



PHẦN 1

LẬP TRÌNH JAVASCRIPT

BÀI 2: BIẾN, KIỂU, TOÁN TỬ VÀ BIỂU THỨC

- ☐ Giới thiệu về JavaScript
- ☐ Lịch sử JavaScript
- ☐ Mục đích của JavaScript
- ☐ JavaScript và Browser
- ☐ Chương trình JavaScript
- ☐ Vị trí mã JavaScript
- ☐ Sơ lược về mã JavaScript
- ☐ IDE cho JavaScript

- ☐ Biến và khai báo biến
- ☐ Kiểu dữ liệu
- ☐ Ép kiểu
- ☐ Toán tử
- ☐ Cấu trúc điều khiển
- ☐ Lệnh lựa chọn đơn
- ☐ Debug



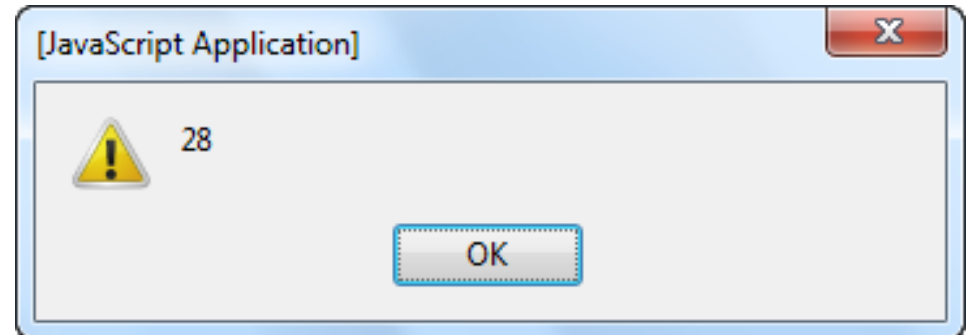
- ❑ Biến dùng để lưu trữ dữ liệu
- ❑ Cú pháp **var tenbien;**
- ❑ Cách đặt tên biến
 - ❖ Tên biến bao gồm chữ cái và số, nhưng không được bắt đầu bằng số
 - ❖ Tên biến không bao gồm dấu cách và dấu câu, ngoại trừ dấu gạch dưới (_)
- ❑ Có thể khai báo nhiều biến trên một dòng
 - ❖ var x, y, zeta;
- ❑ Có thể vừa khai báo vừa khởi tạo giá trị cho biến
 - ❖ var x = 1;
 - ❖ var x = 1, y = "hello";

□ JavaScript hỗ trợ những kiểu dữ liệu sau

- ❖ Kiểu số
- ❖ Kiểu chuỗi ký tự
- ❖ Kiểu Boolean
- ❖ Kiểu mảng
- ❖ Object

- ❑ JavaScript không chia ra kiểu Double, Integer... như các ngôn ngữ khác.
- ❑ Java gộp lại 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>
```

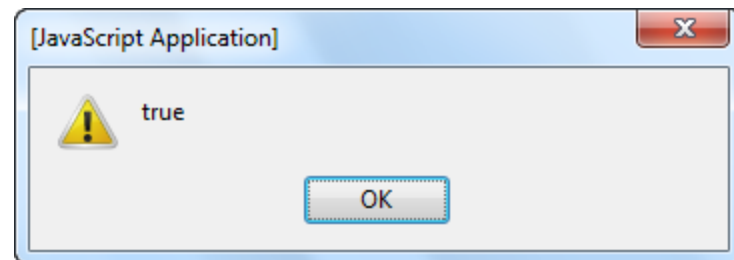
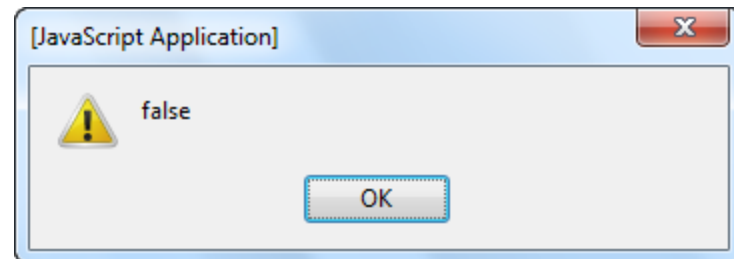


❑ isNaN()

- ❖ Xác định xem tham số truyền 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 lớp Math gồm các phương thức thao tác với dữ liệu kiểu số

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

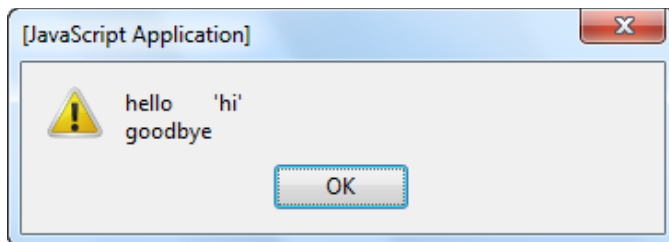
❑ String được bao quanh bởi cặp dấu nháy kép

❖ "Hello world"

❑ Ký tự đặc biệt

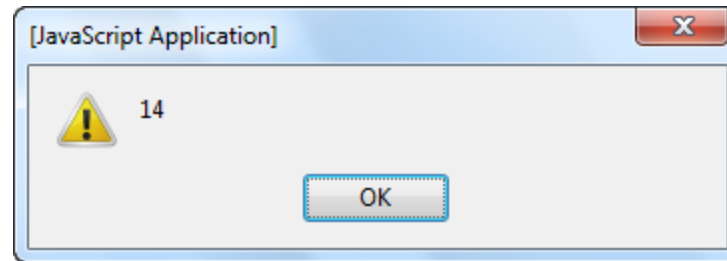
Ký tự	Giải thích
\'	'
\"	"
\b	Dấu cách
\t	Dấu tab
\n	Xuống dòng

❑ `alert("Hello\t'hi'\nngoodbye!")`



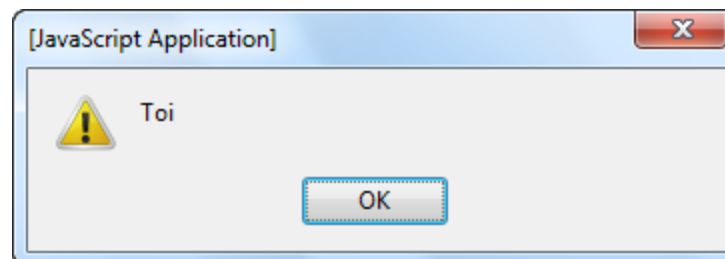
❑ Thuộc tính length

❖ `var x = "Toi la String."; alert(x.length);`



❑ Phương thức subString

❖ `var x = "Toi la String."; alert(x.subString(0,3));`



❑ Phương thức concat

