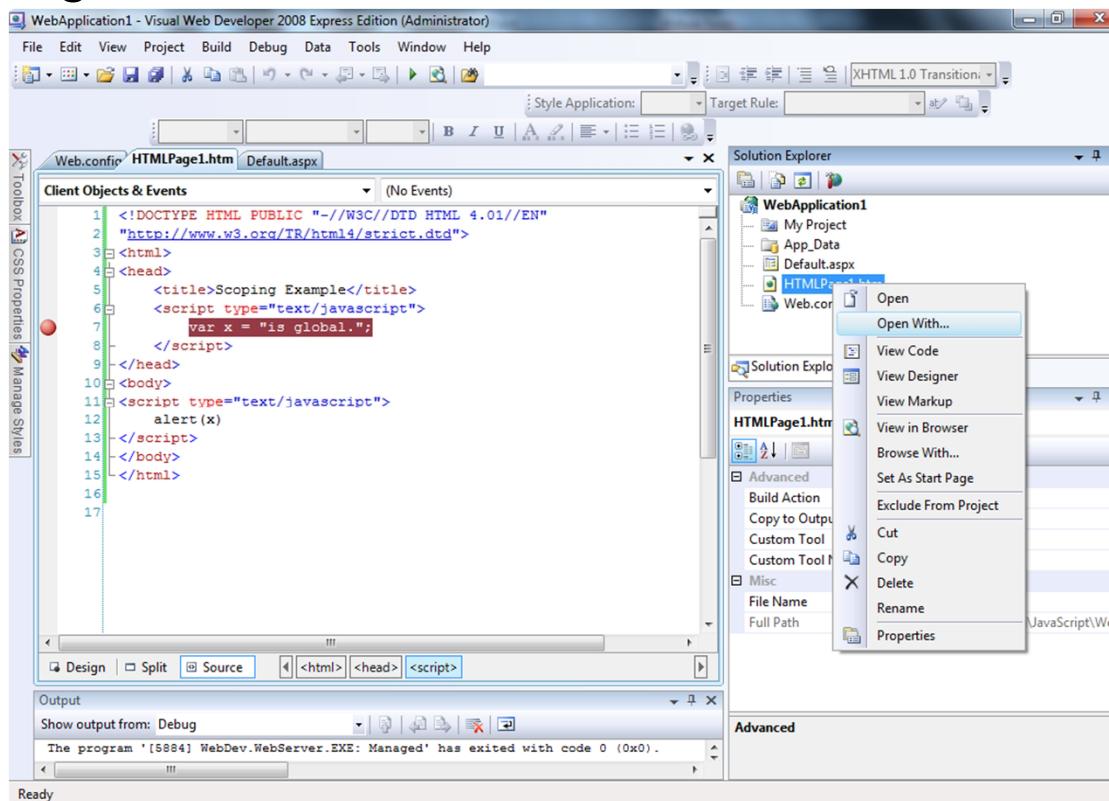
- Công c g l i giúp ng i l p trình tìm và s a l i (bao g m l i lôgíc và l i biên d ch)
- Vì mā JavaScript c thông d ch trên trình duy t, nên m i trình duy t s xây d ng b g l i riêng.
- IE:
 - Microsoft Script Debugger (Debug ngay trên trình duy t IE)
 - Visual Studio
- Firefox
 - Firebug

