

BÀI 7

JAVASCRIPT



PHAN LONG

FB: PHANLONGIT

JavaScript - tổng quan

1. Giới thiệu:

- HTML: Chứa nội dung (các thành phần chính, ..)
 - CSS: Trang trí, trình bày, định dạng,
 - JavaScript: Hành động của trang web
- > JavaScript hồi đáp lại các tương tác của người dùng

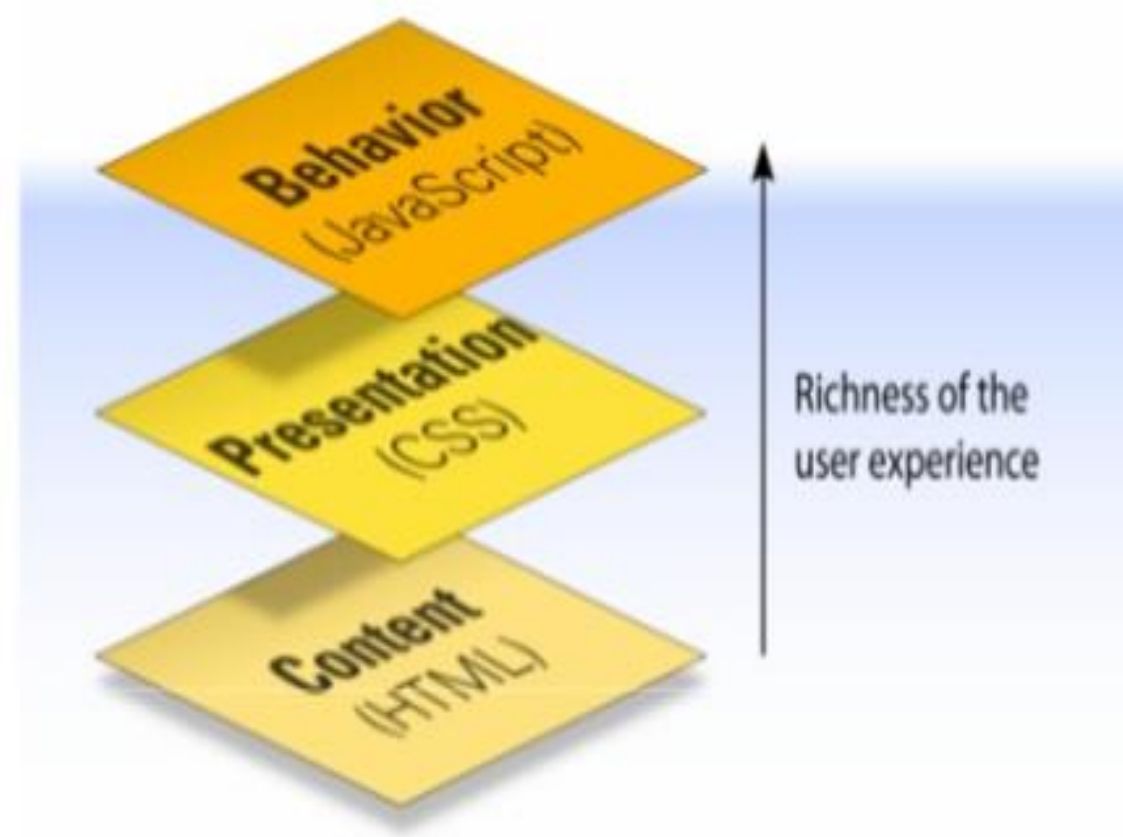


Figure 1.2. The three layers of the Web

JavaScript - tổng quan

- Là ngôn ngữ script ở client, dùng để xử lý và tương tác với các thành phần HTML.
- Là dạng ngôn ngữ thông dịch
- Không liên quan đến ngôn ngữ Java
 - JS được phát triển bởi Netscape
 - Chỉ thực thi trên trình duyệt



Brendan Eich

Mocha → LiveScript → JavaScript

TỔNG QUAN

1. WHAT? (Học về cái gì?):

- Học từ đầu ngôn ngữ Javascript, (phải biết html, css)
- Học lý thuyết.
- Học thực hành các bài tập dựa vào lý thuyết
- Học javascript cho phần front end thay thế cho jquery (nâng cao javascript)

2. WHY?

- Tăng tốc độ load web.
- JS là nền tảng cho các công nghệ mới hiện nay (không cần load thêm thư viện do javascript được cung cấp sẵn trong window hoặc mac)
- Là nền tảng cho Back End (React JS hoặc NodeJS)

