



# Ngôn ngữ Javascript

**ThS. Nguyễn Duy Hải**  
**PGĐ TT CNTT-ĐHSP Hà Nội**

# JavaScript

---

- ✓ JavaScript là ngôn ngữ kịch bản dùng để tạo các client-side scripts.
- ✓ JavaScript làm cho việc tạo các trang Web động và tương tác dễ dàng hơn
- ✓ JavaScript là 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 chạy trên một trình duyệt như Netscape Navigator hoặc Internet Explorer.



# Khả năng của Javascript

---

- ✓ JavaScript có thể tăng cường tính động và tính tương tác của các trang web.
  - Cung cấp sự tương tác người dùng
  - Thay đổi nội dung động
  - Xác nhận tính hợp lệ của dữ liệu



# Quy tắc

---

- ✓ Sử dụng quy tắc chữ hoa
- ✓ Using Pairs
- ✓ Using Spaces
- ✓ Using Comments



# Công cụ và môi trường thực thi

---

- ✓ Các công cụ sinh mã JavaScript
  - Thuận lợi khi soạn thảo
  - Mã lệnh sẵn có
- ✓ Môi trường thực thi
  - Các Scripting ở Client
  - Java Script trên Web Server



# Chèn Javascript vào HTML

## ✓ Sử dụng thẻ SCRIPT:

```
<script language="JavaScript">  
<!--  
    JavaScript statements;  
//-->  
</script>
```

## ✓ Sử dụng một file JavaScript ở ngoài

```
<script language="JavaScript" src="filename.js">  
</script>
```

## ✓ Sử dụng các biểu thức JavaScript trong các giá trị thuộc tính của thẻ

## ✓ Sử dụng trong các trình điều khiển sự kiện



```
<HTML>  
  <HEAD>  
    <SCRIPT LANGUAGE = "Javascript">  
      confirm ("Are you Sure?");  
      alert ("OK");  
      document.write (" Thank You !");  
    </SCRIPT>  
  </HEAD>  
</HTML>
```



- ✓ Biến là một vật chứa tham chiếu đến một vị trí ở bộ nhớ máy tính
- ✓ Nó được sử dụng để giữ giá trị và có thể thay đổi trong khi kịch bản thực thi
- ✓ Các biến tuân theo quy tắc đặt tên.
- ✓ Một biến được khai báo sử dụng từ khoá `'var'`.  
ví dụ: `var A = 10;`
- ✓ Các biến có một phạm vi được xác định trong khi chúng khai báo trong script.
  - Biến toàn cục
  - Biến cục bộ
- ✓ Nguyên dạng là các giá trị không đổi được dùng trong script.





# Các kiểu dữ liệu

- ✓ JavaScript có một tập các kiểu dữ liệu.
  - Số (number)
  - Giá trị logic (boolean)
  - Chuỗi (String)
  - Giá trị rỗng Null
- ✓ 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"`.



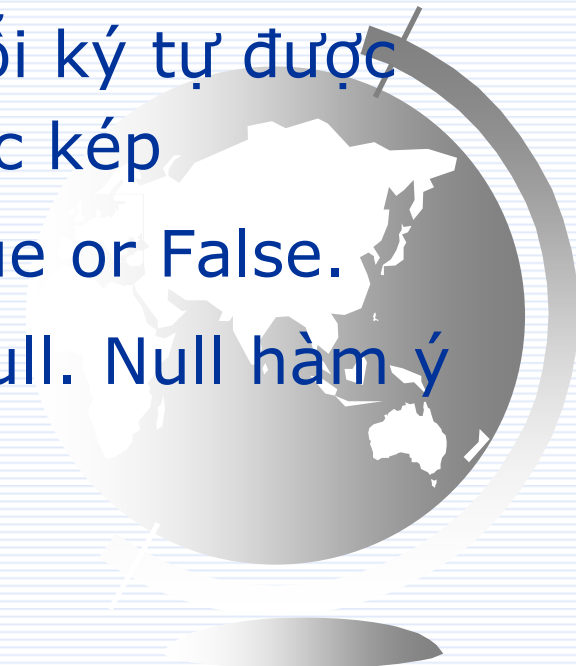
```
<HTML>  
  
<HEAD>  
  
<SCRIPT LANGUAGE = "Javascript">  
    var A = "12" + 7.5;  
    document.write(A);  
  
</SCRIPT>  
  
</HEAD>  
  
</HTML>
```



# Các kiểu nguyên thủy

---

- ✓ 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" or ""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.