DEBUG

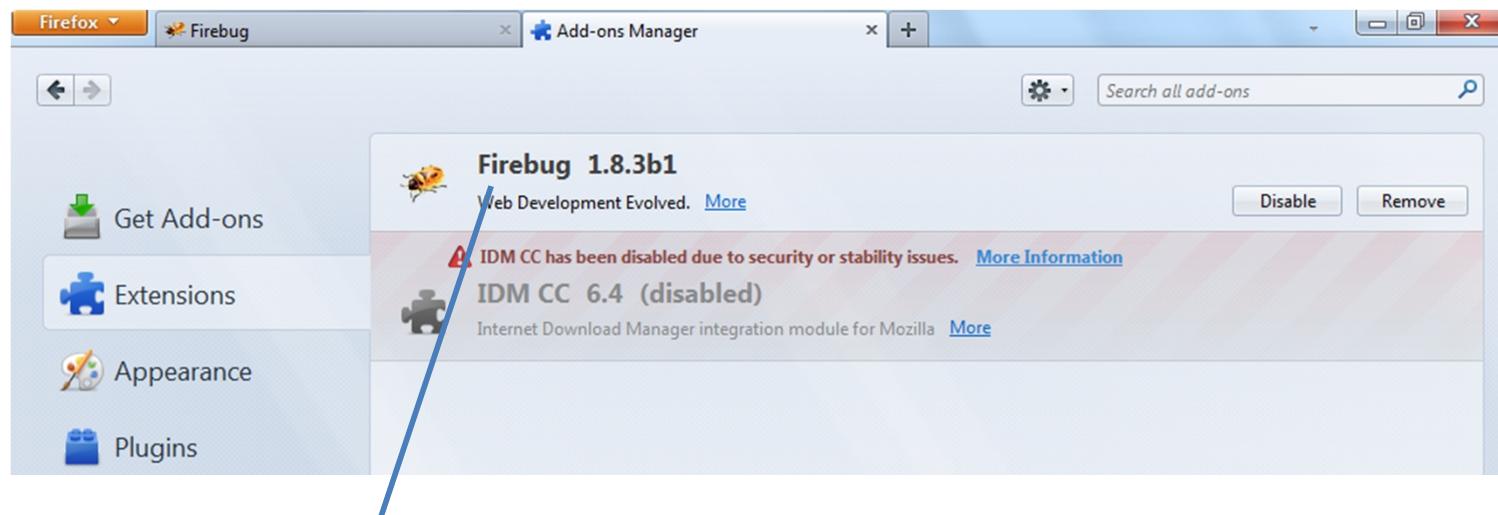


Sử dụng Visual Studio để mở

- Mở file bằng IE: Click vào file → Chọn Open With → Chọn Internet Explorer → Visual Studio sẽ chuyển sang chế độ mở



- Vào trang web <http://www.getfirebug.com>
- Nhấn vào button Install Firebug
- Download bản Firebug phù hợp với version của trình duyệt Firefox đang sử dụng

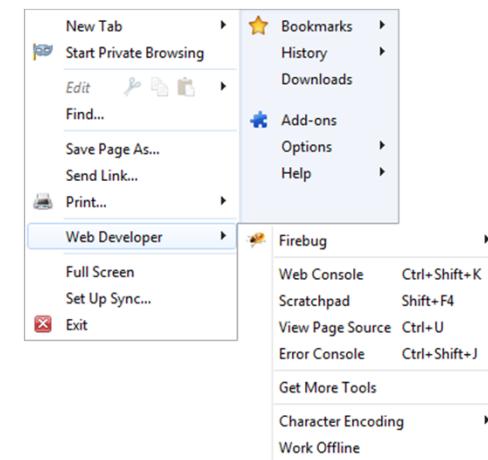
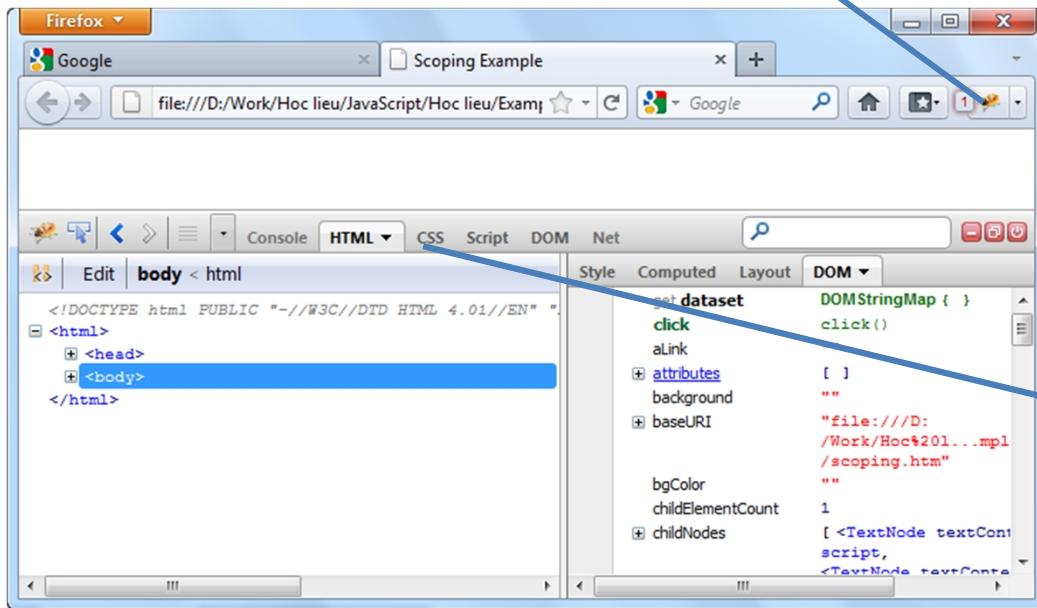


