



JS3: Object

FDP 5.0



Nội dung

- ❖ Định nghĩa
- ❖ Thuộc tính
- ❖ Phương thức
- ❖ Phương thức truy cập/thiết lập (Accessors)
- ❖ Phương thức khởi tạo

ĐỊNH NGHĨA

- ❖ Trong JS, mọi thứ là Object
- ❖ Boolean, Number, String có thể là Object (nếu sử dụng toán tử new)
- ❖ Date, Math, Array, ... luôn là Object
- ❖ Kiểu dữ liệu cơ sở: string, number, boolean, null, undefined
- ❖ Giá trị của kiểu dữ liệu cơ sở là không thể thay đổi (immutable)

ĐỊNH NGHĨA

Object – Variable

❖ Biến chứa giá trị đơn

- `var person = "John Doe";`

❖ Object chứa nhiều giá trị

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

❖ Object bao gồm thuộc tính và phương thức

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

ĐỊNH NGHĨA

Tạo Object

❖ Sử dụng Object Literal

❖ Sử dụng từ khóa new

❖ Sử dụng phương thức khởi tạo

- `var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"};`
- `var person = new Object();`
`person.firstName = "John";`
`person.lastName = "Doe";`
`person.age = 50;`
`person.eyeColor = "blue";`

ĐỊNH NGHĨA

Tạo Object

```
<!DOCTYPE html>
<html>
<body>

<p>Creating a JavaScript Object.</p>

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

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

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>

</body>
</html>
```

Creating a JavaScript Object.

John is 50 years old.

ĐỊNH NGHĨA

Tạo Object

```
<!DOCTYPE html>
<html>
<body>

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

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

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>

</body>
</html>
```

John is 50 years old.

ĐỊNH NGHĨA

Tạo Object

❖ Object có thể thay đổi giá trị (mutable)

```
<!DOCTYPE html>
<html>
<body>

<p>JavaScript objects are mutable.</p>
<p>Any changes to a copy of an object will also change the original.</p>

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

<script>
var person = {firstName:"John", lastName:"Doe", age:50, eyeColor:"blue"}

var x = person;
x.age = 10;

document.getElementById("demo").innerHTML =
person.firstName + " is " + person.age + " years old.";
</script>

</body>
</html>
```

JavaScript objects are mutable.

Any changes to a copy of an object will also change the original.

John is 10 years old.

age:50, eyeColor:"blue"

var x = person;
x.age = 10;

THUỘC TÍNH

❖ Truy cập thuộc tính:

- `objectName.property`
- `objectName["property"]`

```
<p>There are two different ways to access an object property:</p>  
<p>You can use .property or ["property"].</p>
```

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

```
<script>  
var person = {  
  firstname:"John",  
  lastname:"Doe",  
  age:50,  
  eyecolor:"blue"  
};
```

```
document.getElementById("demo").innerHTML = person["firstname"] + " is " +  
person["age"] + " years old.";  
</script>
```

There are two different ways to access an object property:

You can use `.property` or `["property"]`.

John is 50 years old.

THUỘC TÍNH

for in

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Object Properties</h2>

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

<script>
var txt = "";
var person = {fname:"John", lname:"Doe", age:25};
var x;
for (x in person) {
  txt += person[x] + " ";
}
document.getElementById("demo").innerHTML = txt;
</script>

</body>
</html>
```

JavaScript Object Properties

John Doe 25

THUỘC TÍNH

Thêm – xóa

- ❖ `person.nationality = "English";`
- ❖ `delete person.age;`

JavaScript Object Properties

You can delete object properties.

John is undefined years old.

```
<!DOCTYPE html>
<html>
<body>

<h2>JavaScript Object Properties</h2>

<p>You can delete object properties.</p>

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

<script>
var person = {
  firstname:"John",
  lastname:"Doe",
  age:50,
  eyecolor:"blue"
};

delete person.age;
document.getElementById("demo").innerHTML =
person.firstname + " is " + person.age + " years old.";
</script>

</body>
</html>
```

PHƯƠNG THỨC

```
<body>

<h2>The JavaScript this Keyword</h2>

<p>In this example, this represents the person object.</p>
<p>Because the person object "owns" the fullName method.</p>

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

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function() {
    return this.firstName + " " + this.lastName;
  }
};

// Display data from the object:
document.getElementById("demo").innerHTML = person.fullName();
</script>

</body>
</html>
```

The JavaScript *this* Keyword

In this example, **this** represents the **person** object.

Because the person object "owns" the fullName method.

John Doe

PHƯƠNG THỨC

this

```
<body>

<h2>The JavaScript <i>this</i> Keyword</h2>

<p>In this example, <b>this</b> represents the <b>person</b> object.</p>
<p>Because the person object "owns" the fullName method.</p>

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

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  id       : 5566,
  fullName : function(firstname) {
    return this.firstName + " " + firstname;
  }
};

// Display data from the object:
document.getElementById("demo").innerHTML = person.fullName("Phước");
</script>

</body>
```

The JavaScript *this* Keyword

In this example, **this** represents the **person** object.

Because the person object "owns" the fullName method.

John Phước

ACCESSORS

get

❖ Bao gồm get và set

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

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  language : "en",
  get lang() {
    return this.language;
  }
};
// Display data from the object using a getter:
document.getElementById("demo").innerHTML = person.lang;
</script>
```

en

ACCESSORS

set

❖ Bao gồm get và set

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

<script>
// Create an object:
var person = {
  firstName: "John",
  lastName : "Doe",
  language : "NO",
  set lang(value) {
    this.language = value;
  }
};
// Set a property using set:
person.lang = "en";
// Display data from the object:
document.getElementById("demo").innerHTML = person.language;
</script>
```

en

PHƯƠNG THỨC KHỞI TẠO

❖ `var person = {fn: "Phước"; ln = "Trần"}; chỉ tạo ra một person`

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

❖ `var myFather = new Person("John", "Doe", 50, "blue");`

❖ `var myMother = new Person("Sally", "Rally", 48, "green");`

❖ -> Tạo nhiều Person

PHƯƠNG THỨC KHỞI TẠO

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

<script>
// Constructor function for Person objects
function Person(first, last, age, eye) {
  this.firstName = first;
  this.lastName = last;
  this.age = age;
  this.eyeColor = eye;
}

// Create 2 Person objects
var myFather = new Person("John", "Doe", 50, "blue");
var myMother = new Person("Sally", "Rally", 48, "green");

// Add nationality to first object
myFather.nationality = "English";

// Display nationality
document.getElementById("demo").innerHTML =
"My father is " + myFather.nationality;
</script>
```

My father is English



BÀI TẬP

❖ Làm bài tập JS03-01

Thank you and
happy learning !!!