# Toán tử

- ✓ Các toán tử xử lý một hoặc nhiều biến hoặc các giá trị (các toán hạng) và trả lại giá trị kết quả.
- ✓ JavaScript sử dụng cả hai toán tử một ngôi và hai ngôi.
- ✓ Các toán tử được phân loại phụ thuộc quan hệ chúng thực hiện như:
  - Toán tử số học
  - Toán tử so sánh
  - Toán tử logic
  - Toán tử chuỗi
  - Toán tử lượng giá
- ✓ Mức ưu tiên của toán tử



# Ví dụ

```
<HTML>
  <HEAD>
    <SCRIPT LANGUAGE="JavaScript">
      var x = 10;
      var y = 5;
      alert ("The value of x is "
            + x + "The value of y is " + y);
      alert("x AND y = " + (x && y));
      alert("x OR y = " + (x || y));
      alert("NOT x = " + (!x));
    </SCRIPT>
  </HEAD>
</HTML>
```



# Mảng

- ✓ 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.
- ✓ Tạo mảng: Cú pháp

```
arrayObjectName = new Array([element0, element1,  
..., elementN])
```

- ✓ Cộng các phần tử: Chúng ta có thể cộng các phần tử mảng khi chúng ta tạo nó. Ví dụ. `emp[0] = "Ryan Dias"`
- ✓ Các phần tử của một mảng có thể truy cập bằng tên Name hoặc chỉ số Index của phần tử.



# Mảng (tt)

- ✓ Các phương thức của đối tượng mảng có thể dùng thao tác trên mảng.
- ✓ Các phương thức của đối tượng mảng bao gồm:
  - join
  - pop
  - push
  - reverse
  - shift
  - sort
- ✓ JavaScript hỗ trợ mảng nhiều chiều.



# Lệnh rẽ nhánh

---

- ✓ 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`





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



# Hàm

- ✓ JavaScript có sẵn các hàm định nghĩa trước dùng trong script.
- ✓ Một vài hàm định nghĩa trước trong JavaScript bao gồm:
  - Hàm eval
  - Hàm isNaN

- ✓ Hàm do người dùng tự tạo

```
function funcName(argument1, argument2, ...) {  
    statements;  
}
```

- ✓ Gọi hàm
- ✓ Câu lệnh Return





# **Các đối tượng cơ bản trong Javascript**

# Đối tượng

- ✓ Thuộc tính (biến) dùng để định nghĩa đối tượng và các phương thức (hàm) tác động tới dữ liệu đều nằm trong đối tượng.
- ✓ Ví dụ: một chiếc xe hơi là một đối tượng. Các thuộc tính của nó là cấu tạo, kiểu dáng và màu sắc. Hầu hết các chiếc xe hơi đều có một vài phương thức chung như `go()`, `brake()`, `reverse()`.



# Thuộc tính và phương thức

---

- ✓ Để truy cập thuộc tính của đối tượng, chúng ta phải chỉ ra tên đối tượng và thuộc tính của nó:

```
objectName.propertyName
```

- ✓ Để truy cập phương thức của đối tượng, chúng ta phải chỉ ra tên đối tượng và thuộc tính của nó:

```
objectName.method()
```



# Sử dụng đối tượng

---

- ✓ Khi tạo trang web, chúng ta cần sử dụng:
  - Các đối tượng trình duyệt
  - Các đối tượng có sẵn (thay đổi phụ thuộc vào ngôn ngữ kịch bản được sử dụng)
  - HTML elements
- ✓ Chúng ta cũng có thể tạo ra các đối tượng để sử dụng theo yêu cầu của mình.