3. HOW? (Học thế nào)

- Học từ đầu (nắm vững lý thuyết) (sử dụng w3school)
- Áp dụng vào làm bài tập.
- Học javascript thay thế cho jquery (cao thủ, hiểu cách viết hàm, xây dựng hiệu ứng cho phần front end)

I. CÀI ĐẶT (Extension, plugin hỗ trợ)

1. Extension, plugin

- **Fira Code** (tạo code trông đẹp mắt)
- **Indenticator** (giúp nhận biết dễ dàng các block code đang được sử dụng).
- **Javascript (ES6) Code Snippet**: những snippet này có thể kích hoạt từ các severl tập tin khác nhau.
 - . Javascript (.js)
 - . TypeScript (.ts)
 - . Javascript React (.jsx)
 - . Typescript React (.tsx)
 - . Html (.Html)
 - . Vue (.vue)

2. Cài đặt Template: tự làm

II. Tổng quan

1. Chức năng của JS:

- Cung cấp cho người thiết kế HTML công cụ lập trình
- Cho phép đặt đoạn văn bản động vào trang web
- Có thể tác động các sự kiện trong trang HTML
- Có thể đọc/ghi các thành phần của HTML
- Dùng để check dữ liệu từ người dùng
- Có thể check phiên bản trình duyệt
- Có thể thao tác cookie của trang web.

JavaScript - tổng quan

2. Các bước thực thi của JS

1. Trình duyệt tải trang web về
2. Trình duyệt kiểm tra xem có mã JS trong web hay không
3. Nếu có, trình duyệt sẽ chuyển mã JS cho bộ thông dịch
4. Bộ thông dịch xử lý và thực thi các mã lệnh JS
5. Các mã lệnh có thể tác động đến các thành phần của trang web.
6. Trình duyệt hiển thị toàn bộ nội dung web.
7. Javascript được tạo bởi các câu lệnh và kết thúc bằng dấu ;

Ví dụ: <https://themes.muffingroup.com/be/webdesign/>

JavaScript - tổng quan

3. Cách đặt mã lệnh JS vào trang web

a. Internal: đặt trong head hay body

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

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

b. External: tạo tập tin bên ngoài và liên kết tập tin đó trong phần head.

```
<head>
  <script src="YourJsFile.js"></script>
</head>
```


II. Toán tử (operator)

Các toán tử toán học

$y = 5$

Toán tử	Mô tả	Ví dụ	Kết quả
+	Cộng (addition)	$x = y + 2$	$x = 7$
-	Trừ (subtraction)	$x = y - 2$	$x = 3$
*	Nhân (multiplication)	$x = y * 2$	$x = 10$
/	Chia (division)	$x = y / 2$	$x = 2.5$
%	Chia lấy dư (modulus)	$x = y \% 2$	$x = 1$
++	Tăng (increment)	$x = ++y$	$x = 6$
--	Giảm (decrement)	$x = --y$	$x = 4$

Toán tử (operator)

Các toán tử gán

Toán tử	Ví dụ	Tương tự với
=	$x=y$	
+=	$x+=y$	$x=x+y$
-=	$x-=y$	$x=x-y$
=	$x=y$	$x=x*y$
/=	$x/=y$	$x=x/y$
%=	$x\%=y$	$x=x\%y$

Toán tử (operator)

Các toán tử so sánh

Toán tử	Mô tả	Ví dụ
==	bằng	x==8
===	bằng (so sánh vừa giá trị vừa kiểu biến)	x===5
!=	không bằng	x!=8
>	lớn hơn	x>8
<	nhỏ hơn	x<8
>=	lớn hơn hay bằng	x>=8
<=	nhỏ hơn hay bằng	x<=8

Toán tử (operator)

Toán tử logic

Toán tử	Mô tả	Ví dụ
&&	and	(x < 10 && y > 1) → true
	or	(x==5 y==5) → false
!	not	!(x==y) → true

Toán tử điều kiện

Cú pháp: variablename=(condition)?value1:value2

Ví dụ: greeting=(visitor=="PRES")?"Dear President ":"Dear ";

Điều kiện đúng chọn value1, ngược lại chọn value2

IV. Nhập xuất trong JavaScript

1. Document.write()

- Cho phép chúng ta viết các nội dung vào tệp HTML.
- Cú pháp

```
<script>  
    document.write(Nội_dung);  
</script>
```

2. alert()

