



Trường Trung Cấp Hồng Hà Cần Thơ

# NHẬP MÔN LẬP TRÌNH WEB

*Ths. Nguyễn Minh Triết*



# BÀI 4. NGÔN NGỮ JAVASCRIPT

*Ths. Nguyễn Minh Triết*

1



# Kiểu dữ liệu TRONG JAVASCRIPT

# Kiểu dữ liệu trong JavaScript

- Trong JavaScript, **mỗi giá trị dữ liệu** sẽ thuộc một nhóm **kiểu dữ liệu** nhất định.

```
var a = "Lập Trình Web"; //giá trị của biến a thuộc kiểu dữ liệu string
var b = 1993; //giá trị của biến b thuộc kiểu dữ liệu number
var c = true; //giá trị của biến c thuộc kiểu dữ liệu boolean
var d = false; //giá trị của biến d thuộc kiểu dữ liệu boolean
var e = {name:"Nhân", age:24}; //giá trị của biến e thuộc kiểu dữ liệu object
```

- Việc hiểu rõ bản chất một giá trị thuộc kiểu dữ liệu nào và cách sử dụng kiểu dữ liệu đó ra sao là điều hết sức quan trọng.
- Vì trong JavaScript, ta phải thường xuyên thực hiện những biểu thức giữa các giá trị, việc nhầm lẫn kiểu dữ liệu sẽ khiến kết quả không như mong đợi.

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## ● Ví dụ:

### - Trong đoạn mã bên dưới:

- Biến a có giá trị là số 22 (*thuộc kiểu number*)
- Biến b có giá trị là chuỗi 22 (*thuộc kiểu string*)

=> Nếu không phân biệt rõ được sự khác nhau giữa hai kiểu dữ liệu này thì ta sẽ nghĩ biến a cộng b sẽ cho ra kết quả là số 44

```
<script>
  var a = 22;
  var b = "22";
  var c = a + b;
  document.write(c);
</script>
```



2222

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

- Trong JavaScript, các **kiểu dữ liệu** được chia thành những **loại cơ bản** như sau:
  - string
  - number
  - boolean
  - object
  - undefined
  - array (*đây là một trường hợp đặc biệt của kiểu dữ liệu object*)

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 1. Dữ liệu kiểu **string**

- Trong JavaScript, các dữ liệu thuộc kiểu **string** (hay còn được gọi là "**chuỗi**") là một tập hợp gồm các ký tự, chúng được viết bên trong cặp dấu nháy kép hoặc cặp dấu nháy đơn.

```
<p id="demo1"></p>

<p id="demo2"></p>

<script>
  var a = "Tài liệu học HTML";
  var b = 'Lập Trình Web';
  document.getElementById("demo1").innerHTML = a;
  document.getElementById("demo2").innerHTML = b;
</script>
```

Tài liệu học HTML

Lập Trình Web

- Giá trị của biến a là một chuỗi được viết bên trong cặp dấu nháy kép.
- Giá trị của biến b là một chuỗi được viết bên trong cặp dấu nháy đơn.

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 2. Dữ liệu kiểu **number**

- Trong JavaScript, các dữ liệu thuộc **kiểu number** (hay còn được gọi là "**số**") là một tập hợp của các con số (0 - 9) không chứa dấu khoảng trắng và có thể chứa dấu trừ (-) nằm ở đầu để đại diện cho số âm.

```
<p id="demo1"></p>
<p id="demo2"></p>
<script>
  var a = 22;
  var b = -1993;
  document.getElementById("demo1").innerHTML = a;
  document.getElementById("demo2").innerHTML = b;
</script>
```

22

-1993



# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 2. Dữ liệu kiểu **number**

- Nếu một số được đặt bên trong cặp dấu nháy kép hoặc cặp dấu nháy đơn thì nó sẽ bị chuyển sang kiểu dữ liệu string(chuỗi)

```
<script>  
  var a = "22";  
  var b = '-1993';  
</script>
```

22

-1993

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 3. Dữ liệu kiểu **boolean**

- Trong JavaScript, các dữ liệu thuộc kiểu boolean chỉ có thể nhận một trong hai giá trị, đó là:

- true (*đúng*)
- false (*sai*)

- Có hai cách để nhận giá trị kiểu boolean, đó là:

