

# **LẬP TRÌNH WEB**

## **CHƯƠNG 3:** **XỬ LÝ CÁC SỰ KIỆN TRÊN TRANG VỚI JAVASCRIPT**

---

- ❖ Giới thiệu ngôn ngữ JavaScript
- ❖ Biến, kiểu dữ liệu và toán tử
- ❖ Cấu trúc điều khiển
- ❖ Hàm
- ❖ Lớp và đối tượng
- ❖ Sự kiện

## I. Giới thiệu ngôn ngữ Javascript

- **JavaScript** là ngôn ngữ kịch bản dùng để tạo các client-side scripts và server-side scripts.
- **JavaScript** làm cho việc tạo các trang Web động và tương tác dễ dàng hơn
- **JavaScript** làm một ngôn ngữ kịch bản được hãng Sun Microsystems và Netscape phát triển.
- **JavaScript** được phát triển từ *Livescript*. Của Netscape
- Các ứng dụng client cho phép được chạy trên hầu hết các trình duyệt FireFox, googlechrom, internet explorer, opera,....

- ❖ Javascript là 1 trong 3 ngôn ngữ mà 1 web developer phải học:
  - HTML quy định nội dung trang web
  - CSS quy định bố cục trang web
  - Javascript quy định cách ứng xử của trang web
    - Thay đổi nội dung HTML
    - Thay đổi thuộc tính HTML
    - Thay đổi CSS của HTML
    - Validate dữ liệu

# I. Giới thiệu ngôn ngữ javascript

## ❖ **Nền tảng cú pháp ngôn ngữ Javascript**

- **Phân biệt chữ hoa chữ thường**
- **Sử dụng dấu chấm phẩy để phân biệt dòng lệnh**
- **Các lệnh có thể được kết hợp với nhau trong cặp dấu {}**
- **Cho phép ghi chú thích**

# I. Giới thiệu ngôn ngữ Javascript

❖ JavaScript có thể chèn vào một tài liệu HTML theo những cách sau:

- Chèn trực tiếp vào trong thuộc tính của 1 thẻ HTML

```
<button type="button"  
onclick="document.getElementById('demo').innerHTML = 'Hello  
JavaScript!'">  
Click Me!</button>
```

❖ Sử dụng thẻ SCRIPT:

```
<script language="JavaScript">  

```

- ❖ Javascript là ngôn ngữ xử lý sự kiện trên trang web, không có chức năng hiển thị nội dung trên trang
- ❖ Các phương pháp “hiển thị” dữ liệu bằng Javascript:
  - Hộp thoại cảnh báo **window.alert()**.
  - HTML output **document.write()**.
  - Hiển thị trong một HTML element **innerHTML**.
  - Lưu vết trong cửa sổ console của trình duyệt **console.log()**.

## ❖ Single line comment (comment trên 1 dòng)

*// Change heading:*

```
document.getElementById("myH").innerHTML = "My First Page";
```

*// Change paragraph:*

```
document.getElementById("myP").innerHTML = "My first  
paragraph.";
```

## ❖ Multi-line comment (comment khối)

*/\**

*The code below will change  
the heading with id = "myH"  
and the paragraph with id = "myP"  
in my web page:*

*\*/*

```
document.getElementById("myH").innerHTML = "My First Page";  
document.getElementById("myP").innerHTML = "My first  
paragraph.";
```



- ❖ **Giới thiệu ngôn ngữ JavaScript**
- ❖ **Biến, kiểu dữ liệu và toán tử**
- ❖ **Cấu trúc điều khiển**
- ❖ **Hàm**
- ❖ **Lớp và đối tượng**
- ❖ **Sự kiện**

- ❖ Các biến tuân theo quy tắc đặt tên.
- ❖ Một biến được khai báo sử dụng từ khoá '**var**'  
ví dụ:        **var** A = 10; // không cần khai báo kiểu, javascript sẽ tự hiểu
- ❖ The general rules for constructing names for variables (unique identifiers) are:
  - Names can contain letters, digits, underscores, and dollar signs.
  - Names must begin with a letter
  - Names can also begin with \$ and \_
  - Names are case sensitive (y and Y are different variables)
  - Reserved words (like JavaScript keywords) cannot be used as names

