



JAVASCRIPT

TH.S LÊ QUANG SONG

Script là gì ?

● Client-Side Script

- Script được thực thi tại *Client-Side* (trình duyệt):
Thực hiện các tương tác với người dùng (tạo menu chuyển động, ...), kiểm tra dữ liệu nhập,...

● Server-Side Script

- Script được xử lý tại *Server-Side*, nhằm tạo các trang web có khả năng phát sinh nội dung động.
Một số xử lý chính: kết nối CSDL, truy cập hệ thống file trên server, phát sinh nội dung html trả về người dùng...



JavaScript là gì ?

- **JavaScript** Là ngôn ngữ *Client-side script* hoạt động trên trình duyệt của người dùng (*client*)
- **Chia sẻ xử lý** trong ứng dụng web. Giảm các xử lý không cần thiết trên server.
- Giúp **tạo các hiệu ứng, tương tác cho** trang web.



Nhúng JavaScript vào trang HTML



Định nghĩa Script trực tiếp trong trang HTML:

```
<script type="text/javascript">
```

```
<!--
```

```
// Lệnh Javascript
```

```
-->
```

```
</script>
```

Nhúng sử dụng script cài đặt từ 1 file **.js** khác:

```
<script src="abc.js"></script>
```

4



Nhúng JavaScript vào trang HTML



- Web Browser sẽ thực thi **<script>** khi load trang web theo thứ tự từ *trên xuống* dưới.
- Đối với *Source code JavaScript* có thể đặt trong các file *.js* sẽ được *nhúng* vào file HTML trước khi hoạt động.
- Các đoạn *code JavaScript* được Browser xử *cùng thứ tự* với các thẻ *HTML*. Trừ các **hàm** (*function*) chỉ được thực thi khi *có lời gọi hàm*.



Nhúng JavaScript vào trang HTML

```
<html>
  <head>
    <script type="text/javascript">
      some statements
    </script>
  </head>
  <body>
    <script type="text/javascript">
      some statements
    </script>
    <script src="Tên_file_script.js"> |</script>
    <script type="text/javascript">
      // gọi thực hiện các phương thức được định nghĩa
      // trong "Tên_file_script.js"
    </script>
  </body>
</html>
```



Nhúng JavaScript vào trang HTML



- Đặt giữa tag `<head>` và `</head>`: script sẽ thực thi ngay khi trang web được mở.
- Đặt giữa tag `<body>` và `</body>`: script trong phần body được thực thi khi trang web đang mở (sau khi thực thi các đoạn script có trong phần `<head>`).
- Số lượng đoạn client-script chèn vào trang không hạn chế.