```
var firstString = " Đây là một chuỗi ký tự ";  
var finalString = firstString.concat(" Thêm một chuỗi ký tự nữa ");  
alert(finalString);
```

❑ Phương thức toUpperCase

```
var firstString = " Đây là một chuỗi ký tự ";  
var finalString = firstString.toUpperCase();  
alert(finalString);
```

❑ Phương thức toLowerCase

```
var firstString = " Đây là một chuỗi ký tự ";  
var finalString = firstString.toLowerCase();  
alert(finalString);
```

- ❑ Kiểu Boolean có hai giá trị là true và false
- ❑ Các biểu thức Boolean thường được sử dụng trong các cấu trúc điều khiển

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

- ❑ Mảng là kiểu dữ liệu dùng để lưu một tập các dữ liệu có kiểu giống nhau
- ❑ Cú pháp khai báo và gán giá trị cho mảng

- ❖ Cách thứ nhất

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

- ❖ Cách thứ hai

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

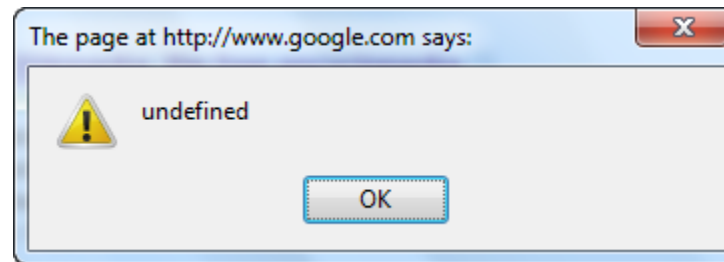
❑ Giá trị Null

- ❖ Null nghĩa là không gì cả
- ❖ `var x = null;` nghĩa là x không tham chiếu đến đâu cả
- ❖ `var x = null;` khác với `var x = "";`

❑ Kiểu Undefined

- ❖ Chưa xác định giá trị

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





PHẦN 2

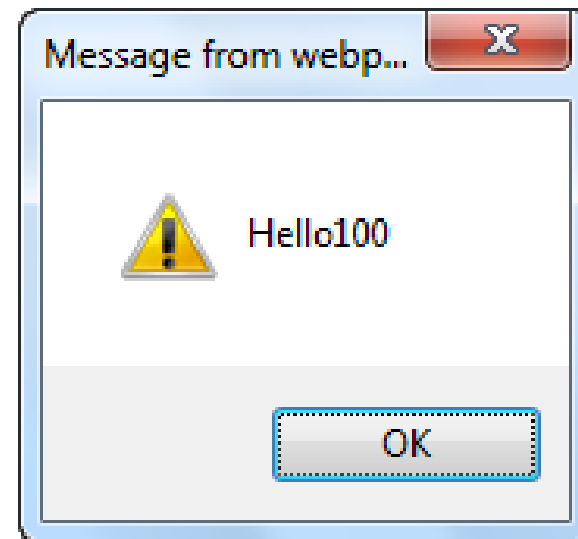
LẬP TRÌNH JAVASCRIPT

BÀI 2: BIẾN, KIỂU, TOÁN TỬ VÀ BIỂU THỨC

❑ Ép kiểu ngầm định

❖ Trình thông dịch tự động chuyển kiểu

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



□ Ép kiểu tường 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

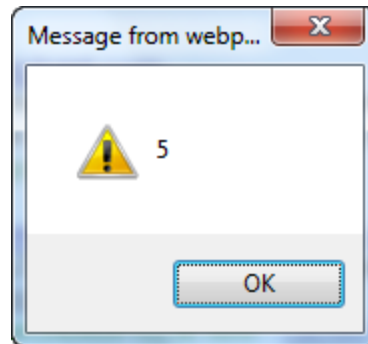
□ Toán tử quan hệ và toán tử bằng

Toán tử	Giải thích
>	Lớn hơn
<	Bé hơn
>=	Lớn hơn hoặc bằng
<=	Bé hơn hoặc bằng
==	Bằng
!=	Khác

□ Toán tử một 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);
```