- ❖ Trong JavaScript không cần chỉ ra kiểu dữ liệu cho biến và kiểu dữ liệu của biến sẽ được tự động chuyển thành kiểu phù hợp khi cần
- ❖ Khai báo biến: *var carName;* // carName = undefined
- ❖ Gán giá trị: *carName = "BMW";*

- ❖ Trong JavaScript, hai biến khác kiểu có thể kết hợp với nhau.

ví dụ: `A = " This apple costs Rs." + 5`

sẽ có kết quả là một chuỗi với giá trị là "This apple costs Rs. 5".

- ❖ Kiểu dữ liệu của 1 biến có thể thay đổi:

*`var x;           // Now x is undefined`*

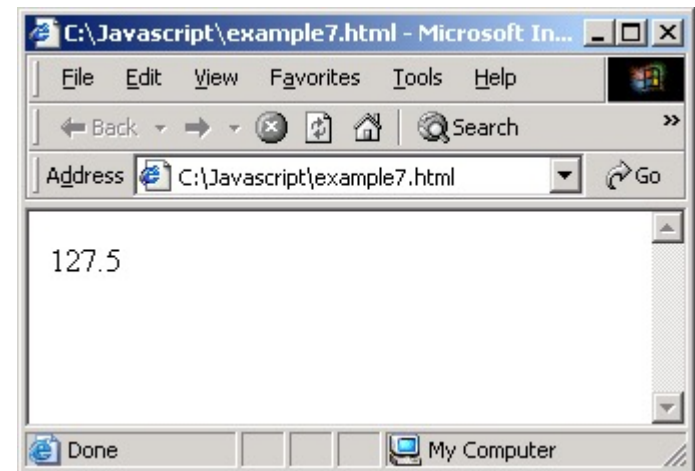
*`var x = 5;       // Now x is a Number`*

*`var x = "John";   // Now x is a String`*

## Ví dụ:

```
<HTML>
<HEAD>
<SCRIPT LANGUAGE = "Javascript">
    var A = "12" + 7.5;
    document.write(A);
</SCRIPT>
</HEAD>
</HTML>
```

## Kết quả:



# Các kiểu nguyên dạng (literal)

- ❖ Integer – là các hệ thống số thập phân, thập lục phân và nhị phân.
- ❖ Floating- point(số thực) – **Các số thập phân** có phần thập phân sử dụng “e” hoặc “E” và theo sau là các số nguyên.
- ❖ String – là một chuỗi rỗng hay chuỗi ký tự được đặt trong cặp ngoặc đơn hoặc ngoặc kép
- ❖ Boolean–Kiểu này có hai giá trị: True or False.
- ❖ null - Kiểu null chỉ có một giá trị: null. Null hàm ý không có dữ liệu.

## Sự khác biệt giữa *null* và *undefined*

- ❖ *undefined* means a variable has been declared but has not yet been assigned a value, such as:

```
var TestVar;
```

```
alert(TestVar); //shows undefined
```

```
alert(typeof TestVar); //shows undefined
```

- ❖ *null* is an assignment value. It can be assigned to a variable as a representation of no value:

```
var TestVar = null;
```

```
alert(TestVar); //shows null
```

```
alert(typeof TestVar); //shows object
```

## 2.3 Các toán tử

- ❖ Toán tử số học
- ❖ Toán tử gán
- ❖ Toán tử với xâu ký tự
- ❖ Toán tử so sánh



## Toán tử số học

- ❖ Các toán tử số học lấy hai toán tử (hoặc nguyên dạng hoặc các biến) và trả về một giá trị số.
- ❖ Toán tử số học gồm:
  - Cộng (+)
  - Trừ (-)
  - Nhân (\*)
  - Chia (/)
  - Toán tử tăng (++)
  - Toán tử giảm (--)
  - Toán tử chia đồng dư (%)