Firebug đã được cài đặt

■ Bắt đầu Firebug

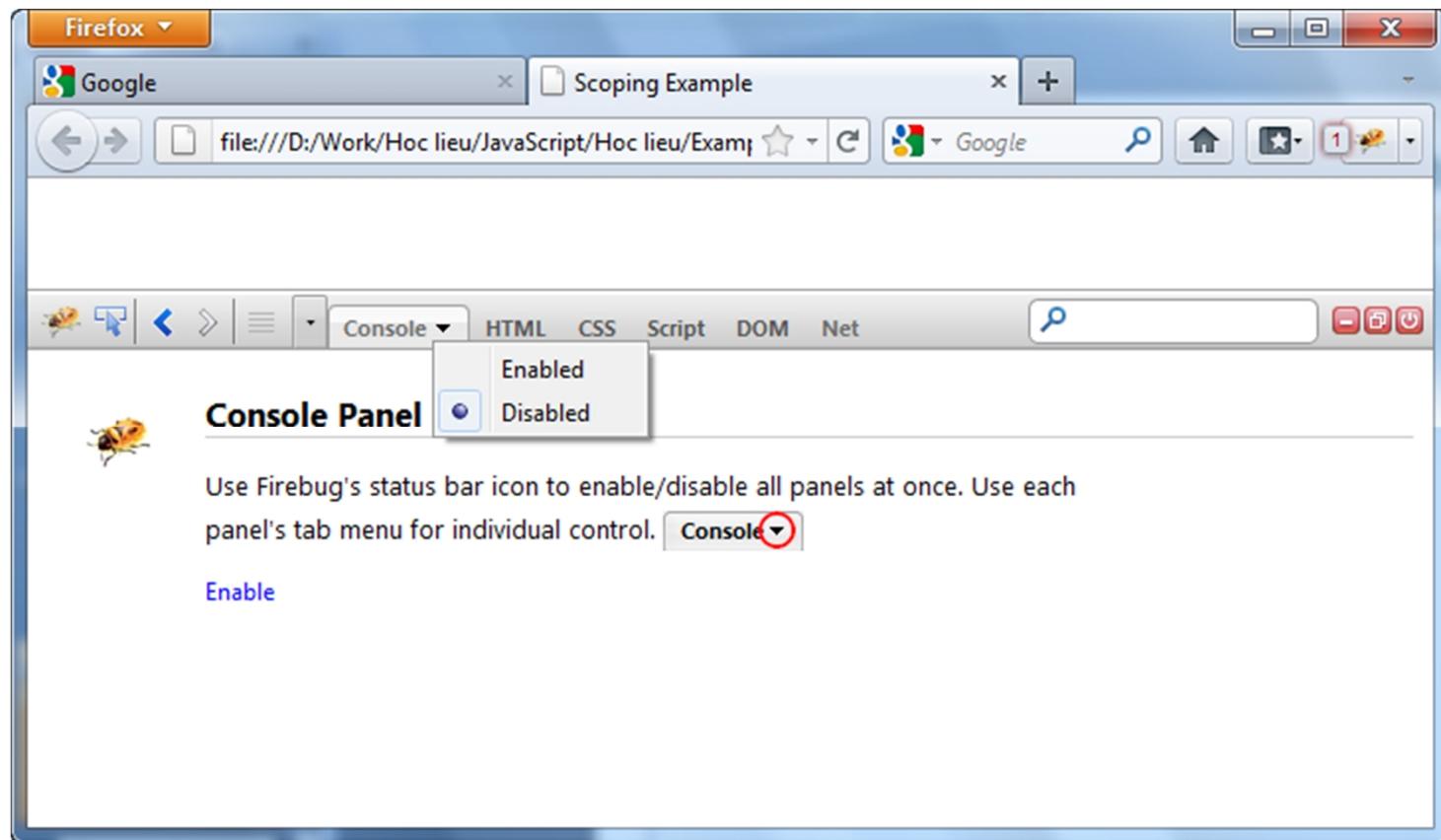
- Mở Firefox
- Bắt đầu Firebug bằng cách nhấp vào biểu tượng Firebug hoặc vào tệp Menu của Firefox như hình bên