□ Sự khác nhau giữa ++x và x++

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

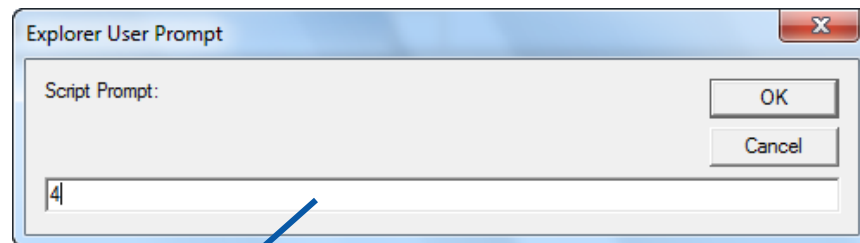


```
var x = 4;  
var y = x++;  
alert ("x = "+x+"   y =" +y);
```

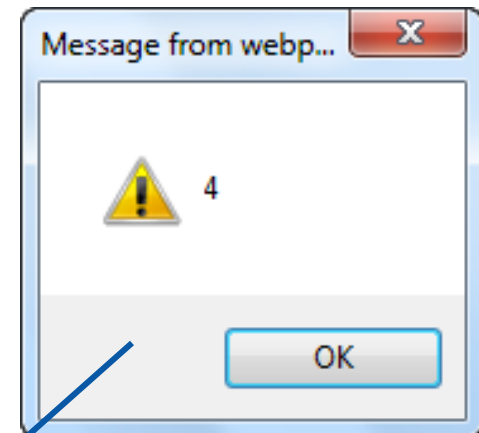


- ❑ Dùng để lấy dữ liệu do người dùng nhập vào
- ❑ Hàm trả về giá trị mà người dùng đã nhập
- ❑ Đoạn mã sau sẽ lấy dữ liệu do người dùng nhập vào và hiển thị kết quả trong hộp thoại:

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



Cửa sổ cho người dùng
nhập thông tin



Hiển thị giá trị người
dùng đã nhập

- ☐ Cấu trúc tuần tự
- ☐ Cấu trúc lựa chọn
- ☐ Cấu trúc lặp

- ☐ Lệnh lựa chọn đơn
- ☐ Lệnh lựa chọn kép
- ☐ Lệnh đa lựa chọn

□ Cú pháp

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

□ Các trường hợp

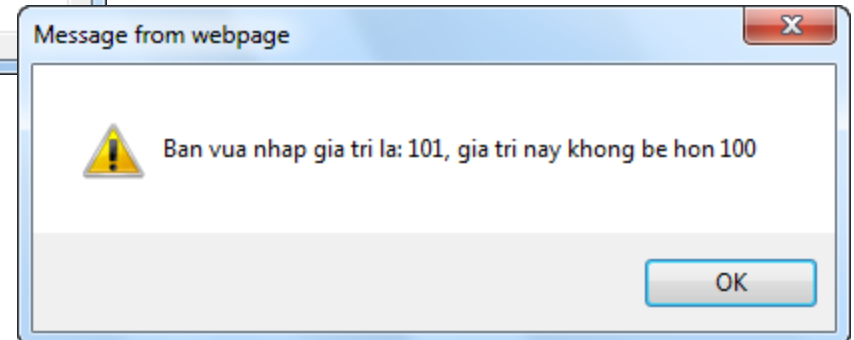
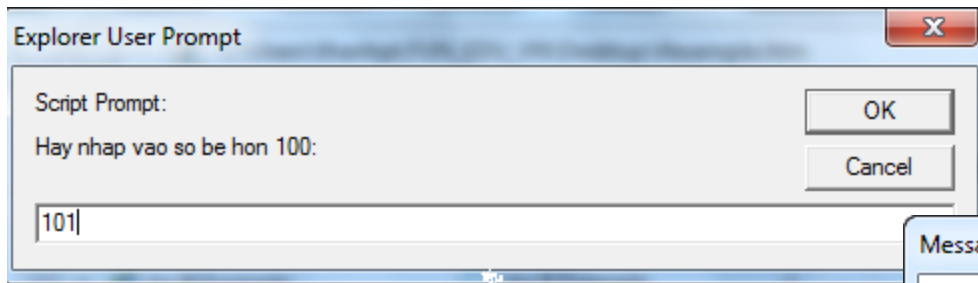
```
if (! điều_kiện) {  
    //Thực hiện  
}
```