Operator	Example	Same As
=	$x = y$	$x = y$
+=	$x += y$	$x = x + y$
-=	$x -= y$	$x = x - y$
*=	$x *= y$	$x = x * y$
/=	$x /= y$	$x = x / y$
%=	$x \% = y$	$x = x \% y$

- ❖ Toán tử so sánh so sánh các toán hạng và trả về giá trị logic dựa trên sự so sánh có đúng hay không.
- ❖ Toán tử so sánh bao gồm:
  - So sánh bằng (==)
  - So sánh bằng cả về giá trị và kiểu (===)
  - So sánh khác (!=)
  - So sánh lớn hơn (>)
  - So sánh lớn hơn hoặc bằng (>=)
  - So sánh nhỏ hơn (<)
  - So sánh nhỏ hơn hoặc bằng (<=)

# Toán tử chuỗi

- ❖ Toán tử chuỗi lấy các toán tử chuỗi như các toán hạng và tạo một chuỗi mới, kết quả là một chuỗi kết hợp các chuỗi con.

Ví dụ:

$x = \text{'yellow'}$ ;

$y = \text{'green'}$ ;

$z = x + y + \text{'white'}$ ; which means  $z$  is  
"yellowgreenwhite"

$w = y + 9$ , which means  $w$  is "green9"

## Mức ưu tiên của toán tử

- ❖ Khi có nhiều toán tử được tính toán trong một biểu thức, mức ưu tiên của toán tử xác định thứ tự thực hiện của toán tử trong biểu thức đó.
- ❖ Bảng sau liệt kê mức ưu tiên của các toán tử từ thấp đến cao:

Operator type	Individual operators
assignment	= += -= *= /= %= <<= >>= >>>= &= ^=  =
conditional	?:
logical-or	
logical-and	&&
bitwise-or	
bitwise-xor	^
bitwise-and	&
equality	== !=
relational	< <= > >=
bitwise shift	<< >> >>>
addition/subtraction	+ -
multiply/divide	* / %
negation/increment	! ~ - ++ -- <u>typeof void</u>

- ❖ Mảng được dùng để lưu trữ một dãy các biến với cùng một tên.
- ❖ Một số (chỉ số) dùng để phân biệt các giá trị khác nhau.
- ❖ Các mảng bắt đầu với chỉ số 0 trong JavaScript
- ❖ Cú pháp *var cars = ["Saab", "Volvo", "BMW"];*
- ❖ Duyệt các phần tử trong mảng:

*Var index;*

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];  
for (index = 0; index < fruits.length; index++) {  
    text += fruits[index];  
}
```

- ❖ Giới thiệu ngôn ngữ JavaScript
- ❖ Biến, kiểu dữ liệu và toán tử
- ❖ Cấu trúc điều khiển
- ❖ Hàm
- ❖ Lớp và đối tượng
- ❖ Sự kiện

## 3. Cấu trúc điều khiển

### ❖ Câu lệnh điều kiện:

- Câu lệnh điều kiện được dùng để kiểm tra điều kiện. Kết quả xác định câu lệnh hoặc khối lệnh được thực thi.
- Các câu lệnh điều kiện bao gồm:
  - If..... Else
  - Switch



### 3. Cấu trúc điều khiển

- ❖ Cấu trúc điều khiển lặp trong chương trình là các lệnh lặp.
- ❖ Các kiểu lệnh lặp bao gồm:
  - For
  - Do .... While
  - While
  - Break & continue

- ❖ Giới thiệu ngôn ngữ JavaScript
- ❖ Biến, kiểu dữ liệu và toán tử
- ❖ Cấu trúc điều khiển
- ❖ **Hàm**
- ❖ Lớp và đối tượng
- ❖ Sự kiện

## 4. Hàm

- ❖ Một hàm trong Javascript là 1 đoạn code (block of code) dùng để thực hiện 1 công việc nào đó
- ❖ Một hàm Javascript được thực hiện khi có sự gọi đến hàm đó
  - Khi có 1 sự kiện được kích hoạt
  - Có 1 câu lệnh Javascript gọi đến
  - Tự gọi hàm (self invoked)
- ❖ Khi nào dùng hàm số:
  - Tái sử dụng các đoạn code
  - Cùng 1 đoạn code với các tham số khác nhau cho ra các kết quả khác nhau
- ❖ Cú pháp:

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

## ❖ Cú pháp:

```
function name(parameter1, parameter2, parameter3) {  
    code to be executed  
}
```

## ❖ Giá trị trả về: khi gặp từ khóa *return*, hàm sẽ dừng thực hiện


```
var x = myFunction(4, 3);    // Function is called, return  
value will end up in x
```

```
function myFunction(a, b) {  
    return a * b;            // Function returns the product of a  
and b  
}
```

- ❖ Giới thiệu ngôn ngữ JavaScript
- ❖ Biến, kiểu dữ liệu và toán tử
- ❖ Cấu trúc điều khiển
- ❖ Hàm
- ❖ Lớp và đối tượng
- ❖ Sự kiện

## ❖ Đối tượng trong thế giới thực:

- Thuộc tính
- Phương thức

Object	Properties	Methods
	<p>car.name = Fiat</p> <p>car.model = 500</p> <p>car.weight = 850kg</p> <p>car.color = white</p>	<p>car.start()</p> <p>car.drive()</p> <p>car.brake()</p> <p>car.stop()</p>

## Đối tượng trong Javascript

- ❖ Đối tượng cũng là 1 biến, nhưng có thể chứa nhiều giá trị
- ❖ Các giá trị chính là các thuộc tính của đối tượng, được định nghĩa bởi các cặp *name: value*

```
var car = {type:"Fiat", model:500,  
color:"white"};
```

- ❖ Phương thức là các hành động có thể được hành động thực hiện, được định nghĩa bởi các hàm

```
fullName : function() {  
    return this.firstName + " " + this.lastName;  
}
```

- ❖ Truy cập đến thuộc tính của đối tượng
  - *objectName.propertyName*
  - *objectName[propertyName]*
- ❖ Gọi phương thức của đối tượng
  - *objectName.methodName()*



- ❖ Sự khác biệt giữa mảng và đối tượng
  - In JavaScript, **arrays** use **numbered indexes**.
  - In JavaScript, **objects** use **named indexes**

- ❖ **Giới thiệu ngôn ngữ JavaScript**
- ❖ **Biến, kiểu dữ liệu và toán tử**
- ❖ **Cấu trúc điều khiển**
- ❖ **Hàm**
- ❖ **Lớp và đối tượng**
- ❖ **Sự kiện**

- ❖ Là những tác động gây ra đối với các HTML element
  - Do trình duyệt tác động
  - Do người dùng tác động
- ❖ Javascript thực hiện những đoạn code khi phát hiện có 1 sự kiện nào đó vừa kích hoạt
  - *<some-HTML-element some-event='some JavaScript'>*
  - *<some-HTML-element some-event="some JavaScript">*

## ❖ Một số sự kiện thường gặp

Event	Description
onchange	An HTML element has been changed
onclick	The user clicks an HTML element
onmouseover	The user moves the mouse over an HTML element
onmouseout	The user moves the mouse away from an HTML element
onkeydown	The user pushes a keyboard key
onload	The browser has finished loading the page

The list is much longer: [W3Schools JavaScript Reference HTML DOM Events](http://www.w3schools.com/jsref/dom_obj_event.asp).

## ❖ Đây đủ:

[http://www.w3schools.com/jsref/dom\\_obj\\_event.asp](http://www.w3schools.com/jsref/dom_obj_event.asp)

- ❖ Sử dụng Javascript để nhập 1 mảng gồm 5 phần tử bất kỳ và hiện thông báo ra màn hình 5 phần tử đó (hiện thông báo qua lệnh `alert()` hoặc thuộc tính `innerHTML`)
- ❖ Thiết kế 2 ô nhập số và 1 button để thực hiện tính cộng 2 số đã nhập.
  - Dùng cú pháp `document.getElementById(id_cua_input).value` để lấy ra giá trị vừa nhập và
  - Sử dụng `Alert` để in ra kết quả