## Từ khóa this

---

- ✓ Giá trị của nó chỉ ra đối tượng hiện hành và có thể có các thuộc tính chuẩn chẳng hạn như tên, độ dài, và giá trị được áp dụng phù hợp.



# Lệnh for...in

- ✓ Câu lệnh `for...in` được dùng để lặp mỗi thuộc tính của đối tượng hoặc mỗi phần tử của một mảng.
- ✓ Cú pháp:

```
for (variable in object) {  
    statements;  
}
```





- ✓ Câu lệnh `with` được dùng để thực thi tập hợp các lệnh mà các lệnh này dùng các phương thức của cùng một loại đối tượng.
- ✓ thuộc tính được gán cho đối tượng đã được xác định trong câu lệnh `with`.
- ✓ Cú pháp:

```
with (object) {  
    statements;  
}
```



# Toán tử new

- ✓ Toán tử **new** được dùng để tạo ra một thực thể mới của một loại đối tượng
- ✓ Đối tượng có thể có sẵn hoặc do người dùng định nghĩa
  - `objectName = new objectType (param1 [,param2] ...[,paramN])`
  - Trong đó:
- ✓ `objectName` là tên của thực thể đối tượng mới.
- ✓ `ObjectType` là một hàm quyết định loại của đối tượng. Ví dụ `Array`.
- ✓ `Param[1, 2, . . .]` là các giá trị thuộc tính của đối tượng.



# Hàm eval

- ✓ Hàm `eval` được dùng để đánh giá một chuỗi mã lệnh mà không cần tham chiếu đến bất cứ đối tượng cụ thể nào.
- ✓ Chuỗi có thể là một biểu thức JavaScript, một câu lệnh hoặc một nhóm câu lệnh
- ✓ Biểu thức có thể bao gồm nhiều biến và nhiều thuộc tính của một đối tượng.

```
var x = 5;
```

```
var z = 10;
```

```
document.write(eval("x + z + 5"));
```



# Đối tượng string

- ✓ Đối tượng **String** được dùng để thao tác và làm việc với chuỗi văn bản.
- ✓ Chúng ta có thể tách chuỗi ra thành các chuỗi con và biến đổi chuỗi đó thành các chuỗi hoa hoặc thường trong một chương trình.

- ✓ Cú pháp tổng quát:

`stringName.propertyName`

hay

`stringName.methodName`



# Tạo đối tượng string

---

- ✓ Có 3 phương thức khác nhau để tạo ra chuỗi.
  - Dùng lệnh var và gán cho nó một giá trị.
  - Dùng một toán tử (=) có gán với một tên biến.
  - Dùng hàm khởi tạo String (string).



# Đối tượng Math

- ✓ Đối tượng Math có các thuộc tính và phương thức biểu thị các phép tính toán học nâng cao.

```
function doCalc(x) {  
    var a;  
    a = Math.PI * x * x;  
    alert ("The area of a circle with a  
    radius of " + x + " is " + a);  
}
```



# Đối tượng Date

---

- ✓ Date là một đối tượng có sẵn chứa thông tin về ngày và giờ.
- ✓ Đối tượng Date không có thuộc tính nào.
- ✓ Nó có nhiều phương thức dùng để thiết lập, lấy và xử lý các thông tin về thời gian.



