



JS2: Các Kiểu Dữ Liệu Nâng Cao Trong JS

FDP 5.0



Nội dung

❖ String

❖ Number

❖ Array

❖ Dates

❖ Math

STRING

Định nghĩa – Ví dụ

❖ Dùng để lưu trữ và xử lý văn bản

- Ví dụ: `var s = "nguyen van teo"`

❖ Có thể sử dụng dấu "" hoặc "

- Ví dụ: `var s = "nguyen van teo"`
- Hoặc `var s1 = 'nguyen van teo'`

❖ Có thể sử dụng dấu " hoặc ' bên trong chuỗi

- Ví dụ:
 - `var answer1 = "It's alright";`
 - `var answer2 = "He is called 'Johnny'";`
 - `var answer3 = 'He is called "Johnny"';`

STRING

Length – escape character

❖ Thuộc tính Length cho biết chiều dài chuỗi

- Ví dụ:
- `var txt = "nguyen van teo";`
`var sln = txt.length;`

❖ Escape Character:

Code	Result	Description
<code>\'</code>	<code>'</code>	Single quote
<code>\"</code>	<code>"</code>	Double quote
<code>\\</code>	<code>\</code>	Backslash

STRING

Phương thức

- ❖ `indexOf(s)`
- ❖ `lastIndexOf(s)`
- ❖ `search(s)`
- ❖ `slice (start, end)`
- ❖ `substring (start, end)`
- ❖ `substr(start, length)`
- ❖ `replace (old, new)`
- ❖ `toLowerCase()` và `toUpperCase()`
- ❖ `concat()`
- ❖ `trim()`
- ❖ `charAt(c)`
- ❖ Property access
- ❖ `split(s)`

NUMBER

Phương thức

❖ parseInt(chuỗi)

```
parseInt("10"); // returns 10
parseInt("10.33"); // returns 10
parseInt("10 20 30"); // returns 10
parseInt("10 years"); // returns 10
parseInt("years 10"); // returns NaN
```

❖ parseFloat(chuỗi)

```
parseFloat("10"); // returns 10
parseFloat("10.33"); // returns 10.33
parseFloat("10 20 30"); // returns 10
parseFloat("10 years"); // returns 10
parseFloat("years 10"); // returns NaN
```

ARRAY

Cú pháp

❖ Dùng để lưu trữ nhiều giá trị trong 1 biến.

❖ Ví dụ: `var cars = ["Saab", "Volvo", "BMW"];`

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

<h2>JavaScript Arrays</h2>

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

<script>
var cars = [
  "Saab",
  "Volvo",
  "BMW"
];
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>
```

JavaScript Arrays

Saab,Volvo,BMW

ARRAY

Cú pháp

❖ `var cars = new Array("Saab", "Volvo", "BMW");`

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

<h2>JavaScript Arrays</h2>

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

<script>
var cars = new Array("Saab", "Volvo", "BMW");
document.getElementById("demo").innerHTML = cars;
</script>

</body>
</html>
```

JavaScript Arrays

Saab, Volvo, BMW

ARRAY

Truy cập – cập nhật phần tử

❖ `var name = cars[0];`

❖ `cars[0] = "Opel";`

❖ Array là **Object**

- `var person = ["John", "Doe", 46];`
- `var person = {firstName:"John", lastName:"Doe", age:46};`

❖ Phương thức và thuộc tính của Array:

- `var x = cars.length; // The length property returns the number of elements`
- `var y = cars.sort(); // The sort() method sorts arrays`

ARRAY

Duyệt các phần tử mảng

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

<h2>JavaScript Arrays</h2>

<p>The best way to loop through an array is using a standard for loop:</p>

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

<script>
var fruits, text, fLen, i;
fruits = ["Banana", "Orange", "Apple", "Mango"];
fLen = fruits.length;

text = "<ul>";
for (i = 0; i < fLen; i++) {
  text += "<li>" + fruits[i] + "</li>";
}
text += "</ul>";

document.getElementById("demo").innerHTML = text;
</script>

</body>
</html>
```

JavaScript Arrays

The best way to loop through an array is using a standard for loop:

- Banana
- Orange
- Apple
- Mango

ARRAY

Thêm phần tử

❖ Cách 1:

- ❖ `var fruits = ["Banana", "Orange", "Apple", "Mango"];`
- ❖ `fruits.push("Lemon");` // adds a new element (Lemon) to fruits

❖ Cách 2:

- ❖ `var fruits = ["Banana", "Orange", "Apple", "Mango"];`
- ❖ `fruits[fruits.length] = "Lemon";` // adds a new element (Lemon) to fruits

❖ toString()

❖ join(chuỗi): tương tự toString()

❖ pop():

❖ var fruits = ["Banana", "Orange", "Apple", "Mango"];

❖ var x = fruits.pop(); // Removes the last element ("Mango")
from fruits

❖ push(chuỗi)

❖ var fruits = ["Banana", "Orange", "Apple", "Mango"];

❖ var x = fruits.push("Kiwi"); // the value of x is 5

❖ shift(): xóa phần tử đầu tiên

❖ var fruits = ["Banana", "Orange", "Apple", "Mango"];

❖ var x = fruits.shift(); // the value of x is "Banana"

❖ unshift(chuỗi): ngược lại với shift()

DATES

- ❖ `var d = new Date();`
- ❖ `var d = new Date(2018, 11, 24, 10, 33, 30, 0);`
- ❖ `getFullYear()`
- ❖ `getMonth()`
- ❖ `getDate()`
- ❖ `getHours()`
- ❖
- ❖ `setDate(số)`
- ❖ `setFullYear(số)`
- ❖

DATES

Ví dụ

JavaScript getFullYear()

The `getFullYear()` method returns the full year of a date:

2020

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

<h2>JavaScript getFullYear()</h2>

<p>The getFullYear() method returns the full year of a date:</p>

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

<script>
var d = new Date();
document.getElementById("demo").innerHTML = d.getFullYear();
</script>

</body>
</html>
```

DATES

Ví dụ

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

<h2>JavaScript setFullYear()</h2>

<p>The setFullYear() method sets the year of a date object:</p>

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

<script>
var d = new Date();
d.setFullYear(2020);
document.getElementById("demo").innerHTML = d;
</script>
</body>
</html>
```

JavaScript setFullYear()

The setFullYear() method sets the year of a date object:

Thu May 28 2020 15:41:17 GMT+0700 (Giờ Đông Dương)

MATH

Phương thức

- ❖ `Math.round(x)`
- ❖ `Math.pow(x,y)`
- ❖ `Math.sqrt(x)`
- ❖ `Math.ceil(x)`
- ❖ `Math.floor(x)`
- ❖ `Math.min (0, 8, 1) ->0`
- ❖ `Math.max (dãy số)`
- ❖ `Math.random(): Trả về giá trị giữa 0 và 1`
- ❖ `Math.floor(Math.random() * 100) + 1; // returns a random integer from 1 to 100`

Thank you and
happy learning !!!