Phương thức alert() sẽ tạo ra một hộp cảnh báo để hiển thị nội dung

Cú pháp

```
<script>  
    alert("Nội dung hiển thị");  
</script>
```

Nhập xuất trong JavaScript

Ví dụ minh họa sử dụng javascript

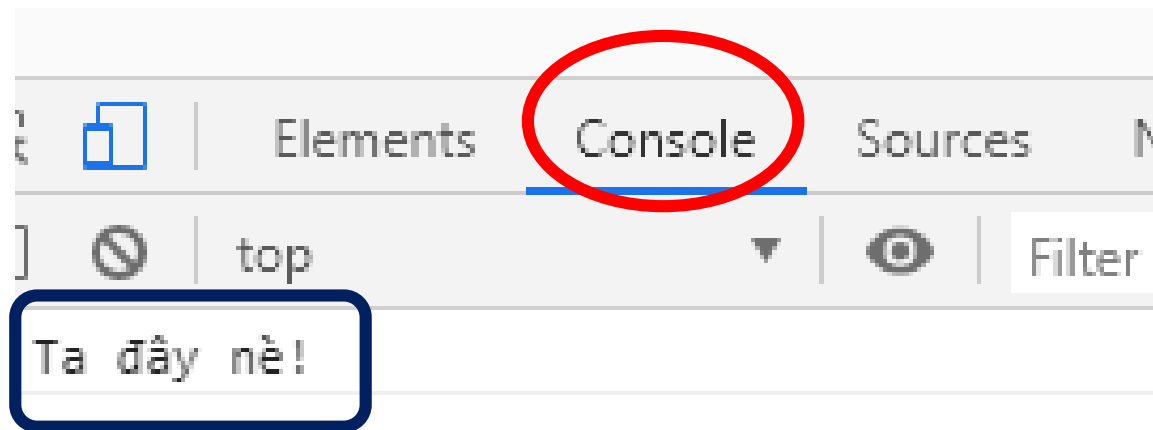
- Trường hợp 1: sử dụng câu lệnh in ra màn hình (console)

// ấn F12 kiểm tra

Cú pháp; `console.log ('nội dung cần in ra')`

Ví dụ: trong file js chèn nội câu lệnh

`console.log('Ta đây nè!')`



Giả sử viết sai lệnh ->

báo lỗi dòng màu đỏ

=> Hàm dùng để check lỗi

Cấu trúc dữ liệu - biến

- Biến được dùng để lưu trữ dữ liệu.
- Khai báo biến

Cú pháp	Ví dụ
var <tên biến>;	var n; n = 1;
var <tên biến> = <giá trị>;	var s = "JavaScript string";
<tên biến> = <giá trị>;	PI = 3.14;

Hoặc có thể không cần var

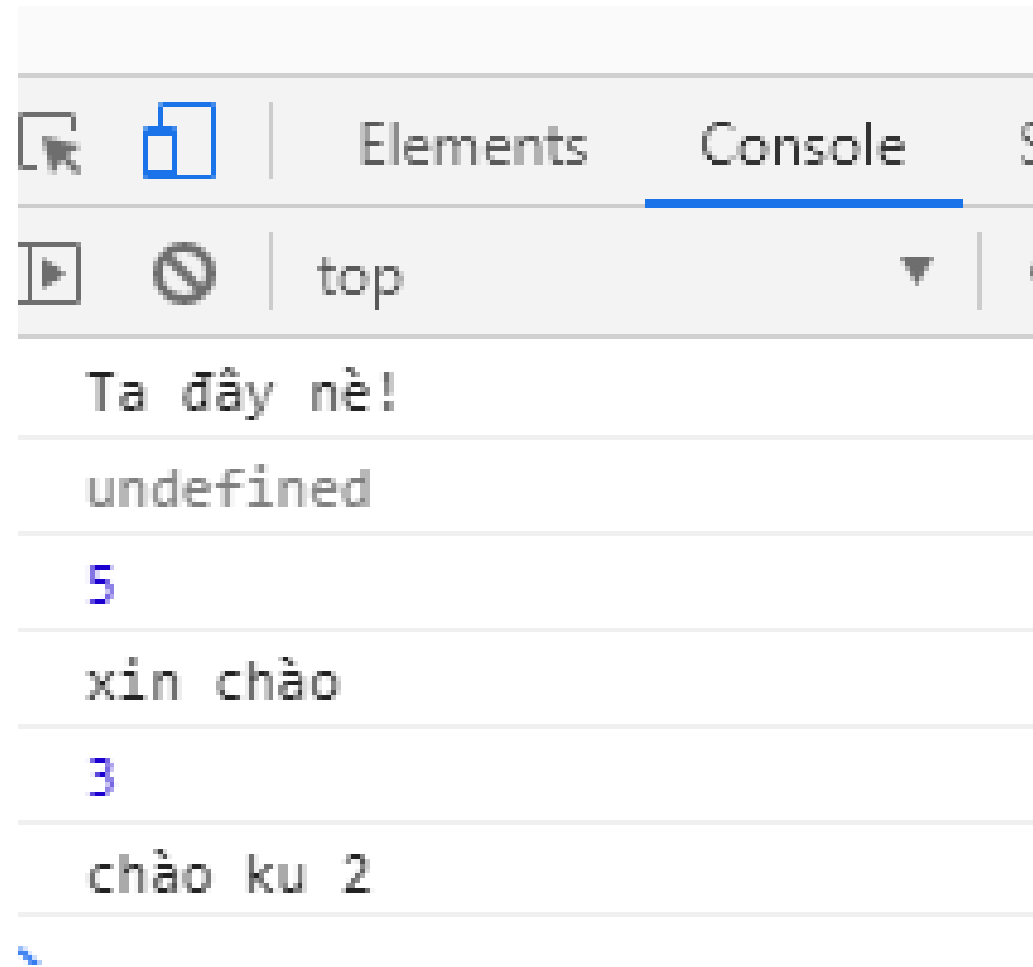
- Cách đặt tên biến
 - Dùng các ký tự a..z, A..Z, 1..9, dấu gạch dưới '_', dấu '\$'
 - Tên biến không trùng với từ khóa JS
 - Tên biến không bắt đầu bởi con số
 - Tên biến không có ký tự khoảng trắng
 - Tên biến là case sensitive.