# Đối tượng Date (tt)

---

- ✓ Đối tượng Date lưu trữ thời gian theo số mili giây tính từ 1/1/1970 00:00:00
- ✓ `DateObject = new Date(parameters)`





## ✓ Các sự kiện JavaScript hỗ trợ

- onClick
- onChange
- onFocus
- onBlur
- onMouseOver

- onMouseOut
- onLoad
- onSubmit
- onMouseDown
- onMouseUp

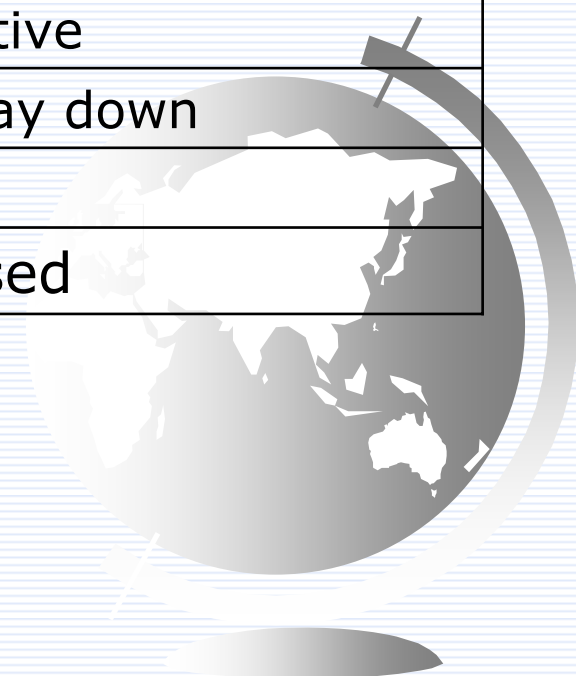
## ✓ Đặt bộ lắng nghe sự kiện:

```
<thẻ sựkiện="lệnh;">
```



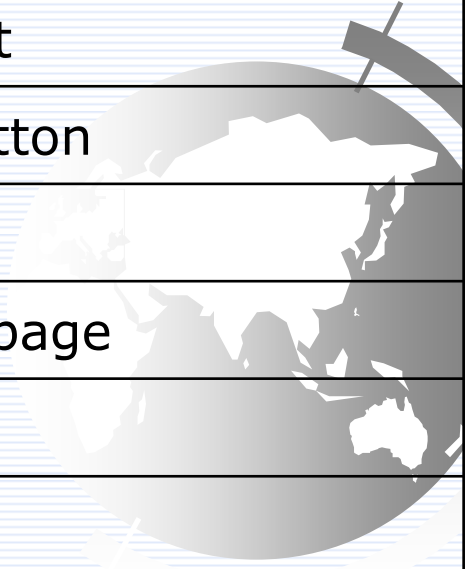
# Danh sách sự kiện đầy đủ

Event	Occurs when...
<b>onabort</b>	a user aborts page loading
<b>onblur</b>	a user leaves an object
<b>onchange</b>	a user changes the value of an object
<b>onclick</b>	a user clicks on an object
<b>ondblclick</b>	a user double-clicks on an object
<b>onfocus</b>	a user makes an object active
<b>onkeydown</b>	a keyboard key is on its way down
<b>Onkeypress</b>	a keyboard key is pressed
<b>onkeyup</b>	a keyboard key is released



# Danh sách sự kiện đầy đủ (tt)

<b>onload</b>	a page is finished loading. <b>Note:</b> In Netscape this event occurs during the loading of a page!
<b>onmousedown</b>	a user presses a mouse-button
<b>onmousemove</b>	a cursor moves on an object
<b>onmouseover</b>	a cursor moves over an object
<b>onmouseout</b>	a cursor moves off an object
<b>onmouseup</b>	a user releases a mouse-button
<b>onreset</b>	a user resets a form
<b>onselect</b>	a user selects content on a page
<b>onsubmit</b>	a user submits a form
<b>onunload</b>	a user closes a page