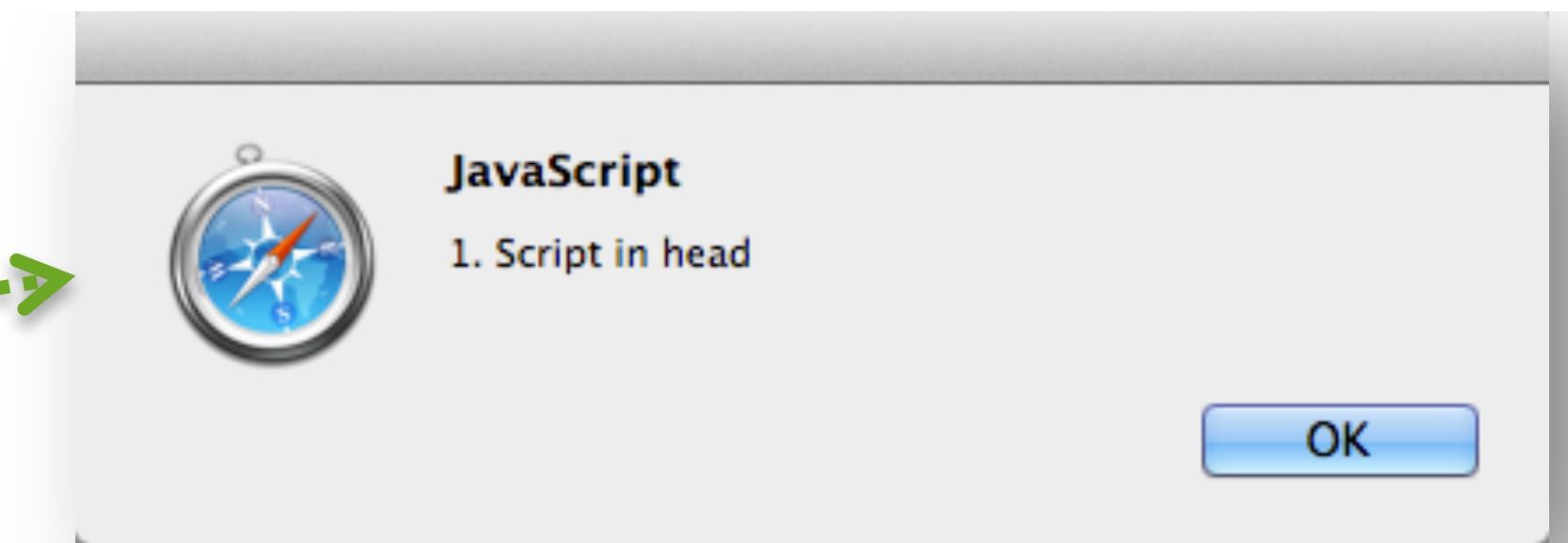


Nhúng JavaScript vào trang HTML

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
  <head>
    <meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
    <title>Title of website</title>
    <script type="text/javascript">
      alert("1. Script in head");
    </script>
  </head>
```

```
<body>
  <script type="text/javascript">
    alert("2. Script in body");
  </script>
```

```
<script type="text/javascript" src="myScript.js" ></script>
<script type="text/javascript">
  showMessage();
</script>
</body>
</html>
```



Đọc lập trình t



KIỂM TRA DỮ LIỆU VÀ CÚ PHÁP

NGÔN NGỮ JAVASCRIPT



Biến số trong Javascript



● Cách đặt tên biến

- Bắt đầu bằng một chữ cái hoặc dấu _
- A..Z,a..z,0..9,_ : phân biệt HOA, Thường

Ví dụ: **var chuoiSo;**

var ChuoiSo;

● Khai báo biến

- Sử dụng từ khóa **var**

Ví dụ: **var count = 10, amount;**

- *Không cần khai báo biến trước khi sử dụng*, biến thật sự tồn tại khi bắt đầu sử dụng lần đầu tiên.

- Biến *không cần khai báo kiểu dữ liệu* vì biến trong javascript *không có kiểu dữ liệu nhất định*



Kiểu dữ liệu trong Javascript



Kiểu dữ liệu	Ví dụ	Mô tả
Object	var listBooks = new Array(10) ;	Trước khi sử dụng, phải cấp phát bằng từ khóa new
String	“The cow jumped over the moon.” “40”	Chứa được chuỗi unicode Chuỗi rỗng “”
Number	0.066218 12	Theo chuẩn IEEE 754
boolean	true / false	
undefined	var myVariable ;	myVariable = undefined
null	connection.Close();	connection = null



Đổi kiểu dữ liệu



- Biến tự đổi kiểu dữ liệu khi giá trị mà nó lưu trữ thay đổi

Ví dụ:

```
var x = 10;           // x kiểu Number  
x = "hello world !"; // x kiểu String
```

- Có thể cộng 2 biến khác kiểu dữ liệu

Ví dụ:

```
var x;  
x = "12" + 34.5; // KQ: x = "1234.5"
```

- Hàm **parselnt(...)**, **parseFloat(...)**: Đổi KDL từ chuỗi sang số.



Hàm trong javascript

- Dạng thức khai báo chung:

```
function Tên_hàm(thamso1, thamso2, ...)  
{  
    ...  
}
```

- Hàm có giá trị trả về:

```
function Tên_hàm(thamso1, thamso2, ...)  
{  
    ...  
    return (value);  
}
```



Ví dụ Hàm trong javascript

- Ví dụ:

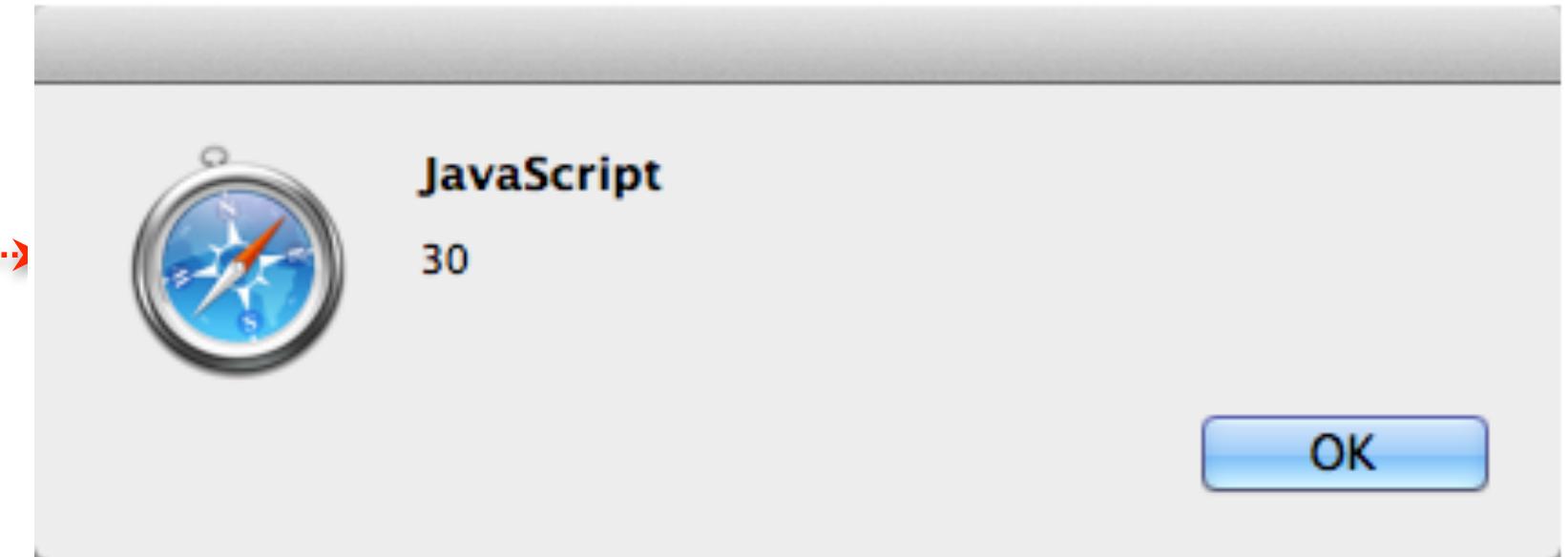
```
function Sum(x, y)
```

```
{
```

```
    tong = x + y;
```

```
    return tong;
```

```
}
```



- Gọi hàm:

```
var x = Sum(10, 20);
```

```
alert(x);
```



Các qui tắc chung

- Khối lệnh được bao trong dấu {}
- Mỗi lệnh nên kết thúc bằng dấu ;
- Cách ghi chú thích:
 - // Chú thích 1 dòng
 - /* Chú thích
nhiều dòng */

Một số sự kiện thông dụng

- Các sự kiện được hỗ trợ bởi hầu hết các đối tượng
 - **onClick**
 - **onFocus**
 - **onChange**
 - **onBlur**
 - **onMouseOver**
 - **onMouseOut**
 - **onMouseDown**
 - **onMouseUp**
 - **onLoad**
 - **onSubmit**



ĐỐI TƯỢNG DOM



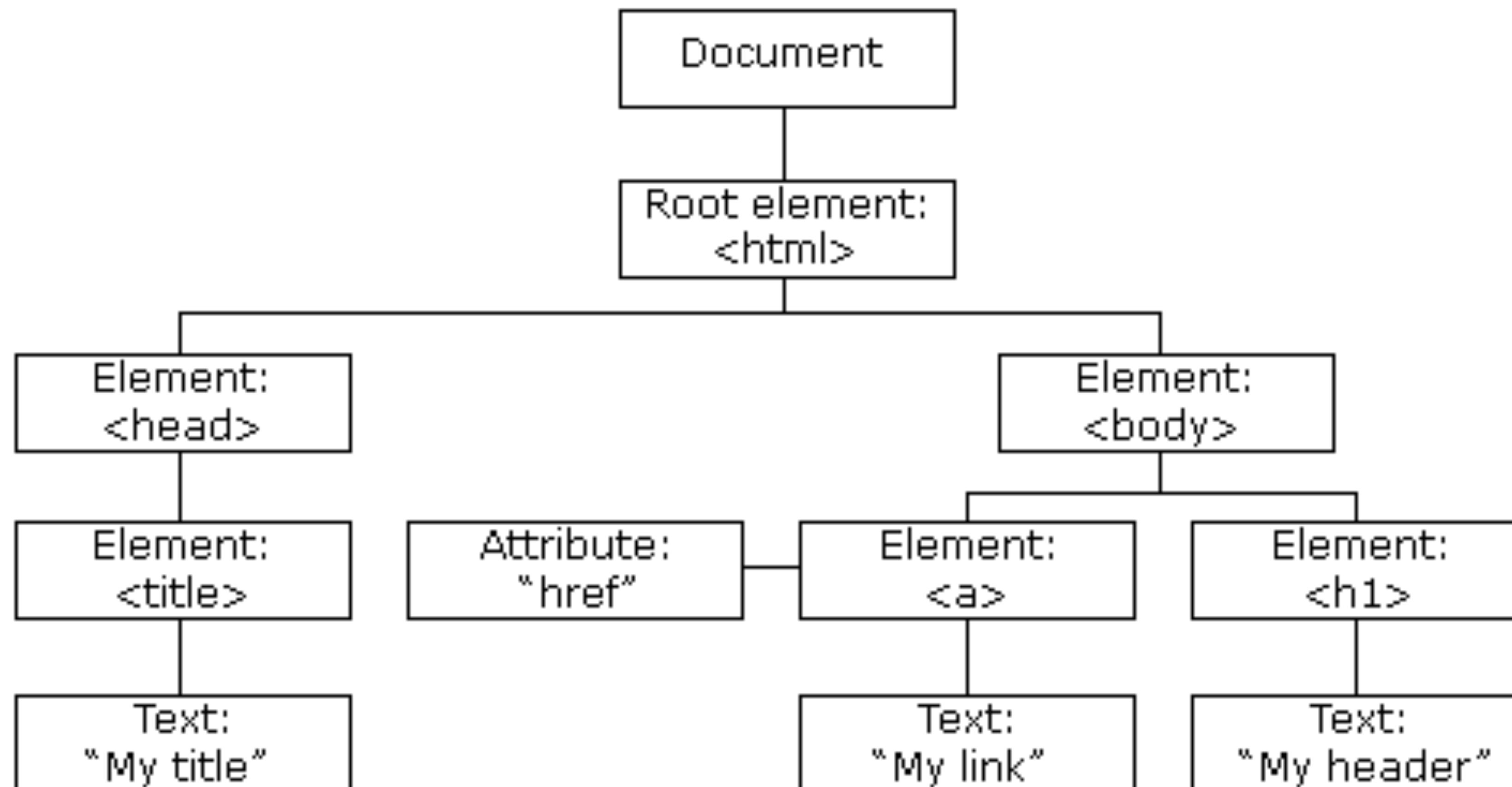
Đối tượng HTML DOM



- **DOM** = Document Object Model
- Là tập hợp các đối tượng HTML chuẩn được dùng để **truy xuất** và **thay đổi thành phần** HTML trong trang web (thay đổi nội dung tài liệu của trang)
- Một số đối tượng của DOM: **window, document, history, link, form, frame, location, event, ...**



ĐỐI TƯỢNG DOM



Đối tượng Window - DOM



- Là thể hiện của đối tượng **cửa sổ trình duyệt**
- Tồn tại khi mở 1 tài liệu HTML
- Sử dụng để truy cập thông tin của các đối tượng trên cửa sổ trình duyệt (tên trình duyệt, phiên bản trình duyệt, thanh tiêu đề, thanh trạng thái ...)



ĐỐI TƯỢNG DOM



Đối tượng Window - DOM



- **Properties**

- **document**
- event
- history
- location
- name
- navigator
- screen
- status

- **Methods**

- Alert
- Confirm
- Prompt
- Blur
- close
- Focus
- open



Đối tượng Document - DOM



- Biểu diễn cho **nội dung trang HTML** đang được hiển thị trên trình duyệt
- Dùng để lấy thông tin về tài liệu, các thành phần HTML và xử lý sự kiện



Đối tượng Document - DOM



- **Properties**
 - aLinkColor
 - bgColor
 - **body**
 - fgColor
 - linkColor
 - title
 - **URL**
 - vlinkColor
 - forms[]
 - images[]
 - **childNodes[]**
- **Methods**
 - **documentElement**
 - cookie
 -
 - close
 - open
 - **createTextNode(" text ")**
 - **createElement("HTMLtag")**
 - **getElementById("id")**
 - ...



TÌM PHẦN TỬ/THẺ (ELEMENT) HTML



PHƯƠNG THỨC	MÔ TẢ
<code>document.getElementById(id)</code>	Tìm thẻ thông qua Id của thẻ
<code>document.getElementsByTagName(name)</code>	Tìm TẤT CẢ các thẻ thông qua tên thẻ
<code>document.getElementsByClassName(name)</code>	Tìm TẤT CẢ các thẻ thông qua CSS class

3 4 5



THAY ĐỔI NỘI DUNG HTML CỦA THẺ



PHƯƠNG THỨC

MÔ TẢ

`element.innerHTML = new html content`

Thay đổi nội dung HTML bên trong của thẻ

`element.style.property = new style`

Thay đổi Style cho thẻ



THÊM VÀ XOÁ THẺ

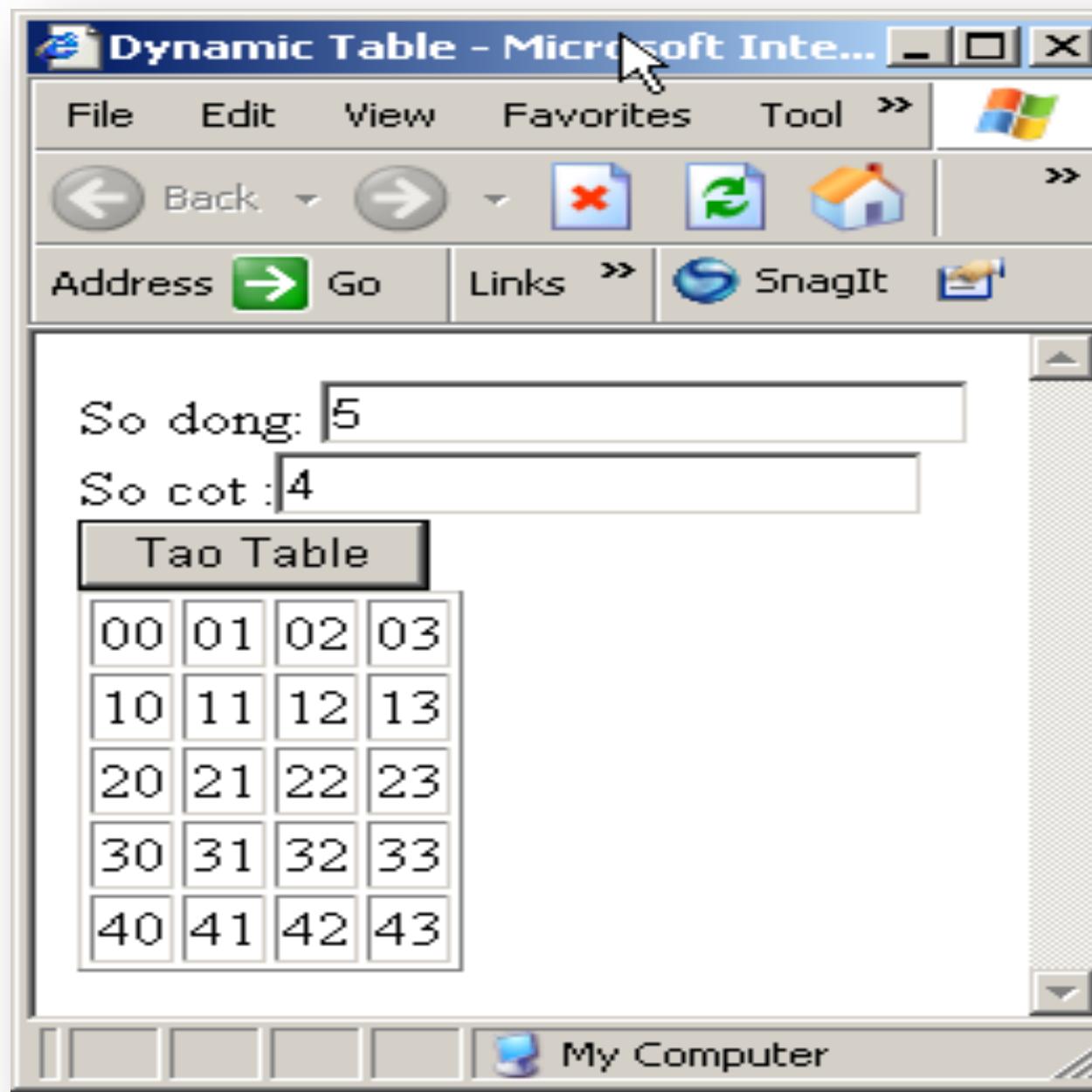


PHƯƠNG THỨC	MÔ TẢ
<code>document.createElement(element)</code>	Tạo một thẻ mới
<code>document.removeChild(element)</code>	Xoá một thẻ
<code>document.appendChild(element)</code>	Thêm một thẻ
<code>document.replaceChild(element)</code>	Thay thế một thẻ
<code>document.write(text)</code>	Viết ngay vào HTML



Ví dụ tạo table động

- Viết trang web cho phép tạo table có số dòng, số cột do người dùng nhập vào.



- + `Document.createElement(...)`: Tạo một đối tượng thẻ DOM HTML
- + `Object.appendChild(...)`: Thêm một đối tượng thẻ DOM HTML như là nút con.
- + Đối tượng Number trong Javascript
 - `parseInt(...)`
 - `parseFloat(...)`



Đối tượng Date trong Javascript

// Đối tượng thời gian hiện tại

```
var d = new Date();
```

```
d.getDate();
```

// getDate() lấy ngày (1 - 31)

```
d.getDay();
```

// getDay() lấy ngày trong tuần (0-6)

```
d.getFullYear();
```

// getFullYear() lấy năm đầy đủ (YYYY)

```
d.getYear();
```

// getYear() lấy 2 số cuối của năm (YY)

```
d.getHours();
```

// getHours() lấy số giờ (0 - 23)

```
d.getMilliseconds();
```

// getMiliSeconds() lấy số mili giây (0 - 999)

```
d.getMinutes();
```

// getMinutes() lấy số phút (0 - 59)

```
d.getMonth();
```

// getMonth() lấy tháng (0 - 11)

```
d.getSeconds();
```

// getSeconds() lấy số giây (0 - 59)

```
d.getTime();
```

// getTime() thời gian đã được chuyển đổi sang dạng mili giây



Các phương thức xử lý chuỗi

1. Tìm kiếm chuỗi con

Có ba phương thức thường dùng để tìm kiếm chuỗi con trong JavaScript như sau:

`indexOf()`

`lastIndexOf()`

`search()`

Để tìm kiếm chuỗi con, chúng ta sử dụng phương thức `String.indexOf(str)`, trong đó `str` là chuỗi con và `String` là chuỗi cha.

Phương thức này sẽ trả về kết quả là **vị trí xuất hiện đầu tiên của chuỗi (bắt đầu là vị trí 0)**, nếu không tìm thấy chuỗi con thì nó sẽ trả về -1.



Các phương thức xử lý chuỗi



Phương thức lastIndexOf()

Trường hợp nếu chuỗi con xuất hiện nhiều lần trong chuỗi cha thì kết quả cũng trả về vị trí xuất hiện của chuỗi con đầu tiên.

Do đó, để lấy vị trí của chuỗi con cuối cùng trong chuỗi cha thì chúng ta phải sử dụng phương thức String.lastIndexOf(str).

Phương thức này sẽ trả về vị trí xuất hiện của chuỗi con cuối cùng và trả về -1 nếu không tìm thấy.



Các phương thức xử lý chuỗi

Phương thức search()

Ngoài hai phương thức trên, bạn có thể sử dụng phương thức **string.search(str)** để tìm kiếm, tác dụng của nó cũng giống như phương thức **string.indexOf(str)**.

2. Cắt chuỗi con

Nếu bạn muốn cắt một chuỗi con từ chuỗi cha thì có thể sử dụng ba phương thức sau:

slice(start, end)

substring(start, end)

substr(start, length)

Ghi chú: Tất cả các vị trí của chuỗi đều bắt đầu từ 0, vì vậy khi tính toán vị trí phải thật cẩn thận.



Các phương thức xử lý chuỗi



Phương thức slice()

Phương thức **slice()** có *hai tham số truyền vào*:

start: vị trí bắt đầu

end: vị trí kết thúc

Nếu tham số truyền vào là số âm thì nó sẽ tính ngược lại, nghĩa là nó sẽ đếm từ cuối lên.

Nếu bạn chỉ truyền một tham số đầu tiên thì nó sẽ tự hiểu vị trí end là vị trí cuối cùng.



Các phương thức xử lý chuỗi

Phương thức substring()

Phương thức **substring()** có cách sử dụng giống với phương thức **slice()**, tuy nhiên tham số truyền vào phương thức **substring()** phải luôn luôn lớn hơn 0.

Phương thức substr()

Phương thức **substr()** có hai tham số là **start** và **length**, trong đó **start** là vị trí bắt đầu và **length** là số ký tự muốn lấy bắt đầu từ vị trí **start**. Nếu bạn truyền tham số **start** là số âm thì nó sẽ tính từ cuối trở lên, còn tham số **length** phải luôn luôn là số dương.



DOM & CSS

Thiết lập CSS bằng JavaScript

Cú pháp:

```
document.getElementById("object").style.[tên thuộc tính] = '[giá trị]';
```

Lấy giá trị CSS bằng JavaScript

```
var value = document.getElementById("object").style.[tên thuộc tính];
```

Nếu thuộc tính có dấu gạch ngang như: *font-size, line-height, margin-bottom* thì thuộc tính đó trong *style* sẽ có tên là *fontSize, lineHeight, marginBottom*



Chuỗi, hộp thoại trong js

<http://nhatdesign.com/hoc-javascript/cac-phuong-thuc-xu-ly-chuoi-trong-javascript.html>

<http://nhatdesign.com/hoc-javascript/hop-thoai-thong-bao-trong-javascript.html>

<http://tamnguyen.com.vn/Xu-ly-ngay-thang-voi-Javascript.html>



Dom CSS - animation , hide, show, event



<http://nhatdesign.com/hoc-javascript/dom-css-trong-javascript.html>

http://www.w3schools.com/js/js_htmldom_css.asp

http://www.w3schools.com/js/js_htmldom_animate.asp

Thay đổi source , cái đèn : http://www.w3schools.com/js/js_intro.asp



Kiểm tra Form hợp lệ



Học lập trình trực tuyến tại myclass.vn



Ví dụ Form Đăng nhập, Đăng Kí



Học lập trình trực tuyến tại myclass.vn