Khai báo sử dụng biến

- Kiểu dữ liệu của biến
 - JS không quy định kiểu biến khi khai báo biến, kiểu của biến sẽ được tự động xác định khi gán dữ liệu cho biến
- Các kiểu dữ liệu của JS
 - Kiểu số (number): số nguyên, số thực
 - Kiểu chuỗi (string)
 - Kiểu boolean: true/false
 - Kiểu đối tượng (object)
 - Kiểu hàm (function)

Ví dụ:

```
console.log("Ta đây nè!");  
var n;  
console.log(n);  
var n2=5; //vừa khai báo vừa gán giá trị  
console.log(n2);  
n3="xin chào";  
console.log(n3);  
//sử dụng toán tử  
var x=1, y=2;  
console.log(x+y);  
var z="chào ku ";  
console.log(z+y); // nối chữ.
```



Ví dụ 2:

Kiểm tra gọi hàm cho đối tượng

```
<button onclick="test()">test</button>
```

```
.js=>
```

```
function test()
```

```
{
```

```
    console.log('test hàm');
```

```
}
```

Khi thực thi



Lưu ý

- Javascript phân biệt ký tự Hoa , thường
- Đặt tên cho biến, hàm không được bắt đầu bằng số
- Biểu thức so sánh:

```
if(điều kiện)
{
    đúng
}
else
{
    sai
}
```

Cấu trúc điều khiển

Cấu trúc điều khiển if

```
if (biểu_thức_điều_kiện_1)
{
    //khởi lệnh được thực hiện nếu biểu thức 1 đúng;
}
else if (biểu_thức_điều_kiện_2)
{
    //khởi lệnh được thực hiện nếu biểu thức 2 đúng;
}
else
{
    //khởi lệnh được thực hiện nếu cả hai biểu thức trên đều không đúng;
}
```


Cấu trúc điều khiển

Cấu trúc điều khiển switch

```
switch (biểu_thức_điều_kiện)
{
    case kết_quả_1 :
        //khởi lệnh cần thực hiện nếu biểu_thức_điều_kiện bằng kết_quả_1;
        break;
    case kết_quả_2 :
        //khởi lệnh cần thực hiện nếu biểu_thức_điều_kiện bằng kết_quả_2;
        break;
    default :
        //khởi lệnh cần thực hiện nếu biểu_thức_điều_kiện cho ra một kết quả khác;
}
```

Cấu trúc điều khiển

Cấu trúc điều khiển for

```
for (bt_khởi_tạo; bt_điều_kiện; bt_thay_đổi_giá_trị)
{
    //Khởi lệnh cần lặp;
}
for (biên in đối tượng)
{
    //Khởi lệnh cần lặp;
}
var x;
var mycars = ["Saab", "Volvo", "BMW"];
for (x in mycars){
    document.write(mycars[x]);
}
```

Cấu trúc điều khiển

Cấu trúc điều khiển while

```
while (biểu_thức_điều_kiện)
{
    //khởi lệnh lặp cần thực hiện nếu
    //biểu_thức_điều_kiện trả về true;
}
```

Cấu trúc điều khiển do while

```
do
{
    //khởi lệnh lặp cần thực hiện
} while (biểu_thức_điều_kiện);
```

KIỂU DỮ LIỆU MẢNG TRONG JAVASCRIPT

Trong JavaScript Array là đối tượng toàn cục được dùng để xây dựng nên các mảng; là những đối tượng cấp cao và giống một dạng danh sách. Các thao tác trên mảng:

- Khai báo mảng: `var fruits = ['Apple', 'Banana'];`
`=> console.log(fruits.length); // 2`

Cách 2: `mang[]`; liệt kê các phần tử mảng

`mang[0]='mot'; mang[1]='hai'; mang[2]='ba';`

- Truy cập mảng: truy cập bằng chỉ mục: `tenmang[chỉ số]`

`var first = fruits[0]; // Apple`

- Thêm phần tử vào cuối Mảng: `tenmang.push()`

`var newLength = fruits.push('Orange');`
`// ["Apple", "Banana", "Orange"]`

KIỂU DỮ LIỆU MẢNG TRONG JAVASCRIPT

- Thêm phần tử vào đầu Mảng: `tenmang.unshift()`

```
var newLength = fruits.unshift('Strawberry')  
// ["Strawberry", "Banana"];
```

- Xóa phần tử cuối Mảng: `tenmang.pop()`

```
var last = fruits.pop(); // xoá bỏ Orange (từ vị trí cuối)
```

- Xóa phần tử đầu Mảng: `tenmang.shift()`

```
var first = fruits.shift();  
// xoá bỏ Apple từ vị trí đầu
```

Tìm chỉ mục của một phần tử trong Mảng: `fruits.indexOf`

```
var pos = fruits.indexOf('Banana'); // 1
```

-Xóa một phần tử bằng chỉ mục của nó: `tenmang.splice(pos, số);`

```
var removedItem = fruits.splice(pos, 1);
```


Kiểu dữ liệu đối tượng (OBJECT)

- Khái niệm: Về mặt định nghĩa, một đối tượng (một object) là một **danh sách các item**, mỗi item là một cặp name-value, trong đó value có thể là: các kiểu dữ liệu cơ bản, function, hay cũng có thể là một object khác (kiểu dữ liệu phức hợp).

- Khai báo:

`var tenObject = {các thuộc tính} hoặc var tenObject = new object()`

Ví dụ:

```
var conmeo={  
    maulong:"trang",  
    ten:"Tom",  
    cannang:"3 ký"  
}
```

Phương thức

Phương thức là những hành động (có thể hiểu là hàm) của đối tượng. Ví dụ trong đối tượng Conmeo thì mình cần 3 phương thức là: ngủ, ăn, chạy

Hàm (function)

- Hàm là khối câu lệnh với một danh sách tham số (hoặc không có tham số)
- Trong JS cho phép hàm không tên
- Hàm có thể trả về một giá trị

```
function tên_hàm(đôi_số_1, đôi_số_2)
{
    //các câu lệnh cần thực hiện mỗi khi hàm được gọi;
    return giá_trị_cần_trả_về;
}
```

Hàm (function)

- Gọi hàm trong JS
 - Gọi tên hàm và truyền tương ứng các tham số vào
 - VD: tên_hàm(đối số 1, đối số 2)
 - Khi gọi hàm không nhất thiết phải truyền đủ các đối số khi định nghĩa hàm. Nếu số đối số ít hơn khi định nghĩa hàm, khi đó những đối số không được truyền cho hàm sẽ mang giá trị **undefined**
 - Các kiểu cơ bản sẽ được truyền vào hàm theo giá trị, đối tượng sẽ được chuyển vào hàm theo tham chiếu.

Hàm (function)

```
<script type="text/javascript">  
function myfunction(txt)  
{  
    alert(txt);  
}  
</script>
```



```
<body>  
    <form>  
        <input type="button" onclick="myfunction('Hello')"  
            value="Call function">  
    </form>  
</body>
```

Hàm (function)

```
<script type="text/javascript">  
    //tính diện tích hình chữ nhật  
    function dienTichHinhChuNhat(rong, dai) {  
        var ketqua= rong * dai;  
        return ketqua;  
    }  
</script>  
  
<script type="text/javascript">  
    var dientich = dienTichHinhChuNhat(100, 150);  
    document.write(dientich);  
</script>
```