Biểu tượng Firebug



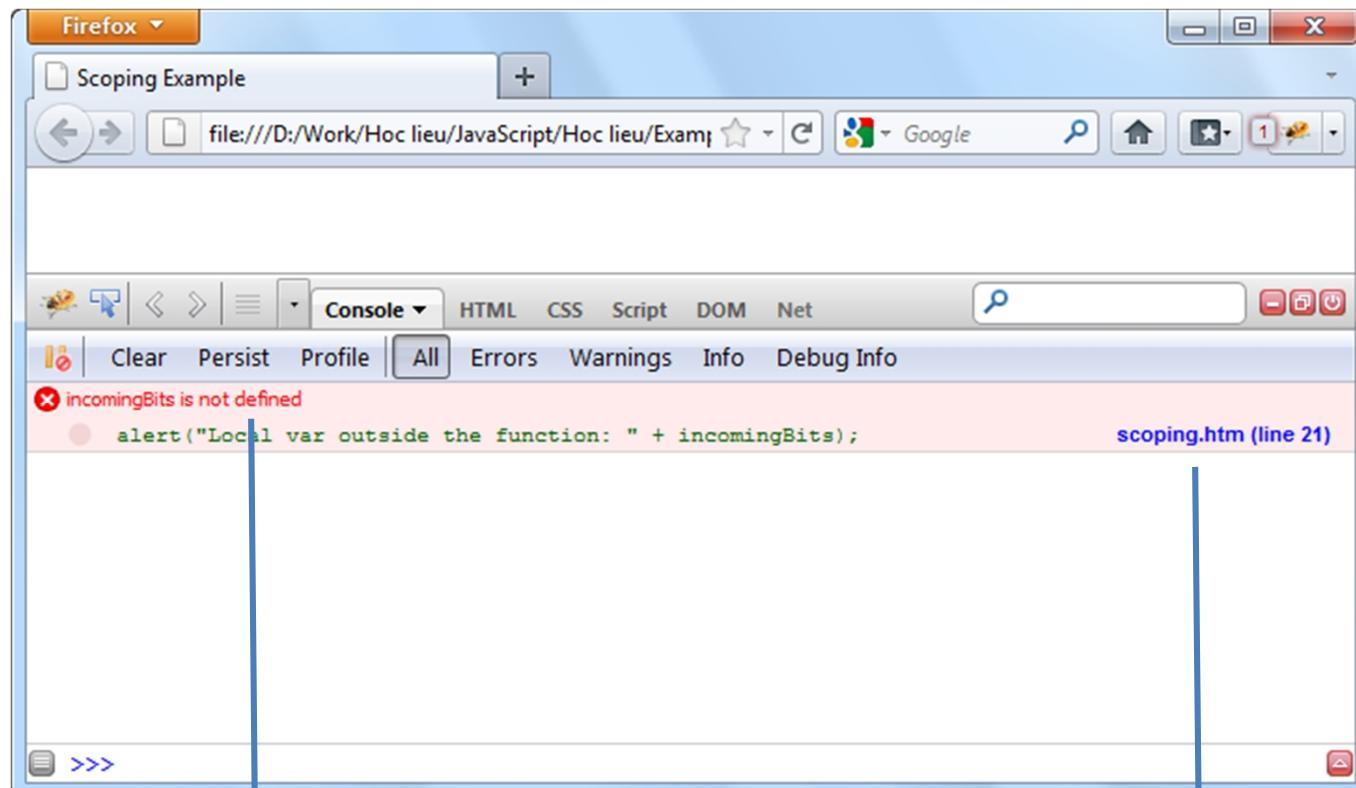
Cửa sổ Firebug xuất hiện

- Bước 1: Nhấn vào mũi tên bên phải Console, chọn Enable



■ G | i

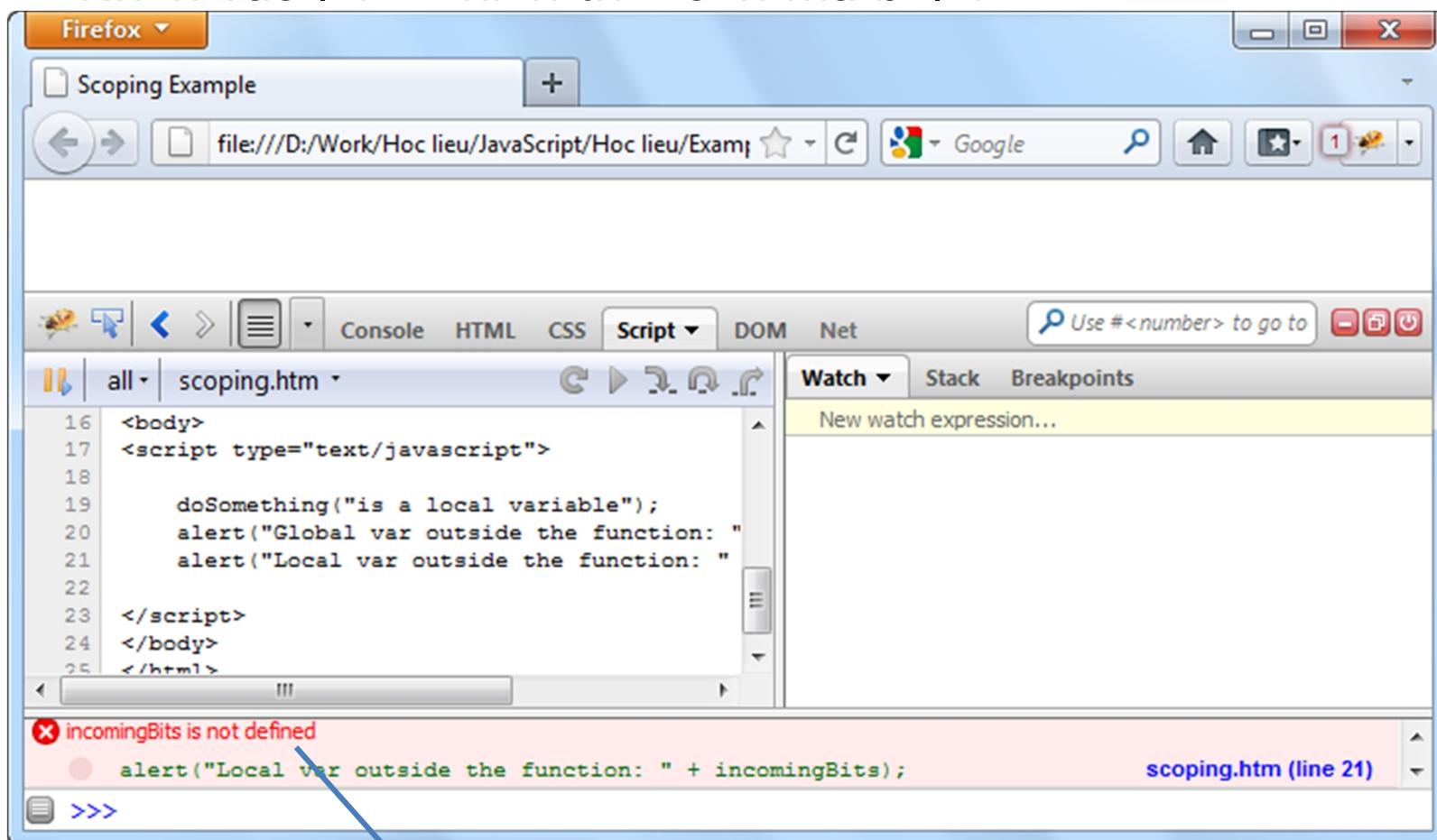
- M trang scoping.html



Cửa sổ thông báo lỗi

Lỗi ở dòng 21

- Nhập vào lỗi hiển thị trên mã bài



Cửa sổ thông báo lỗi

G ı i b ng Microsoft Script Debugger

- BTVN: Tìm hiểu các giao tiếp trên IE bằng Microsoft Script Debugger

- JavaScript hỗ trợ các kiểu biến: Kiểu số, kiểu String, kiểu Boolean và kiểu Object
- Kiểu mảng lưu các dữ liệu cùng kiểu với nhau
- JavaScript hỗ trợ giá trị null và undefined. Null nghĩa là biến không tham chiếu đến đâu cả, còn undefined nghĩa là biến chưa xác định
- JavaScript cũng có hai kiểu ép kiểu: Ép kiểu sang minh và ép kiểu sang mờ nh
- Biến được khai báo bằng từ khóa var, sau đó là tên biến
- JavaScript hỗ trợ các toán tử lôgic, toán tử so sánh, toán tử quan hệ và toán tử mảng ngôi

- JavaScript cung cấp 3 cấu trúc điều khiển: Cấu trúc tuần tự, cấu trúc lặp và cấu trúc lồng
- Cấu trúc lồng là lệnh lặp (if)
- Vì mã JavaScript có thông dịch trên trình duyệt, nên mỗi trình duyệt sẽ xây dựng bộ gõ riêng.
- Firebug gõ cho Firefox, Microsoft Script Debugger gõ cho IE. Visual Studio tích hợp bộ gõ cho IE

XIN C M N!