

- JavaScript Objects
- JS HTML DOM
- JS HTML BOM
- JSON

# JavaScript Objects

In JavaScript, objects are king.

If you understand objects, you understand  
JavaScript.

# JavaScript Objects

**Trong JavaScript, hầu hết mọi thứ đều object.**

- Booleans có thể là một object (Nếu định nghĩa bằng từ khóa **new**)
- Numbers có thể là một object (nếu được định nghĩa bằng từ khóa **new**)
- Strings có thể là một object (nếu được định nghĩa bằng từ khóa **new**)
- Dates luôn objects
- Maths luôn objects
- Regular expressions là objects
- Arrays luôn objects
- Functions luôn objects
- Objects luôn objects

Tất cả các giá trị, ngoại trừ nguyên thủy đều là object.

# Objects are Variables Containing Variables

Các biến JavaScript có thể chứa các giá trị đơn:

```
var person = "John Doe";
```

# Objects are Variables Containing Variables

Đối tượng cũng là 1 biến nhưng đối tượng chứa nhiều giá trị.

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

# JavaScript Objects

Một đối tượng trong JS là một tập hợp các giá trị được đặt tên

# Object Properties

Các giá trị được đặt tên trong JS object, được gọi là **properties**.

Property	Value
firstName	John
lastName	Doe
age	50
eyeColor	blue

# Object Methods

Đối tượng bao gồm các thuộc tính và phương thức:

Property	Value
firstName	John
lastName	Doe
age	50
eyeColor	blue
fullName	<code>function() {return this.firstName + " " + this.lastName;}</code>



# Creating a JavaScript Object

- Tạo object sử dụng object literal.
- Tạo object sử dụng từ khóa new.
- object constructor, and then create objects of the constructed type.

# Using an Object Literal

- Có thể khai báo các thuộc tính và phương thức của Object trong 1 dòng hoặc nhiều dòng

```
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};
```

```
var person = {  
  firstName:"John",  
  lastName:"Doe",  
  age:50,  
  eyeColor:"blue"  
};
```

# Using the JavaScript Keyword new

- Có thể khởi tạo một đối tượng với từ khóa new:

```
var person = new Object();  
person.firstName = "John";  
person.lastName = "Doe";  
person.age = 50;  
person.eyeColor = "blue";
```

# JavaScript Objects are Mutable

- Trong JS 1 object là một tham chiếu, không phải là một tham trị.
- Các đối tượng trong JS có thể biến đổi.

**Lưu ý:** trong Javascript các biến thì luôn không thay đổi, chỉ có các đối tượng JavaScript mới thay đổi.

# The **this** Keyword

- Sử dụng trong một hàm (function).
- Sử dụng trong một đối tượng (object).

“this” không phải là một biến, nó là một từ khóa. Giá trị của “this” không thể thay đổi.

# The **this** Keyword

“this” is God, and “this” is weird

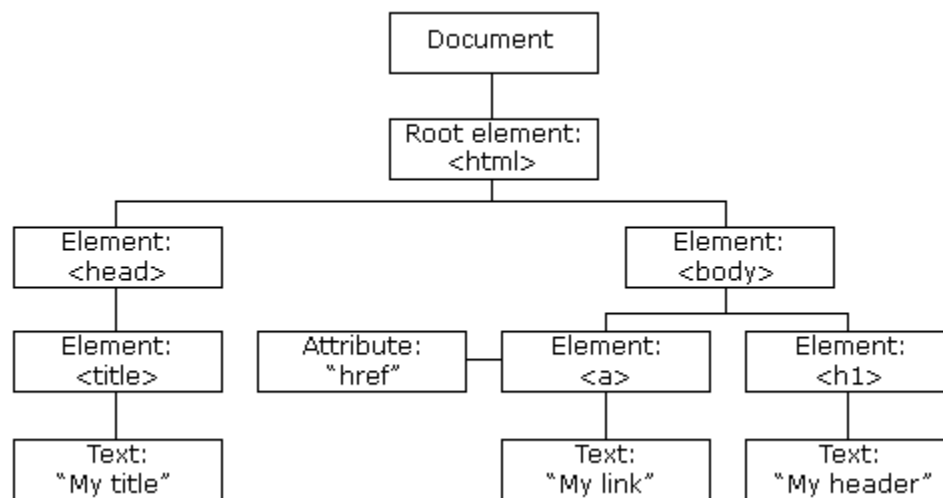
# JavaScript Constructors

```
var x1 = new Object();    // A new Object object
var x2 = new String();    // A new String object
var x3 = new Number();    // A new Number object
var x4 = new Boolean();   // A new Boolean object
var x5 = new Array();     // A new Array object
var x6 = new RegExp();    // A new RegExp object
```

```
var x1 = {};              // new object
var x2 = "";              // new primitive string
var x3 = 0;               // new primitive number
var x4 = false;           // new primitive boolean
var x5 = [];              // new array object
var x6 = /()/;            // new regexp object
var x7 = function(){};    // new function object
```