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

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

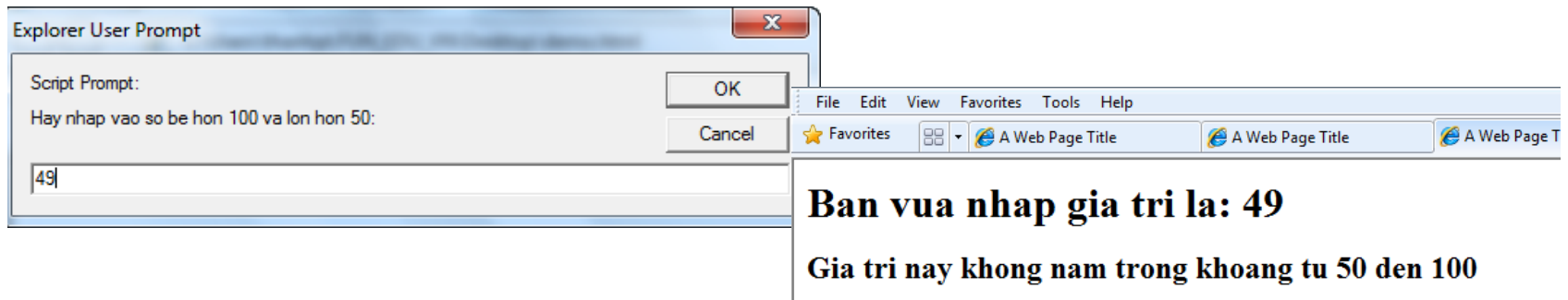
VÍ DỤ LỆNH LỰA CHỌN IF...THEN

```
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 kết hợp nhiều mệnh đề đơn thành một mệnh đề phức tạp hơn

```
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>");  
}
```

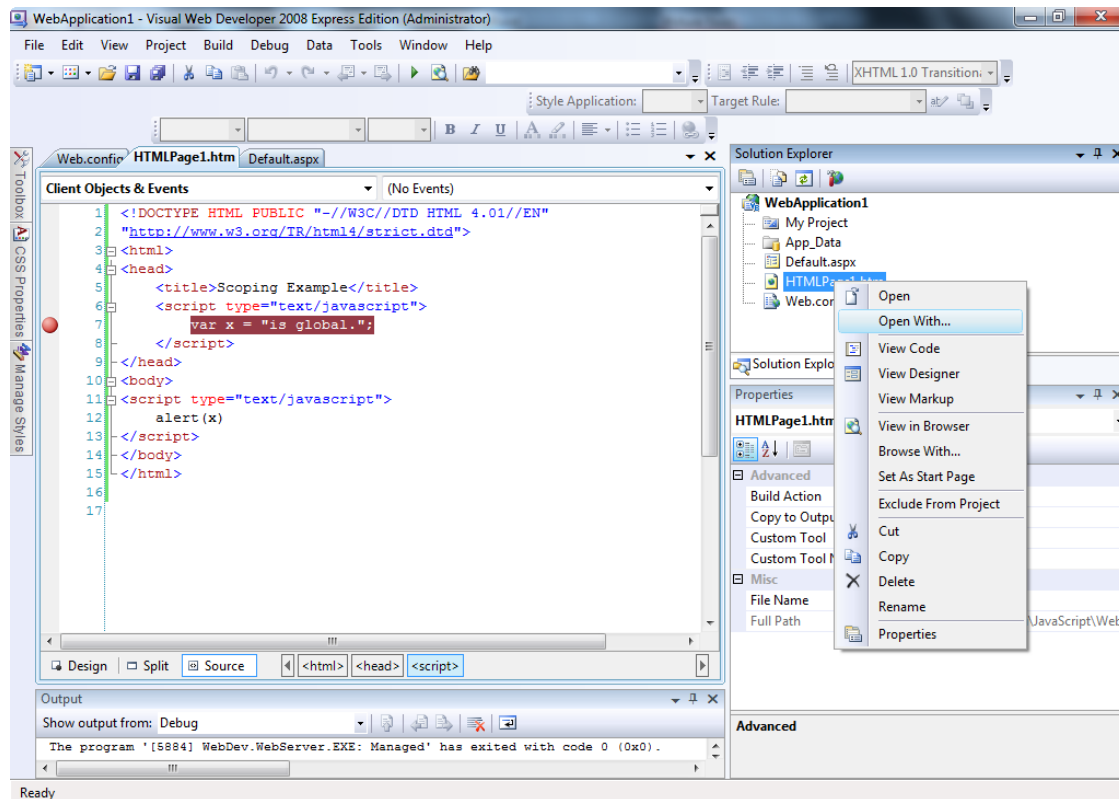


- ❑ Công cụ gỡ lỗi giúp người lập trình tìm và sửa lỗi (bao gồm lỗi logic và lỗi biên dịch)
- ❑ Vì mã JavaScript được thông dịch trên trình duyệt, nên mỗi trình duyệt sẽ xây dựng bộ gỡ lỗi riêng.
- ❑ IE:
 - ❖ Microsoft Script Debugger (Debug ngay trên trình duyệt IE)
 - ❖ Visual Studio
- ❑ Firefox
 - ❖ Firebug

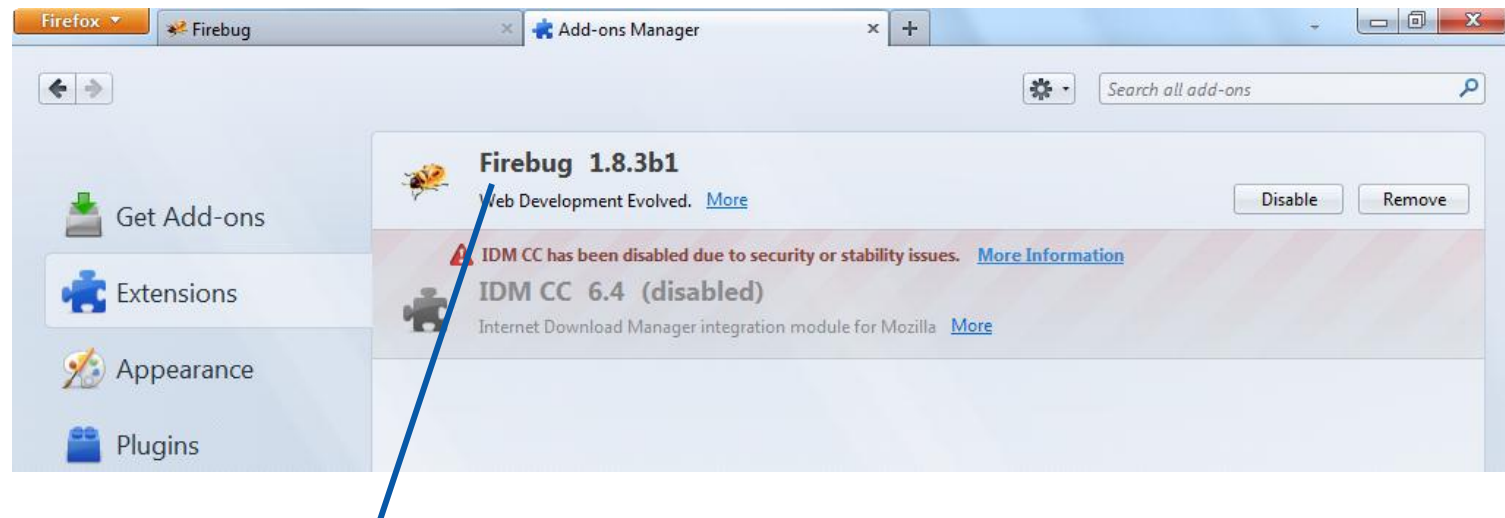


❑ Sử dụng Visual Studio để gỡ lỗi

- ❖ Mở file bằng IE: Click vào file → Chọn Open With → Chọn Internet Explorer → Visual Studio tự động chuyển sang chế độ gỡ lỗi



- ❑ 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



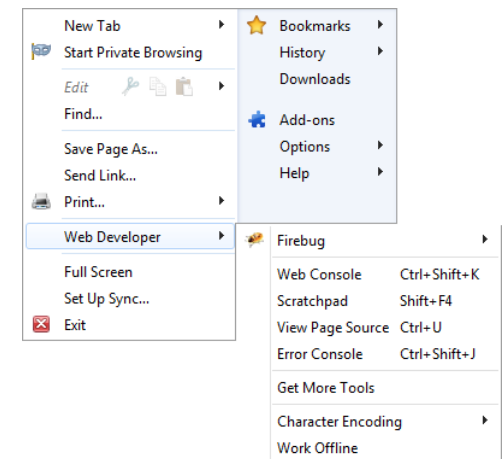
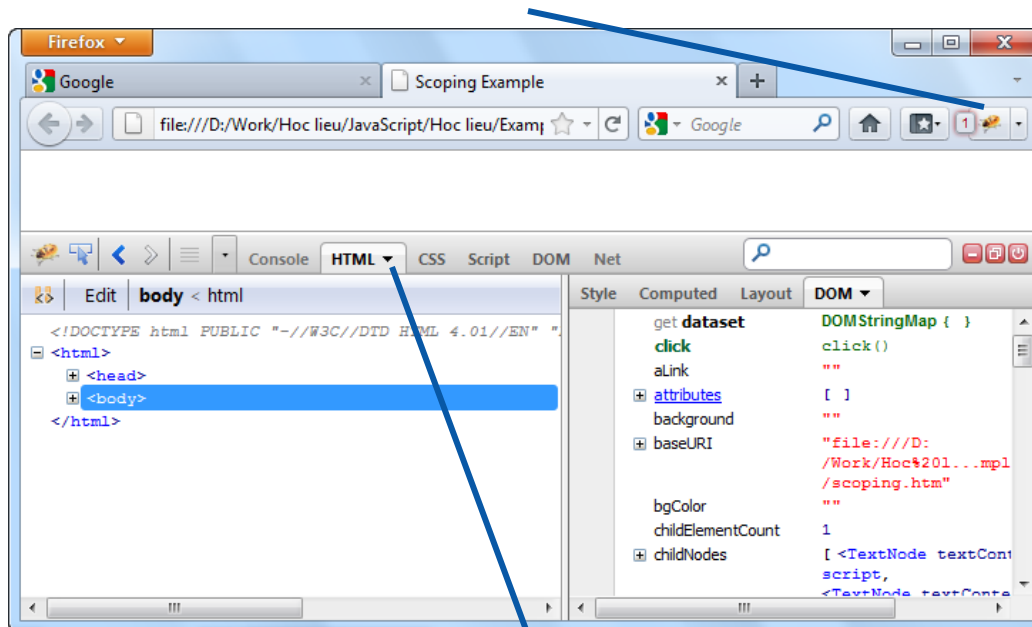
Firebug đã được cài đặt

❑ Bật chế độ Firebug

❖ Mở Firefox

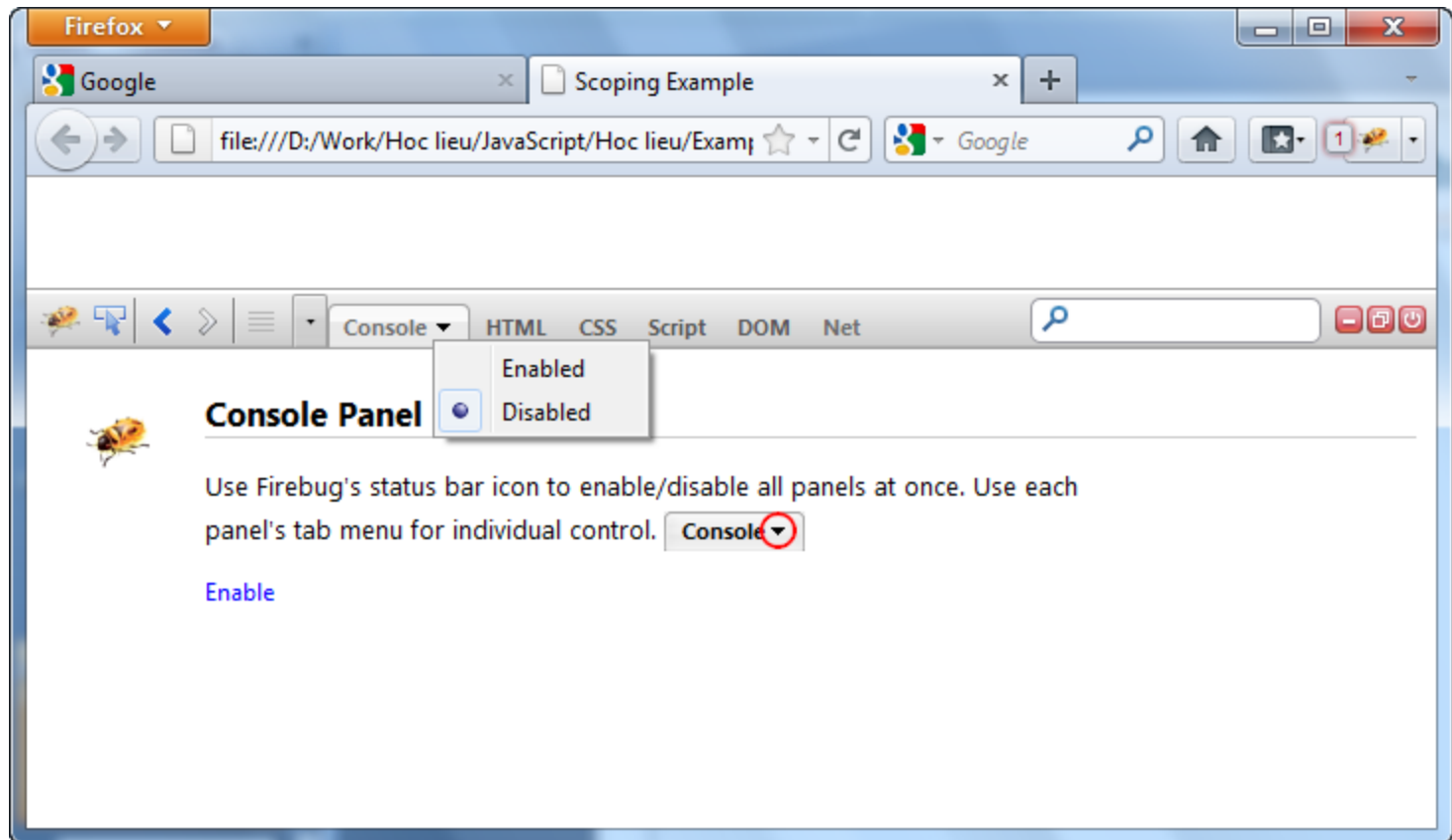
❖ Bật chế độ Firebug bằng cách nhấn vào biểu tượng Firebug hoặc vào từ Menu của Firefox như hình bên

Biểu tượng Firebug



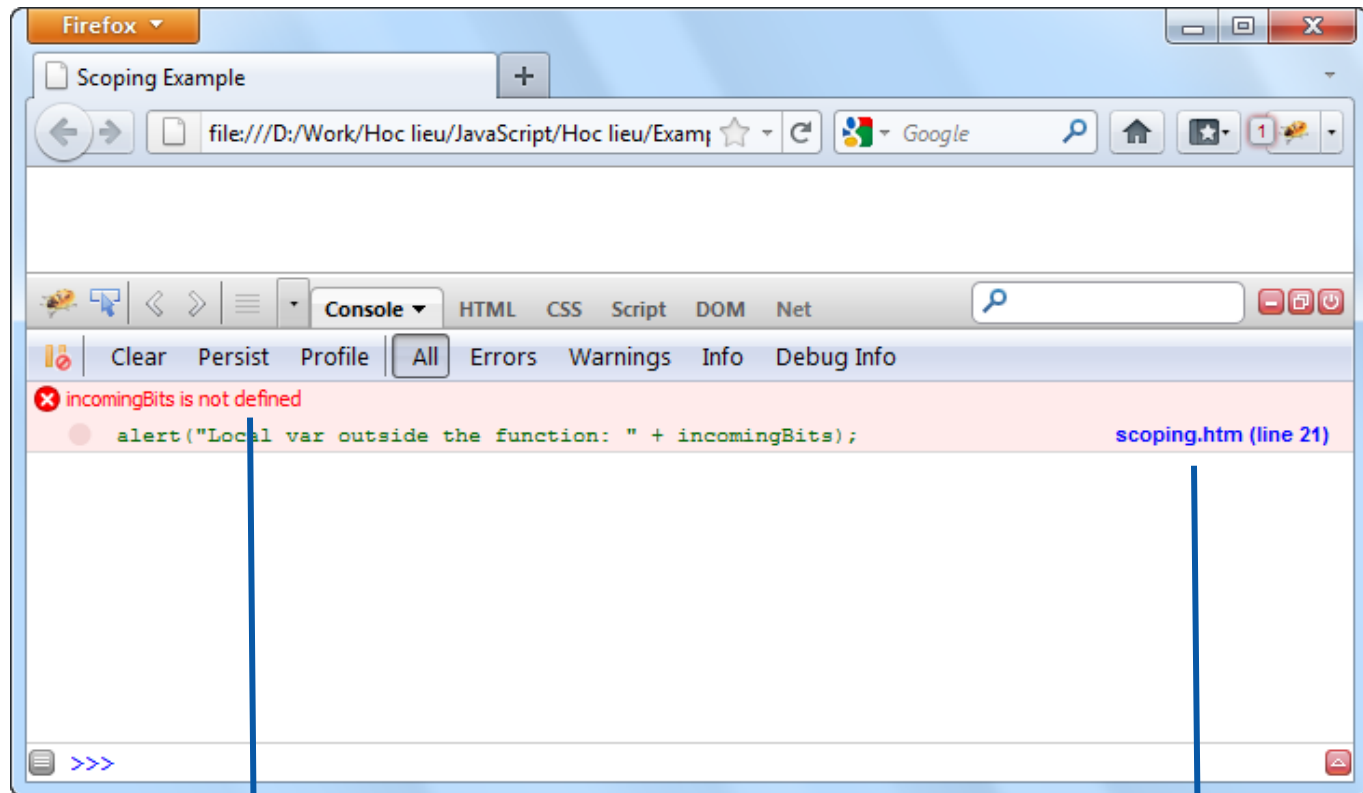
Cửa sổ Firebug xuất hiện

- ❑ **Bật chế độ Console để gỡ lỗi:** Nhấn vào mũi tên bên phải Console, chọn Enable



❑ Gỡ lỗi

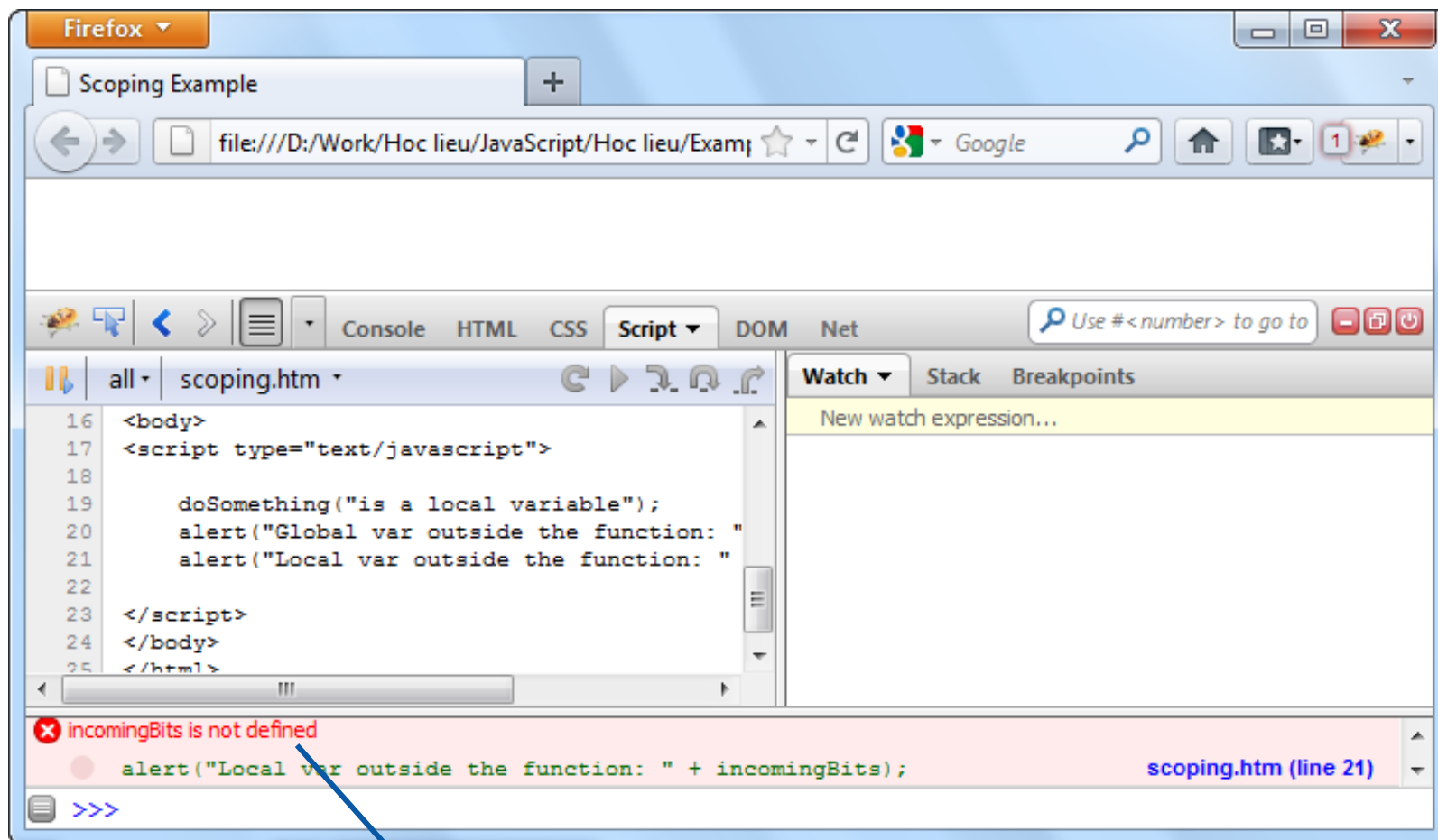
❖ Mở trang scoping.html



Cửa sổ thông báo lỗi

Lỗi ở dòng 21

❖ Nhấn vào lỗi để hiển thị đoạn mã bị lỗi



Cửa sổ thông báo lỗi

□ Bài tập về nhà:

- ❖ Tìm hiểu cách gỡ lỗi trực tiếp trên Chrome

- ❑ JavaScript hỗ trợ các kiểu biến: Kiểu số, kiểu String, kiểu Boolean và kiểu Object
- ❑ Kiểu mảng để 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 tường minh và ép kiểu ngầm định
- ❑ Biến được khai báo bằng từ khóa var, theo sau là tên biến
- ❑ JavaScript hỗ trợ các toán tử logic, toán tử số học, toán tử quan hệ và toán tử một ngôi

- ❑ JavaScript cung cấp 3 cấu trúc điều khiển: Cấu trúc tuần tự, cấu trúc lựa chọn và cấu trúc lặp
- ❑ Cấu trúc lặp gồm lệnh lựa chọn đơn (if)
- ❑ Vì mã JavaScript được thông dịch trên trình duyệt, nên mỗi trình duyệt sẽ xây dựng bộ gỡ lỗi riêng.
- ❑ Firebug gỡ lỗi cho Firefox, Microsoft Script Debugger gỡ lỗi cho IE. Visual Studio tích hợp bộ gỡ lỗi cho IE

FPT POLYTECHNIC



KẾT THÚC