- Gán giá trị trực tiếp
- Nhận được từ một điều kiện

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 3. Dữ liệu kiểu **boolean**

```
<p id="demo1"></p>

<p id="demo2"></p>

<p id="demo3"></p>

<p id="demo4"></p>

<script>
  var a = true;
  var b = false;
  var c = 6 > 2;
  var d = 6 > 10;
  document.getElementById("demo1").innerHTML = a;
  document.getElementById("demo2").innerHTML = b;
  document.getElementById("demo3").innerHTML = c;
  document.getElementById("demo4").innerHTML = d;
</script>
```

true

false

true

false

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 4. Dữ liệu kiểu **object**

- Trong JavaScript, các dữ liệu thuộc kiểu **object** (hay còn được gọi là "*đối tượng*") là một tập hợp gồm những cái tên và mỗi cái tên sẽ chứa đựng một giá trị dữ liệu.
- Lưu ý: Những cái tên còn được gọi là "*thuộc tính*" của đối tượng, giá trị của những cái tên còn được gọi là "*giá trị thuộc tính của đối tượng*".

```
<script>
  var SinhVien = {
    name:"Nhân",
    gender:"Nam",
    year:1993
  }
  document.write("Giá trị của thuộc tính name là: " + SinhVien.name);
  document.write("<hr>");
  document.write("Giá trị của thuộc tính gender là: " + SinhVien.gender);
  document.write("<hr>");
  document.write("Giá trị của thuộc tính year là: " + SinhVien.year);
</script>
```

Giá trị của thuộc tính name là: Nhân

Giá trị của thuộc tính gender là: Nam

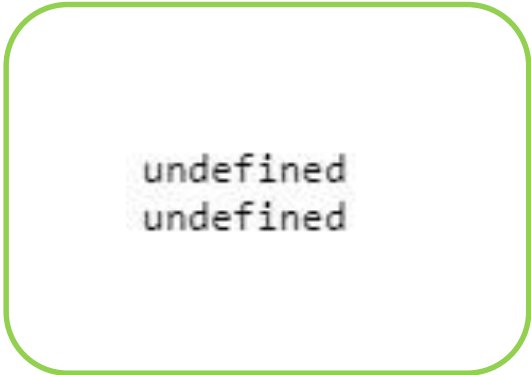
Giá trị của thuộc tính year là: 1993

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 5. Dữ liệu kiểu **undefined**

- Trong JavaScript, khi một biến được khai báo mà không gán giá trị thì biến đó sẽ có **giá trị** là **undefined** và kiểu dữ liệu cũng là **undefined**.

```
<pre>  
<script>  
    var myName;  
    document.writeln(myName);  
  
    document.writeln(typeof(myName));  
  
</script>  
</pre>
```



undefined  
undefined

# KIỂU DỮ LIỆU TRONG JAVASCRIPT

## 6. Dữ liệu kiểu **array**

- Trong JavaScript, array còn được gọi là **mảng**, nó là một trường hợp đặc biệt của đối tượng. *(Thật ra, mảng có kiểu dữ liệu là object)*
- Mảng là **một loại biến đặc biệt có thể lưu trữ nhiều giá trị đồng thời**, mỗi giá trị được gọi là một phần tử mảng.

```
<script>
  var mobile = ["HTC", "Nokia", "SamSung"];
  document.write(mobile[0] + "<hr>");
  document.write(mobile[1] + "<hr>");
  document.write(mobile[2] + "<hr>");
</script>
```

HTC

---

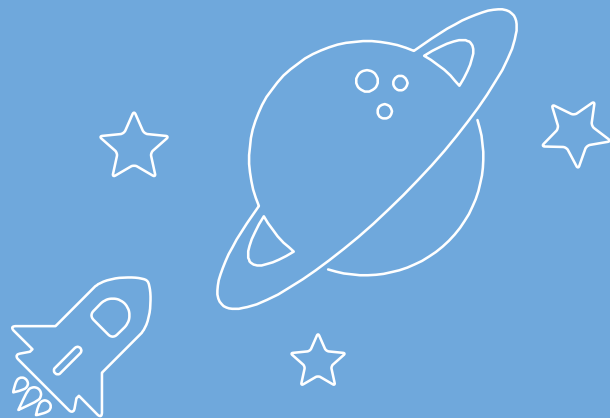
Nokia

---

SamSung

---

# 2



# PHƯƠNG THỨC XỬ LÝ CHUỖI TRONG JAVASCRIPT

# PHƯƠNG THỨC XỬ LÝ CHUỖI

- Chuỗi (hay còn được gọi là chuỗi ký tự) là một dãy các ký tự, ví dụ:

Tai	Chuỗi có 3 ký tự
Tai lieu	Chuỗi có 8 ký tự
Tai lieu hoc	Chuỗi có 12 ký tự
Tai lieu hoc JavaScript	Chuỗi có 23 ký tự

- Một chuỗi sẽ có các phương thức và thuộc tính, với việc truy cập vào các phương thức và thuộc tính của chuỗi, ta có thể làm được những công việc như: *đếm số lượng ký tự của chuỗi, chuyển toàn bộ ký tự của chuỗi về dạng chữ in hoa, chuyển toàn bộ ký tự của chuỗi về dạng chữ thường, trích xuất một chuỗi con bên trong chuỗi, ....*



# PHƯƠNG THỨC XỬ LÝ CHUỖI

- Tuy nhiên, để sử dụng được các phương thức và thuộc tính này thì trước hết cần phải hiểu rõ khái niệm **chỉ số** của ký tự trong chuỗi
- Trong một chuỗi, các ký tự được sắp xếp theo thứ tự vị trí tăng dần từ 1 đến hết, ví dụ:

Chuỗi	T	À	I		L	I	Ệ	U		H	Ọ	C		H	T	M	L
Vị trí	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

- Nhưng khi làm việc với các phương thức thì không dùng đến vị trí mà là dùng đến **chỉ số** (*chỉ số của ký tự luôn có giá trị nhỏ hơn vị trí của ký tự một đơn vị*), ví dụ:

Chuỗi	T	À	I		L	I	Ệ	U		H	Ọ	C		H	T	M	L
Chỉ số	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Vị trí	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17

# PHƯƠNG THỨC XỬ LÝ CHUỖI

- Một số phương thức và thuộc tính thường được sử dụng nhất trong việc xử lý chuỗi:

Phương thức / Thuộc tính	Mô tả chức năng
length	Trả về số lượng ký tự của chuỗi
toUpperCase()	Chuyển toàn bộ ký tự của chuỗi về dạng chữ in hoa
toLowerCase()	Chuyển toàn bộ ký tự của chuỗi về dạng chữ thường
concat()	Nối các chuỗi lại với nhau
indexOf()	Trả về chỉ số của chuỗi trùng khớp được tìm thấy đầu tiên <i>(Khi bạn muốn tìm một chuỗi con nào đó trong chuỗi)</i>
lastIndexOf()	Trả về chỉ số của chuỗi trùng khớp được tìm thấy sau cùng <i>(Khi bạn muốn tìm một chuỗi con nào đó trong chuỗi)</i>
charAt()	Trích xuất <u>một ký tự</u> trong chuỗi
substring()	Trích xuất <u>một chuỗi con</u> trong chuỗi
substr()	Trích xuất <u>một chuỗi con</u> trong chuỗi
replace()	Thay thế một nội dung nào đó trong chuỗi bằng nội dung mới

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 1. Đếm số lượng ký tự của chuỗi

- Để đếm số lượng ký tự của một chuỗi, ta truy cập vào thuộc tính **length** của chuỗi đó.

```
<script>
  var text = "Tai lieu hoc JavaScript";
  var a = text.length;
  document.write(a);
</script>

<hr>

<script>
  var b = "JavaScript".length;
  document.write(b);
</script>
```

23

10

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 2. Chuyển chuỗi về dạng chữ in hoa

- Để chuyển toàn bộ ký tự của một chuỗi về dạng chữ in hoa, ta truy cập vào phương thức **toUpperCase()** của chuỗi đó.

```
<script>
  var text = "Tai lieu hoc JavaScript";
  var a = text.toUpperCase();
  document.write(a);
</script>
```

```
<hr>
```

```
<script>
  var b = "JavaScript".toUpperCase();
  document.write(b);
</script>
```

TAI LIEU HOC JAVASCRIPT

JAVASCRIPT

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 3. Chuyển chuỗi về dạng chữ in thường

- Để chuyển toàn bộ ký tự của một chuỗi về dạng chữ thường, ta truy cập vào phương thức **toLowerCase()** của chuỗi đó.

```
<script>
  var text = "Tai lieu hoc JavaScript";
  var a = text.toLowerCase();
  document.write(a);
</script>

<hr>

<script>
  var b = "JavaScript".toLowerCase();
  document.write(b);
</script>
```

tai lieu hoc javascript

javascript

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 4. Nối các chuỗi lại với nhau

- Thay vì sử dụng toán tử + để nối các chuỗi lại với nhau thì ta cũng có thể sử dụng phương thức **concat()** với cú pháp như sau để nối chuỗi:

```
<div id="demo1"></div>
<div id="demo2"></div>
<div id="demo3"></div>

<script>
  var a = "Lập ";
  var b = "trình";
  var c = " web";

  var str1 = a.concat(b, c);
  var str2 = a.concat(b, " di động");
  var str3 = "Java".concat("Script");

  document.getElementById("demo1").innerHTML = str1;
  document.getElementById("demo2").innerHTML = str2;
  document.getElementById("demo3").innerHTML = str3;
</script>
```

Lập trình web  
Lập trình di động  
JavaScript

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 5. Xác định chỉ số của một chuỗi con

- Ta có thể xác định được chỉ số của một chuỗi con nào đó nằm bên trong chuỗi bằng cách sử dụng phương thức **indexOf()** hoặc **lastIndexOf()**

*string.indexOf(chuỗi mà bạn xác định chỉ số của nó trong chuỗi string)*

*string.lastIndexOf(chuỗi mà bạn xác định chỉ số của nó trong chuỗi string)*

- Phương thức **indexOf()** sẽ trả về chỉ số của chuỗi trùng khớp **được tìm thấy đầu tiên**.
- Phương thức **lastIndexOf()** sẽ trả về chỉ số của chuỗi trùng khớp **được tìm thấy cuối cùng**.

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 5. Xác định chỉ số của một chuỗi con

```
<div id="demo1"></div>
<div id="demo2"></div>

<script>
  var text = "Tài liệu học HTML và tài liệu học CSS";
  var a = text.indexOf("học");
  var b = text.lastIndexOf("học");

  document.getElementById("demo1").innerHTML = a;
  document.getElementById("demo2").innerHTML = b;
</script>
```

9  
30



# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 6. Trích xuất một ký tự trong chuỗi

- Ta có thể trích xuất một ký tự nào đó trong chuỗi bằng cách sử dụng phương thức **charAt()** với cú pháp như sau:

```
<script>
  var text = "TÀI LIỆU HỌC HTML";
  var a = text.charAt(4);
  document.write(a);
</script>
```



L

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 7. Trích xuất một chuỗi con

- Ta có thể trích xuất được một chuỗi con bên trong chuỗi bằng phương thức **substring()** hoặc **substr()**

### 7.1. Phương thức **substring()**

*string.substring(start, end)*

- start là chỉ số của ký tự bắt đầu được trích xuất
- end là chỉ số của ký tự kết thúc việc trích xuất

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 7.1. Phương thức **substring()**

```
<script>  
  var text = "TÀI LIỆU HỌC HTML";  
  var a = text.substring(4, 14);  
  document.write(a);  
</script>
```

LIỆU HỌC H

Chỉ số	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Chuỗi	T	À	I		L	I	Ệ	U		H	Ọ	C		H	T	M	L

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 7.2. Phương thức **substr()**

*string.substr(start, length)*

- start là chỉ số của ký tự bắt đầu được trích xuất
- length là số lượng ký tự được trích xuất

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 7.2. Phương thức **substr()**

```
<script>  
  var text = "TÀI LIỆU HỌC HTML";  
  var a = text.substr(4, 3);  
  document.write(a);  
</script>
```

LIỆ

Chỉ số	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Chuỗi	T	À	I		L	I	Ệ	U		H	Ọ	C		H	T	M	L

# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 8. Thay thế nội dung trong chuỗi

- Để thay thế một nội dung nào đó trong chuỗi bằng nội dung mới, ta sử dụng phương thức **replace()** với cú pháp như sau:

*string.replace(nội dung cũ, nội dung mới)*

Nội dung ở đây có thể là một ký tự hoặc một chuỗi ký tự.

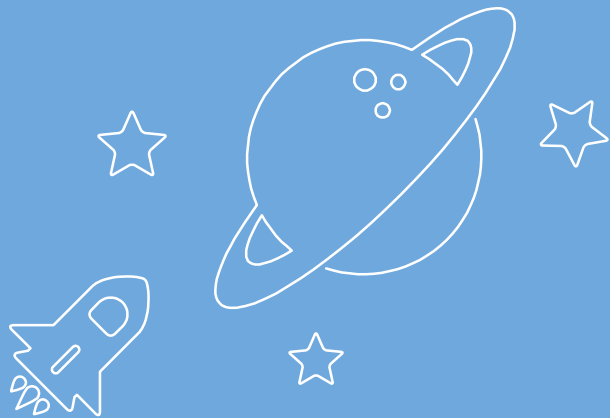
# PHƯƠNG THỨC XỬ LÝ CHUỖI

## 8. Thay thế nội dung trong chuỗi

```
<script>  
    var text = "Tài liệu học HTML";  
    var a = text.replace("học", "tham khảo");  
    document.write(a);  
</script>
```

Tài liệu tham khảo HTML

# 3



# ĐỐI TƯỢNG TOÁN HỌC MATH TRONG JAVASCRIPT



# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

- Trong JavaScript có một đối tượng được xây dựng sẵn tên là **Math**. **Đối tượng Math** sẽ có **các phương thức**, với việc truy cập vào các phương thức đó, ta có thể thực hiện được những công việc toán học trên các con số, ví dụ như: *lấy giá trị tuyệt đối của một số, tính căn bậc hai của một số, tìm số lớn nhất, tìm số nhỏ nhất, làm tròn số, ....*
- Dưới đây là danh sách các phương thức thường được sử dụng nhất của đối tượng Math:

Phương thức	Mô tả chức năng
abs()	Trả về giá trị tuyệt đối của một số
random()	Trả về một số ngẫu nhiên trong khoảng từ 0 đến 1
sqrt()	Trả về căn bậc hai của một số
pow()	Trả về lũy thừa của hai số xác định
ceil()	Làm tròn một số lên số nguyên gần nó nhất
floor()	Làm tròn một số xuống số nguyên gần nó nhất
round()	Làm tròn một số đến số nguyên gần nó nhất
max()	Trả về số lớn nhất trong danh sách
min()	Trả về số nhỏ nhất trong danh sách

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 1. Phương thức **abs()**

- Phương thức **abs()** dùng để trả về giá trị tuyệt đối của một số.
- Lưu ý, **phương thức abs()** sẽ:
  - Trả về 0 nếu giá trị ban đầu là null
  - Trả về NaN nếu giá trị ban đầu không phải là một số

```
<script>
  var a = Math.abs(8.75);
  var b = Math.abs(-8.75);
  var c = Math.abs(null);
  var d = Math.abs("JavaScript");
  var e = Math.abs(-7+3);

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
  document.write(d + "<br>");
  document.write(e + "<br>");
</script>
```

8.75

8.75

0

NaN

4

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 2. Phương thức **random()**

- Phương thức **random()** dùng để trả về một số ngẫu nhiên trong khoảng từ 0 đến 1

```
<script>  
  var a = Math.random();  
  document.write(a + "<br>");  
</script>
```

0.344113097538977

- Lưu ý: Phương thức **random()** có thể trả về số 0, nhưng không bao giờ trả về số 1

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 3. Phương thức `sqrt()`

- Phương thức `sqrt()` dùng để trả về **giá trị căn bậc hai** của một số.
- Lưu ý: Phương thức `sqrt()` sẽ trả về NaN nếu giá trị ban đầu là số âm.

```
<script>
  var a = Math.sqrt(0);
  var b = Math.sqrt(1);
  var c = Math.sqrt(4);
  var d = Math.sqrt(81);
  var e = Math.sqrt(49);
  var f = Math.sqrt(-4);

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
  document.write(d + "<br>");
  document.write(e + "<br>");
  document.write(f + "<br>");
</script>
```

0  
1  
2  
9  
7  
NaN

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 4. Phương thức **pow()**

- Phương thức **pow()** dùng để trả về lũy thừa của hai số xác định.
- Cú pháp:

`Math.pow(x,y);` //tương đương với  $x^y$  trong đại số

```
<script>
  var a = Math.pow(5,2);
  document.write(a);
</script>
```

25

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 5. Phương thức **ceil()**

- Phương thức **ceil()** dùng để làm tròn một số lên số nguyên gần nó nhất.
- Lưu ý: Nếu giá trị ban đầu là một số nguyên thì nó sẽ không được làm tròn.

```
<script>
  var a = Math.ceil(3.60);
  var b = Math.ceil(3.40);
  var c = Math.ceil(3);
  var d = Math.ceil(-3.60);
  var e = Math.ceil(-3.40);
  var f = Math.ceil(-3);

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
  document.write(d + "<br>");
  document.write(e + "<br>");
  document.write(f + "<br>");
</script>
```



4  
4  
3  
-3  
-3  
-3

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 6. Phương thức **floor()**

- Phương thức floor() dùng để làm tròn một số xuống số nguyên gần nó nhất.
- Lưu ý: Nếu giá trị ban đầu là một số nguyên thì nó sẽ không được làm tròn.

```
<script>
  var a = Math.floor(3.60);
  var b = Math.floor(3.40);
  var c = Math.floor(3);
  var d = Math.floor(-3.60);
  var e = Math.floor(-3.40);
  var f = Math.floor(-3);

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
  document.write(d + "<br>");
  document.write(e + "<br>");
  document.write(f + "<br>");
</script>
```



3  
3  
3  
-4  
-4  
-3

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 7. Phương thức **round()**

- Phương thức **round()** dùng để làm tròn một số đến số nguyên gần nó nhất.
- Lưu ý: Nếu giá trị ban đầu là một số nguyên thì nó sẽ không được làm tròn.

```
<script>
  var a = Math.round(3.60);
  var b = Math.round(3.50);
  var c = Math.round(3.49);
  var d = Math.round(-3.60);
  var e = Math.round(-3.50);
  var f = Math.round(-3.49);

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
  document.write(d + "<br>");
  document.write(e + "<br>");
  document.write(f + "<br>");
</script>
```



4  
4  
3  
-4  
-3  
-3



# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 8. Phương thức **max()**

- Phương thức `max()` dùng để **trả về số lớn nhất trong danh sách** các đối số.
- Cú pháp:

`Math.max(đối số thứ nhất, đối số thứ hai, đối số thứ ba, ..., đối số thứ n);`

- Lưu ý, phương thức `max()` sẽ:
  - Trả về NaN nếu trong các đối số của phương thức `max()` có tồn tại đối số không phải là số.
  - Trả về Infinity nếu phương thức `max()` không có đối số.

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 8. Phương thức **max()**

```
<script>
  var a = Math.max(0, 16, -3, 5.25, 100, -1);
  var b = Math.max(0, "JavaScript", -3, 5.25, 100, -1);
  var c = Math.max();

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
</script>
```

100  
NaN  
-Infinity

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 9. Phương thức **min()**

- Phương thức min() dùng để **trả về số nhỏ nhất trong danh sách** các đối số.
- Cú pháp:

`Math.min(đối số thứ nhất, đối số thứ hai, đối số thứ ba, ..., đối số thứ n);`

- Lưu ý, phương thức min() sẽ:
  - Trả về NaN nếu trong các đối số của phương thức min() có tồn tại đối số không phải là số.
  - Trả về Infinity nếu phương thức min() không có đối số.

# ĐỐI TƯỢNG MATH TRONG JAVASCRIPT

## 9. Phương thức **min()**

```
<script>
  var a = Math.min(0, 16, -3, 5.25, 100, -1);
  var b = Math.min(0, "JavaScript", -3, 5.25, 100, -1);
  var c = Math.min();

  document.write(a + "<br>");
  document.write(b + "<br>");
  document.write(c + "<br>");
</script>
```

-3  
NaN  
Infinity

# 4



# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG TRONG JAVASCRIPT

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

- Đối tượng ngày tháng là một tập hợp các thông tin về thời gian (*thứ, ngày, tháng, năm, giờ, phút, giây, mili giây, múi giờ*) của một thời điểm nào đó.
- Các phương thức xử lý ngày tháng trong JavaScript giúp chúng ta có thể làm một số công việc với đối tượng ngày tháng như:
  - *Lấy các thông tin về thời gian của đối tượng ngày tháng: ngày, tháng, năm, giờ, phút, giây, ....*
  - *Thiết lập lại các thông tin về thời gian của đối tượng ngày tháng*

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

- Dưới đây là danh sách các phương thức dùng để lấy thông tin về thời gian của đối tượng ngày tháng theo giờ địa phương:

Phương thức	Mô tả
getDay()	Trả về ngày trong tuần ( <i>Thứ</i> ) của đối tượng ngày tháng (có giá trị từ 0 - 6)
getDate()	Trả về ngày trong tháng của đối tượng ngày tháng (có giá trị từ 1 - 31)
getMonth()	Trả về tháng của đối tượng ngày tháng (có giá trị từ 0 - 11)
getFullYear()	Trả về năm của đối tượng ngày tháng (có giá trị từ 1000 - 9999)
getHours()	Trả về giờ của đối tượng ngày tháng (có giá trị từ 0 - 23)
getMinutes()	Trả về phút của đối tượng ngày tháng (có giá trị từ 0 - 59)
getSeconds()	Trả về giây của đối tượng ngày tháng (có giá trị từ 0 - 59)
getMilliseconds()	Trả về mili giây của đối tượng ngày tháng (có giá trị từ 0 - 999)
getTime()	Trả về tổng số mili giây từ thời điểm 01/01/1970 00:00:00 đến thời điểm của đối tượng ngày tháng (tính theo giờ tiêu chuẩn UTC)

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

- Dưới đây là danh sách các phương thức dùng để thiết lập lại thông tin về thời gian của đối tượng ngày tháng theo giờ địa phương:

Phương thức	Mô tả
setDate()	Thiết lập lại ngày trong tháng của đối tượng ngày tháng
setMonth()	Thiết lập lại tháng của đối tượng ngày tháng
setFullYear()	Thiết lập lại năm của đối tượng ngày tháng
setHours()	Thiết lập lại giờ của đối tượng ngày tháng
setMinutes()	Thiết lập lại phút của đối tượng ngày tháng
setSeconds()	Thiết lập lại giây của đối tượng ngày tháng
setMilliseconds()	Thiết lập lại mili giây của đối tượng ngày tháng



# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 1. Phương thức **getDay()**

- Phương thức **getDay()** trả về ngày trong tuần (*hay còn gọi là Thứ*) của đối tượng ngày tháng.
- Kết quả trả về sẽ có giá trị nằm trong khoảng từ 0 đến 6:
  - Chủ Nhật có giá trị 0
  - Thứ hai có giá trị 1
  - Thứ ba có giá trị 2
  - Thứ tư có giá trị 3
  - Thứ năm có giá trị 4
  - Thứ sáu có giá trị 5
  - Thứ bảy có giá trị 6

```
<script>  
    var date = new Date();  
    var result = date.getDay();  
    document.write(result);  
</script>
```

5

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 2. Phương thức **getDate()**

- Phương thức **getDate()** trả về ngày trong tháng của đối tượng ngày tháng.
- Kết quả trả về sẽ có giá trị nằm trong khoảng từ 1 đến 31.

```
<script>
  var date = new Date();
  var result = date.getDate();
  document.write(result);
</script>
```



7

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 3. Phương thức **getMonth()**

- Phương thức **getMonth()** trả về tháng của đối tượng ngày tháng.
- Kết quả trả về sẽ có giá trị nằm trong khoảng từ 0 đến 11:
  - Tháng một có giá trị 0
  - Tháng hai có giá trị 1
  - Tháng ba có giá trị 2
  - ....
  - Thứ mười hai có giá trị 11

```
<script>
  var date = new Date();
  var result = date.getMonth();
  document.write(result);
</script>
```

6

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 4. Phương thức **getFullYear()**

- Phương thức **getFullYear()** trả về năm của đối tượng ngày tháng.
- Kết quả trả về sẽ có giá trị nằm trong khoảng từ 1000 đến 9999.

```
<script>  
    var date = new Date();  
    var result = date.getFullYear();  
    document.write(result);  
</script>
```

2023

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 5. Phương thức **getHours()**

- Phương thức **getHours()** trả về giờ của đối tượng ngày tháng.
- Kết quả trả về sẽ có giá trị nằm trong khoảng từ 0 đến 23.

```
<script>  
    var date = new Date();  
    var result = date.getHours();  
    document.write(result);  
</script>
```

16

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 6. Phương thức **getSeconds()**

- Phương thức **getSeconds()** trả về giây của đối tượng ngày tháng.
- Kết quả trả về sẽ có giá trị nằm trong khoảng từ 0 đến 59.

```
<script>
  var date = new Date();
  var result = date.getSeconds();
  document.write(result);
</script>
```

51

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 7. Phương thức **setDate()**

- Phương thức **setDate()** dùng để thiết lập lại ngày trong tháng của đối tượng ngày tháng.
  - Muốn thiết lập lại thành ngày 1, nhập đối số có là giá trị 1
  - Muốn thiết lập lại thành ngày 2, nhập đối số có là giá trị 2
  - Muốn thiết lập lại thành ngày 3, nhập đối số có là giá trị 3
  - ....
  - Muốn thiết lập lại thành ngày 31, nhập đối số có là giá trị 31

```
<script>

var date = new Date();

document.write("Thời điểm ban đầu: " + date + "<br>");

date.setDate(15);

document.write("Sau khi thiết lập lại: " + date);

</script>
```

Thời điểm ban đầu: Fri Jul 07 2023 16:23:07 GMT+0700 (GMT+07:00)  
Sau khi thiết lập lại: Sat Jul 15 2023 16:23:07 GMT+0700 (GMT+07:00)

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 8. Phương thức **setMonth()**

- Phương thức **setMonth()** dùng để thiết lập lại tháng của đối tượng ngày tháng.
  - Muốn thiết lập lại thành tháng một, nhập đối số có là giá trị 0
  - Muốn thiết lập lại thành tháng hai, nhập đối số có là giá trị 1
  - Muốn thiết lập lại thành tháng ba, nhập đối số có là giá trị 2
  - ....
  - Muốn thiết lập lại thành tháng mười hai, nhập đối số có là giá trị 11

```
<script>

var date = new Date();

document.write("Thời điểm ban đầu: " + date + "<br>");

date.setMonth(7);

document.write("Sau khi thiết lập lại: " + date);

</script>
```

Thời điểm ban đầu: Fri Jul 07 2023 16:25:23 GMT+0700 (GMT+07:00)  
Sau khi thiết lập lại: Mon Aug 07 2023 16:25:23 GMT+0700 (GMT+07:00)



# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 9. Phương thức **setFullYear()**

- Phương thức **setFullYear()** dùng để thiết lập lại năm của đối tượng ngày tháng.

```
<script>

var date = new Date();

document.write("Thời điểm ban đầu: " + date + "<br>");

date.setFullYear(1993);

document.write("Sau khi thiết lập lại: " + date);

</script>
```

Thời điểm ban đầu: Fri Jul 07 2023 16:27:57 GMT+0700 (GMT+07:00)  
Sau khi thiết lập lại: Wed Jul 07 1993 16:27:57 GMT+0700 (GMT+07:00)

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 10. Phương thức **setHours()**

- Phương thức **setHours()** dùng để thiết lập lại giờ của đối tượng ngày tháng.
  - Muốn thiết lập lại thành 1 giờ, nhập đối số có là giá trị 1
  - Muốn thiết lập lại thành 2 giờ, nhập đối số có là giá trị 2
  - Muốn thiết lập lại thành 3 giờ, nhập đối số có là giá trị 3
  - ...
  - Muốn thiết lập lại thành 24 giờ, nhập đối số có là giá trị 0

```
<script>
```

```
var date = new Date();
```

```
document.write("Thời điểm ban đầu: " + date + "<br>");
```

```
date.setHours(17);
```

```
document.write("Sau khi thiết lập lại: " + date);
```

```
</script>
```

Thời điểm ban đầu: Fri Jul 07 2023 16:29:54 GMT+0700 (GMT+07:00)  
Sau khi thiết lập lại: Fri Jul 07 2023 17:29:54 GMT+0700 (GMT+07:00)

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 11. Phương thức **setMinutes()**

- Phương thức **setMinutes()** dùng để thiết lập lại phút của đối tượng ngày tháng.

```
<script>
var date = new Date();

document.write("Thời điểm ban đầu: " + date + "<br>");

date.setMinutes(35);

document.write("Sau khi thiết lập lại: " + date);
</script>
```

Thời điểm ban đầu: Fri Jul 07 2023 16:31:43 GMT+0700 (GMT+07:00)  
Sau khi thiết lập lại: Fri Jul 07 2023 16:35:43 GMT+0700 (GMT+07:00)

# PHƯƠNG THỨC XỬ LÝ NGÀY THÁNG

## 12. Phương thức **setSeconds()**

- Phương thức **setSeconds()** dùng để thiết lập lại giây của đối tượng ngày tháng.

```
<script>
    var date = new Date();

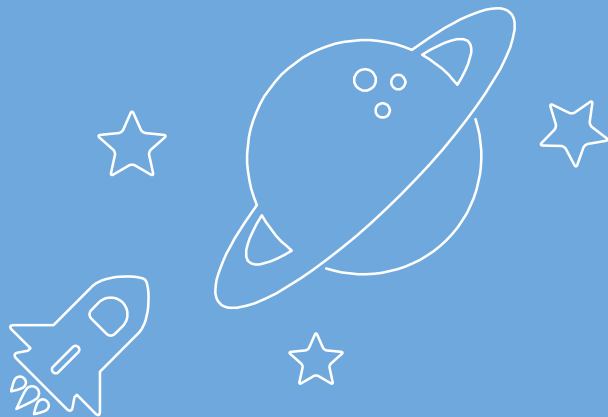
    document.write("Thời điểm ban đầu: " + date + "<br>");

    date.setSeconds(59);

    document.write("Sau khi thiết lập lại: " + date);
</script>
```

Thời điểm ban đầu: Fri Jul 07 2023 16:34:09 GMT+0700 (GMT+07:00)  
Sau khi thiết lập lại: Fri Jul 07 2023 16:34:59 GMT+0700 (GMT+07:00)

# 5



# CÁCH BẮT SỰ KIỆN (EVENT) TRONG JAVASCRIPT

# SỰ KIỆN TRONG JAVASCRIPT

## 1. Sự kiện là gì ?

- Thông thường, sự kiện là những hành động của người dùng khi tương tác lên phần tử HTML.

- Khi người dùng click chuột vào phần tử, đó là một sự kiện.

=> Ví dụ, bạn hãy thử

- Khi người dùng gõ văn bản vào textfield, đó là một sự kiện.

=> Ví dụ, bạn hãy thử

- Khi người dùng di chuyển con trỏ vào phần tử, đó là một sự kiện.

=> Ví dụ, bạn hãy thử 

Di chuyển con trỏ vào đây

- ....

# SỰ KIỆN TRONG JAVASCRIPT

## 2. Các sự kiện trong Javascript

- Trong JavaScript, mỗi sự kiện sẽ tương ứng với một cái tên.
- Dưới đây là danh sách một vài sự kiện phổ biến mà ta thường dùng trong JavaScript.

Tên sự kiện	Mô tả
onclick	Sự kiện xảy ra khi người dùng click chuột vào phần tử
ondblclick	Sự kiện xảy ra khi người dùng click kép chuột vào phần tử
onmouseenter	Sự kiện xảy ra khi người dùng di chuyển con trỏ vào phần tử
onmouseleave	Sự kiện xảy ra khi người dùng di chuyển con trỏ ra khỏi phần tử.

# SỰ KIỆN TRONG JAVASCRIPT

## 2. Các sự kiện trong Javascript

onkeydown	Sự kiện xảy ra khi người dùng đang nhấn một phím
onkeyup	Sự kiện xảy ra khi người dùng thả phím ra
oncopy	Sự kiện xảy ra khi người dùng sao chép nội dung của phần tử
oncut	Sự kiện xảy ra khi người dùng cắt nội dung của phần tử
onpaste	Sự kiện xảy ra khi người dùng dán nội dung vào phần tử
onchange	Sự kiện xảy ra khi người dùng thay đổi giá trị của phần tử



# SỰ KIỆN TRONG JAVASCRIPT

## 3. Bắt sự kiện trong Javascript là gì ?

- **"Bắt sự kiện"** là khi một sự kiện nào đó xảy ra thì ta muốn JavaScript phản ứng lại với sự kiện đó bằng việc thực thi một đoạn mã xác định.

- