



# THIẾT KẾ WEB

BỘ MÔN KỸ THUẬT VÀ MẠNG MÁY TÍNH

Date: 19/08/2020

# Bài 6: Đối tượng trong Javascript

Giới thiệu về đối tượng

01

Cách tạo đối tượng

02

Đối tượng nội tại trong Javascript

03

BOM, DOM HTML

04

# 1



Phần thứ nhất

# GIỚI THIỆU VỀ ĐỐI TƯỢNG

# Giới thiệu về đối tượng

Đối tượng là gì?

- Về mặt định nghĩa, một đối tượng (object) là một danh sách các **thành phần**, mỗi thành phần là một cặp **tên** và **giá trị của nó**, trong đó **giá trị** có thể là: các **kiểu dữ liệu cơ bản**, các **hàm**, hay cũng có thể là một **đối tượng** khác (kiểu dữ liệu phức hợp)

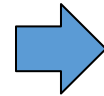
Object	Properties	Methods
	<code>car.name = Fiat</code>	<code>car.start()</code>
	<code>car.model = 500</code>	<code>car.drive()</code>
	<code>car.weight = 850kg</code>	<code>car.brake()</code>
	<code>car.color = white</code>	<code>car.stop()</code>

# Giới thiệu về đối tượng

Truy xuất vào thuộc tính và phương thức của đối tượng

- Truy xuất vào thuộc tính của đối tượng
- Truy xuất vào phương thức của đối tượng

```
objectName.methodName()
```



```
person.lastName;
```

```
name = person.fullName(); person["lastName"];
```



Phần thứ hai

CÁCH TẠO

**ĐỐI TƯỢNG**

2

# Cách tạo đối tượng trong Javascript

## Khởi tạo đối tượng

- Sử dụng khai báo và định nghĩa 1 đối tượng cố định: cách thức này thông qua cặp ngoặc nhọn “{...}”
- Sử dụng từ khóa **new**

```
var person = {firstName:"Joh
```

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

# Cách tạo đối tượng trong Javascript

## Cách tạo đối tượng

- Sử dụng phương thức tạo của đối tượng.
- Thêm phương thức cho đối tượng được thực hiện thông qua phương thức tạo của đối tượng
- Tạo Object prototype bằng cách sử dụng phương thức tạo

```
function Person(first, last, age, eyecolor) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eyecolor;  
}
```

```
Person.prototype.nationality = "English";  
,
```

```
myFather.name = function () {  
    return this.firstName + " " + this.lastName;  
};
```



# Cách tạo đối tượng trong Javascript

Bổ sung thuộc tính và phương thức vào prototype

```
function Person(first, last, age, eyecolor) {  
    this.firstName = first;  
    this.lastName = last;  
    this.age = age;  
    this.eyeColor = eyecolor;  
}  
Person.prototype.name = function() {  
    return this.firstName + " " + this.lastName;  
};
```

- Cách 1: Bổ sung vào phương thức tạo của đối tượng
- Cách 2: Sử dụng thuộc tính prototype

# 3



Phần **thứ ba**

## Đối tượng nội tại trong **Javascript**

# Đối tượng nội tại trong Javascript

## Đối tượng Math

- Đối tượng **Math** là nơi xử lý các vấn đề toán học. Đối tượng này được sử dụng trực tiếp thông qua tên của nó mà không cần phải khởi tạo đối tượng
- Đối tượng Math bao gồm các thuộc tính là các hằng số toán học như PI, LOG2E, SQRT2 ...

```
Math.PI; // returns 3.141592653589793
```

```
Math.E      // returns Euler's number
Math.PI     // returns PI
Math.SQRT2  // returns the square root of 2
Math.SQRT1_2 // returns the square root of 1/2
Math.LN2    // returns the natural logarithm of 2
Math.LN10   // returns the natural logarithm of 10
Math.LOG2E  // returns base 2 logarithm of E
Math.LOG10E // returns base 10 logarithm of E
```

# Đối tượng nội tại trong Javascript

## Đối tượng Math

- Đối tượng **Math** cũng chứa các phương thức là các hàm số toán học mà Javascript đã định nghĩa sẵn như `sin()`, `cos()`, `round()`...

Method	Description
<code>abs(x)</code>	Returns the absolute value of x
<code>acos(x)</code>	Returns the arccosine of x, in radians
<code>acosh(x)</code>	Returns the hyperbolic arccosine of x
<code>asin(x)</code>	Returns the arcsine of x, in radians
<code>asinh(x)</code>	Returns the hyperbolic arcsine of x
<code>atan(x)</code>	Returns the arctangent of x as a numeric value between -PI/2 and PI/2 radians
<code>atan2(y, x)</code>	Returns the arctangent of the quotient of its arguments
<code>atanh(x)</code>	Returns the hyperbolic arctangent of x
<code>cbrt(x)</code>	Returns the cubic root of x
<code>ceil(x)</code>	Returns x, rounded upwards to the nearest integer
<code>cos(x)</code>	Returns the cosine of x (x is in radians)
<code>cosh(x)</code>	Returns the hyperbolic cosine of x

Method	Description
<code>exp(x)</code>	Returns the value of $E^x$
<code>floor(x)</code>	Returns x, rounded downwards to the nearest integer
<code>log(x)</code>	Returns the natural logarithm (base E) of x
<code>max(x, y, z, ..., n)</code>	Returns the number with the highest value
<code>min(x, y, z, ..., n)</code>	Returns the number with the lowest value
<code>pow(x, y)</code>	Returns the value of x to the power of y
<code>random()</code>	Returns a random number between 0 and 1
<code>round(x)</code>	Rounds x to the nearest integer
<code>sin(x)</code>	Returns the sine of x (x is in radians)
<code>sinh(x)</code>	Returns the hyperbolic sine of x
<code>sqrt(x)</code>	Returns the square root of x
<code>tan(x)</code>	Returns the tangent of an angle
<code>tanh(x)</code>	Returns the hyperbolic tangent of a number
<code>trunc(x)</code>	Returns the integer part of a number (x)

# Đối tượng nội tại trong Javascript

## Đối tượng Date

- Đối tượng **Date** là nơi xử lý các vấn đề về thời gian
- Đối tượng Date được khởi tạo ra bởi câu lệnh **new** **var d = new Date();**
- Đối tượng **Date** thường được sử dụng thông qua các phương thức của nó. Có thể kể đến hai nhóm phương thức là **phương thức get** và **phương thức set**

Method	Description
getDate()	Get the day as a number (1-31)
getDay()	Get the weekday as a number (0-6)
getFullYear()	Get the four digit year (yyyy)
getHours()	Get the hour (0-23)
getMilliseconds()	Get the milliseconds (0-999)
getMinutes()	Get the minutes (0-59)
getMonth()	Get the month (0-11)
getSeconds()	Get the seconds (0-59)
getTime()	Get the time (milliseconds since January 1, 1970)

Method	Description
setDate()	Set the day as a number (1-31)
setFullYear()	Set the year (optionally month and day)
setHours()	Set the hour (0-23)
setMilliseconds()	Set the milliseconds (0-999)
setMinutes()	Set the minutes (0-59)
setMonth()	Set the month (0-11)
setSeconds()	Set the seconds (0-59)
setTime()	Set the time (milliseconds since January 1, 1970)

# Đối tượng nội tại trong Javascript

## Đối tượng String

- Đối tượng **String** giúp bạn làm việc với một dãy các ký tự; nó giúp xử lý các kiểu dữ liệu chuỗi gốc trong JavaScript với một số phương thức giúp đỡ
- Để tạo đối tượng **String** ta sử dụng cách **khai báo biến** hoặc qua từ khóa **new**
- Một số thuộc tính và phương thức thông dụng: `charAt()`, `concat()`, `indexOf()`, ....
- Bên cạnh đó các thẻ HTML như `<b>`, `<link>`, `<script>` ... cũng được coi là một thuộc tính của đối tượng **String**

```
var x = "John";  
var y = new String("John");
```



# Đối tượng nội tại trong Javascript

## Đối tượng String

- Thuộc tính **length** trả về độ dài của chuỗi
- Phương thức **indexOf()**, **lastIndexOf()**, **search()** tìm kiếm một chuỗi trong một chuỗi
- Phương thức **slice()**, **substring()**, **substr()** thực hiện trích lấy một phần của chuỗi
- Phương thức **replace()** thực hiện thay thế chuỗi này bằng chuỗi khác

```
str = "Please visit Microsoft!";
var n = str.replace("Microsoft", "W3Schools");
```

```
var str = "Apple, Banana, Kiwi";
var res = str.slice(7, 13);
```

Start  
pos

End  
pos

```
var str = "Please visit Microsoft!";
var pos = str.lastIndexOf("locate");
```

```
var str = "Apple, Banana, Kiwi";
var res = str.substring(7, 13);
```

```
var str = "Apple, Banana, Kiwi";
var res = str.substr(7, 6);
```

# Đối tượng nội tại trong Javascript

## Đối tượng String

```
var text1 = "Hello World!";    // String
var text2 = text1.toUpperCase(); // text2 is text1 converted to upper
```

```
var text1 = "Hello World!";    // String
var text2 = text1.toLowerCase(); // text2 is text1 converted to lower
```

```
var str = "HELLO WORLD";
str.charAt(0);                // returns H
```

```
var str = "HELLO WORLD";
str.charCodeAt(0);            // returns 72
```

- Phương thức **toUpperCase()**, **toLowerCase()** thực hiện chuyển đổi chữ hoa thành chữ thường và ngược lại
- Phương thức **charAt()**, **charCodeAt()** lấy ký tự từ một chuỗi
- Phương thức **split()** thực hiện phân tách chuỗi thành chuỗi con

```
var txt = "a,b,c,d,e";    // String
txt.split(",");           // Split on commas
txt.split(" ");           // Split on spaces
txt.split("|");           // Split on pipe
```





Phần **thứ tư**

# BOM, DOM HTML TRONG **Javascript**

4

# BOM trong Javascript

BOM là gì?

- BOM được viết tắt bởi cụm từ Brower Object Model – Mô hình đối tượng trình duyệt. BOM cho phép Javascript tương tác với trình duyệt web. Để thực hiện được việc này BOM sử dụng đối tượng **window**.
- Tất cả các đối tượng, hàm và biến trong Javascript sẽ tự động trở thành thuộc tính hoặc phương thức của **đối tượng window** như **biến toàn cục là thuộc tính** của window, **hàm toàn cục sẽ là phương thức** của window

```
window.document.getElementById("header");
```

□ Một số **thuộc tính** của đối tượng **window**:

- **window.innerHeight** - chiều cao bên trong của cửa sổ trình duyệt (tính bằng pixel)
- **window.innerWidth** - chiều rộng bên trong của cửa sổ trình duyệt (tính bằng pixel)
- **window.screen** - màn hình của giao diện, qua thuộc tính này có thể truy xuất lấy giá trị về chiều rộng, cao và nhiều thông số khác của màn hình mà khách truy cập (tính bằng pixel)
- **window.location** – xác định địa chỉ trang hiện tại (URL) của trình duyệt, qua thuộc tính này có thể chuyển hướng truy cập hoặc thực hiện một số thao tác khác với URL
- **window.history** – là thuộc tính chứa lịch sử trình duyệt, nó được thực hiện kết hợp với hai phương thức `back()` và `forward()`
- **window.navigator** - tượng chứa thông tin về trình duyệt của khách truy cập

# BOM trong Javascript

Phương thức của đối tượng window

- Một số phương thức của đối tượng window:
  - `window.open()` - mở một cửa sổ mới
  - `window.close()` - đóng cửa sổ hiện tại
  - `window.moveTo()` - di chuyển cửa sổ hiện tại
  - `window.resizeTo()` - thay đổi kích thước cửa sổ hiện tại
  - `window.alert()` – Hộp cảnh báo
  - `window.confirm()` – hộp xác nhận
  - `window.prompt()` – Hộp nhắc nhở

```
alert("I am an alert box!");
```

```
if (confirm("Press a button!")) {  
    txt = "You pressed OK!";  
} else {  
    txt = "You pressed Cancel!";  
}
```

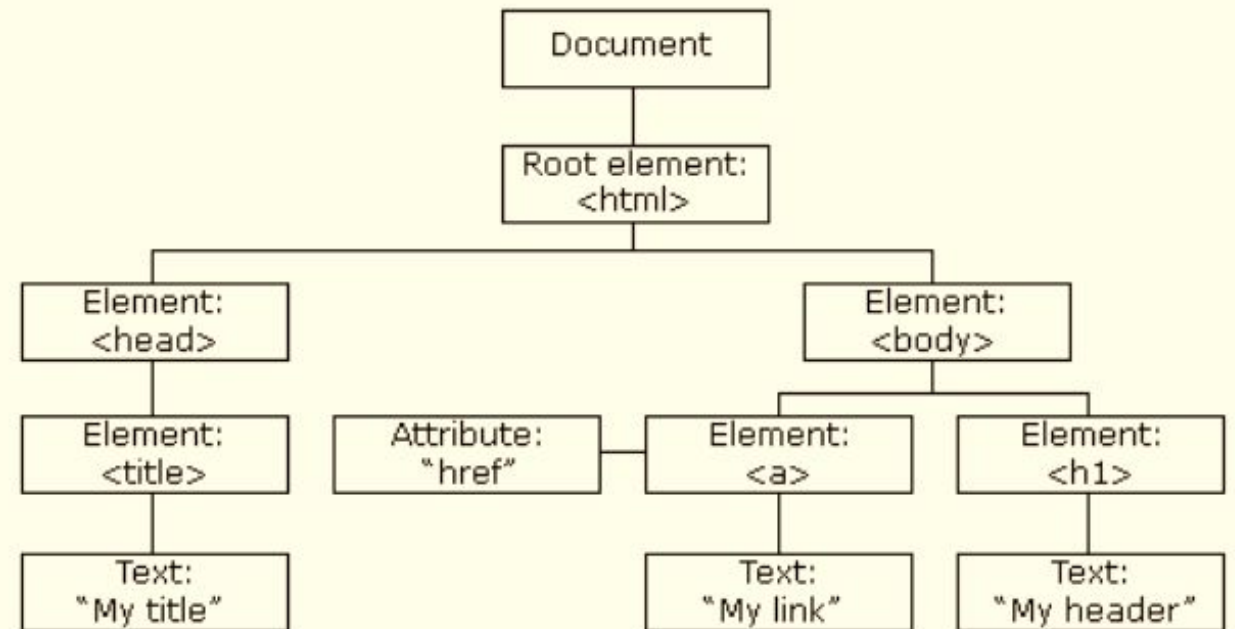
```
var person = prompt("Please enter your name", "Harry Potter");  
  
if (person == null || person == "") {  
    txt = "User cancelled the prompt.";  
} else {  
    txt = "Hello " + person + "! How are you today?";  
}
```

# DOM HTML trong Javascript

DOM HTML là gì?

- Khi một trang web được tải về, trình duyệt sẽ tạo ra một đối tượng DOM (document object model) của trang đó.
- Các DOM HTML được tạo ra như một cây của các đối tượng. HTML DOM là một tiêu chuẩn cho cách lấy, thay đổi, thêm hoặc xóa các phần tử HTML

## Cây đối tượng HTML DOM



# DOM HTML trong Javascript

## Phương thức DOM HTML

- Các phương thức của DOM HTML là các hành động có thể thực hiện trên các phần tử HTML còn thuộc tính của nó là các giá trị của các phần tử đó.
- Các phổ biến nhất để truy cập vào một phần tử HTML là sử dụng id của phần tử thông qua phương thức getElementById()
- Cách lấy nội dung của một phần tử thông qua thuộc tính innerHTML. Thuộc tính này cũng cho phép thay thế nội dung của các phần tử HTML

```
<html>
<body>

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

<script>
document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

# DOM HTML trong Javascript

## Các phần tử DOM HTML

- Thông thường thông qua DOM HTML người lập trình thường thao tác với các phần tử HTML. Để thực hiện có thể qua các phương thức gồm có:
- ✓ `document.getElementById()`: Tìm các phần tử HTML theo id
  - ✓ `document.getElementsByTagName()`: Tìm các phần tử HTML theo tên thẻ
  - ✓ `document.getElementsByClassName()`: Tìm các phần tử HTML theo tên lớp
  - ✓ `document.querySelectorAll()`: Tìm các phần tử HTML theo bộ chọn CSS
  - ✓ Tìm các phần tử HTML theo **tập các đối tượng** được tạo ra bởi cây đối tượng: `document.anchors`, `document.body`, `document.documentElement`, `document.embeds`, `document.forms`, `document.head`, `document.images`, `document.links`, `document.scripts` và tiêu đề tài liệu



# Xin cảm ơn

Date: 8/19/2020