# **Các đối tượng của trình duyệt**

# DOM (Document Object Models)

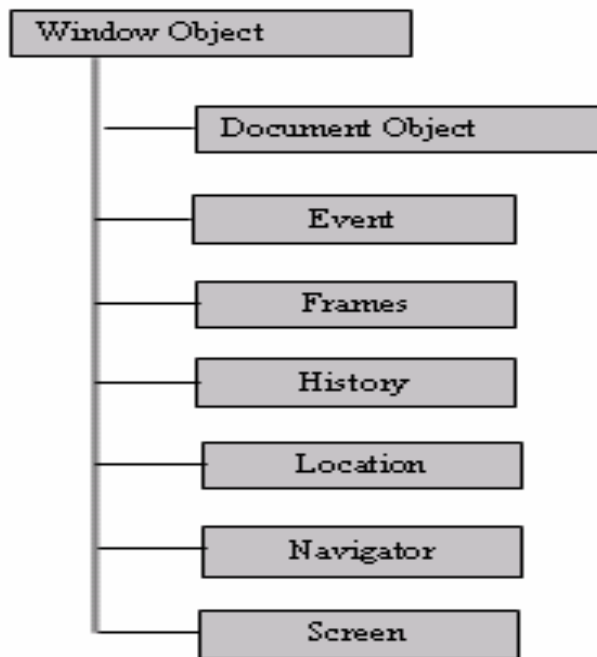
---

- ✓ Một tính năng quan trọng của JavaScript là ngôn ngữ dựa trên đối tượng.
- ✓ Giúp người dùng phát triển chương trình theo môđun và có thể sử dụng lại.
- ✓ Đối tượng được định nghĩa là một thực thể đơn nhất bao gồm các thuộc tính và phương thức.
- ✓ Thuộc tính là giá trị của một đối tượng.

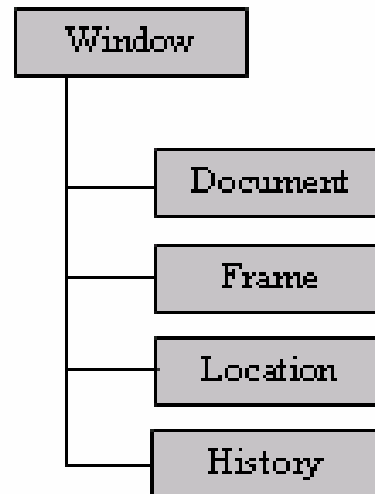


# Các đối tượng trên trình duyệt

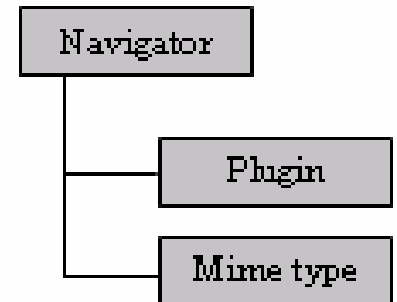
- ✓ Trình duyệt là một ứng dụng được sử dụng để hiển thị nội dung của tài liệu HTML.
- ✓ Các trình duyệt cũng đưa ra một số đối tượng có thể được truy cập và sử dụng trong script .
- ✓ Chi tiết các thuộc tính: tra tài liệu hoặc [w3schools](http://w3schools.com)



**IE Browser Objects**



**Netscape Browser Objects**



# Một vài ứng dụng

---

- ✓ Thay đổi thanh trạng thái, tiêu đề
- ✓ Tự động refresh
- ✓ Kiểm tra tính hợp lệ của dữ liệu form
- ✓ Hộp thoại tự tạo



# Một vài ứng dụng (tiếp)

---

- ✓ Calendars
- ✓ Date & Time
- ✓ Document Effects
- ✓ Dynamic Content (Iframe & Ajax)
- ✓ Form Effects
- ✓ Games
- ✓ Image Effects
- ✓ Galleries, Mouseover, Slideshows
- ✓ Links & Tooltips
- ✓ Menus & Navigation
- ✓ CSS Based, Multi-levels
- ✓ Mouse and Cursor
- ✓ Scrollers
- ✓ Text Animations
- ✓ User/System Preference
- ✓ Window and Frames
- ✓ XML and RSS
- ✓ Other





# Tham khảo

---

- ✓ <http://www.javascriptkit.com>
- ✓ <http://www.dynamicdrive.com>
- ✓ <http://www.javascriptbank.com>
- ✓ <http://www.dhtmlcentral.com>





**Một vài ví dụ**



# **Hiệu ứng chữ chạy trên trình thanh trạng thái của trình duyệt**

# Lý thuyết

- ✓ `window` là đối tượng quản lý cửa sổ trình duyệt.
- ✓ Đối tượng `window` có thuộc tính `status` để xác định thông báo tạm thời xuất hiện trên thanh trạng thái.
- ✓ VD: Để thể hiện dòng chữ Hello World trên thanh trạng thái ta sử dụng lệnh:  

```
window.status = 'Hello World'
```



# Lý thuyết (tt)

- ✓ Lệnh `setTimeout(f, n)` quy định sau khoảng thời gian `n` mili giây hàm `f` sẽ được gọi. (`f` là chuỗi lưu lệnh cần thực hiện)
- ✓ Giả sử `str` là một chuỗi ta có
  - `str.length`: Thuộc tính cho biết độ dài chuỗi
  - `str.substr(i, n)`: lấy ra `n` ký tự kể từ vị trí thứ `i` (Ký tự đầu tiên được đánh số là 0)



## ✓ Vài lệnh khác cùng nhóm setTimeout

- `timeID = setTimeout(f, n)`
- `clearTimeout(timeID)`: Hủy `setTimeout`
- `intervalID = setInterval(f, n)`: Sau mỗi khoảng thời gian `n ms` lệnh `f` được gọi.
- `clearInterval(intervalID)`: Hủy `interval`.

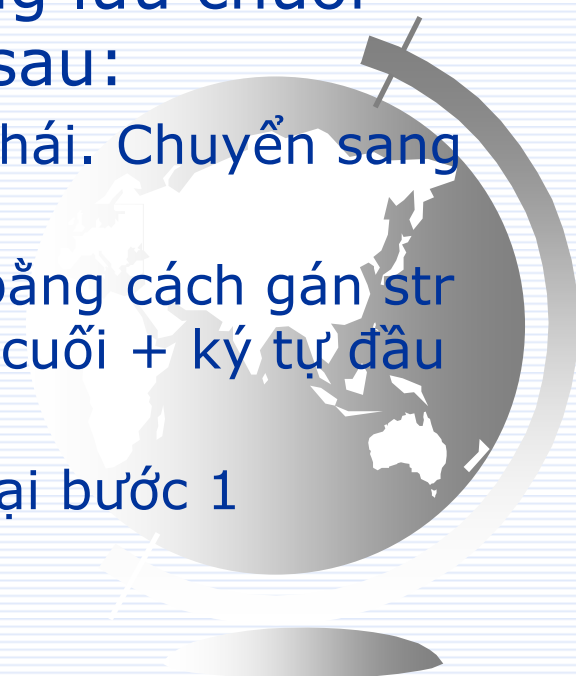


## ✓ Ý tưởng giải thuật

- Để có được cảm giác chữ chạy trên thanh trạng thái ta cần thay đổi thuộc tính status của cửa sổ bằng cách copy ký tự đầu của thanh dòng trạng thái hiện tại đưa xuống cuối cùng và lặp lại như vậy sau mỗi khoảng thời gian.

## ✓ Giải thuật: Giả sử ta có biến str đang lưu chuỗi cần chạy. Công việc thực hiện như sau:

- B1: Thể hiện chuỗi str lên thanh trạng thái. Chuyển sang bước 2
- B2: Chuyển ký tự đầu của str về cuối (bằng cách gán  $str = \text{xâu con kể từ vị trí thứ 2 của str đến cuối} + \text{ký tự đầu tiên của str}$ ). Chuyển sang bước 3
- B3: Trễ một khoảng thời gian rồi quay lại bước 1

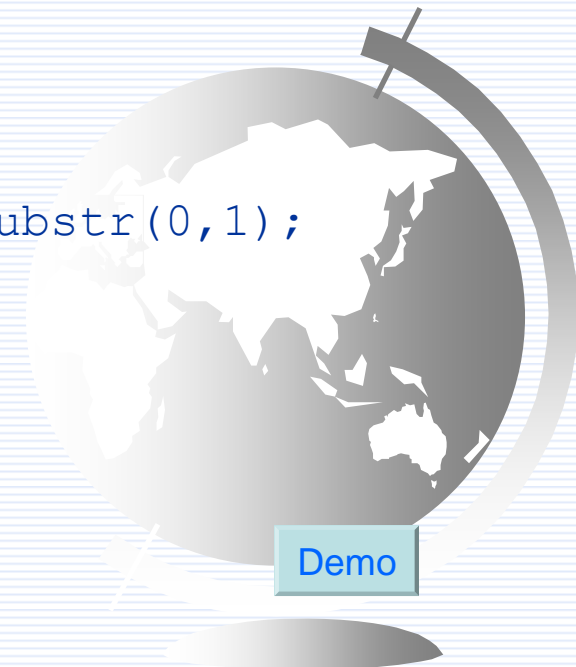


# Mã lệnh

```
<script language="javascript">
var str= 'Khoa CNTT-DHSP Ha Noi'
//Đưa vào nhìn cho đẹp (có khoảng cách trống giữa 2 lần
    chạy)
for (i=str.length; i<100; i++){
    str = str + ' '
}

function ChuChay(){
    window.status = str
    str = str.substr(1,str.length-1) + str.substr(0,1);

    setTimeout (ChuChay,100)
}
ChuChay()
</script>
```





# Phát triển

---

- ✓ Thay bằng nhiều dòng chữ chạy khác nhau (sử dụng mảng để lưu trữ)
- ✓ Chữ chạy theo nhiều cách khác nhau
- ✓ Cho chữ chạy trên thanh tiêu đề
- ✓ Cho chữ chạy trên một đối tượng khác



# Đối tượng window

---

- ✓ Tập hợp: frames[]
- ✓ Thuộc tính:
  - document
  - history
  - location
  - opener
  - status:



# Đối tượng window

---

## ✓ Sự kiện:

- onLoad
- onUnload

## ✓ Phương thức

- alert(strMessage)
- confirm(strMessage)
- prompt(strMessage, defaultText)
  
- open(url, name, option, replace)



# Đối tượng window

---

- Interval\_ID = setInterval(strLệnh, Thời\_gian)
- Timeout\_ID = setTimeout(strLệnh, Thời\_gian)
  
- clearInterval(Interval\_ID)
- clearTimeout(Timeout\_ID)



# Đối tượng document

---

## ✓ Tập hợp

- anchors[]
- links[]
- forms[]
- images[]

## ✓ Thuộc tính

- title
- cookie

## ✓ Phương thức

- getElementById(ID)
- getElementByName(ten)
- getElementByTagName(Ten\_The)



# Đối tượng document

---

- ✓ Truy xuất đến các form:
  - `window.document.tên_form`
- ✓ Truy xuất các đối tượng trong form:
  - `objForm.Tên_ĐT`
- ✓ Thuộc tính đối tượng:
  - `value`



✓ Tạo form gồm các thông tin:

- Tên truy cập
- Mật khẩu
- Nhập lại mật khẩu

✓ Kiểm tra dữ liệu vào có hợp lệ không?

- Hợp lệ:

- tên truy cập không rỗng
- 2 mật khẩu giống nhau, khác rỗng, >4 ký tự

Nếu hợp lệ được submit, ngược lại thông báo lỗi.



## Ví dụ 2

- ✓ Web link dạng combo box: Trong combo box có tên các đơn vị liên kết. Người dùng chọn đơn vị nào thì hiện web site của đơn vị đó trong 1 cửa sổ.
- ✓ Làm đồng hồ điện tử (hiện số) trên trang web.
  - Hiện thị thời gian hiện tại
  - Hiện thị khoảng thời gian đến 1 đích nào đó