# JavaScript HTML DOM (Document Object Model)

- Mô hình DOM HTML được xây dựng như một cây của Đối tượng.





# JavaScript HTML DOM

- JS có thể thay đổi tất cả HTML elements, attributes , CSS Style
- JS có thể remove toàn bộ các HTML elements và attributes đang tồn tại.
- JS có thể tác động (thêm mới, sửa,...) tới toàn bộ các evens trong page.

# JavaScript HTML DOM

- **HTML DOM document** là đối tượng sở hữu toàn bộ các đối tượng khác thuộc trang web.
- JS có thể tìm kiếm, thay đổi **HTML Elements**
- ```
document.getElementById("myBtn").addEventListener("click", displayDate);
```

  
(onclick=*JavaScript*)

# JavaScript - HTML DOM Methods

- **HTML DOM methods** là các **actions** có thể thực hiện (trong HTML Elements)
- **HTML DOM properties** là các giá trị (thuộc HTML Elements) có thể thay đổi hoặc thiết lập

```
<html>
```

```
<body>
```

```
<p id="demo"></p>
```

```
<script>
```

```
document.getElementById("demo").innerHTML = "Hello World!";
```

```
</script>
```

```
</body>
```

```
</html>
```

# Browser Object Model

- Browser Object Model cho phép JS gọi tới trình duyệt (Browser).
- Window Screen
- Window Location....

```
var w = window.innerWidth
|| document.documentElement.clientWidth
|| document.body.clientWidth;

var h = window.innerHeight
|| document.documentElement.clientHeight
|| document.body.clientHeight;
```

# JS JSON (JavaScript Object Notation)

# Định nghĩa JSON

- Là một cú pháp để lưu trữ và trao đổi dữ liệu.
- Có thể chuyển đổi 1 JS object thành một chuỗi JSON.
- Có thể chuyển đổi chuỗi JSON thành một object bằng JS.

```
// Chuyển đổi object thành JSON
var myObj = { "name":"John", "age":31, "city":"New York" };
var myJSON = JSON.stringify(myObj);

// Chuyển đổi JSON thành Object
var myJSON = '{ "name":"John", "age":31, "city":"New York" }';
var myObj = JSON.parse(myJSON);
```

# JSON Syntax Rules

- Tên và giá trị theo từng cặp
- Được phân cách bằng dấu phẩy (,)
- Dấu “[” và “]” giữ 1 mảng.
- Dấu “{” và “}” giữ 1 item.

# JSON Values

- Trong JSON giá trị phải là một trong những kiểu sau:
  1. string
  2. number
  3. object
  4. array
  5. boolean
  6. null



# Giống nhau JSON vs XML

- Đều có thể mô tả chính nó
- Đều có thứ bậc.
- Có thể phân tích cú pháp là sử dụng bởi nhiều ngôn ngữ lập trình.
- Đều có thể được nạp với một XMLHttpRequest

# Khác nhau JSON vs XML

- JSON không sử dụng thẻ kết thúc (end tag)\
- JSON ngắn hơn
- JSON đọc và ghi nhanh hơn.
- JSON có thể sử dụng các mảng.

# Tại sao lại sử dụng JSON

- XML khó phân biệt cú pháp hơn JSON
- JSON có thể chuyển đổi thành một JS Object để sử dụng.

# JSON

- JSON Objects
- JSON Arrays
- JSON.parse()
- JSON.stringify()
- JSON HTML.

# Tổng kết

- Nắm được kiến thức về hướng đối tượng trong Javascript
- Sử dụng BOM
- Sử dụng DOM
- Sử dụng JSON Object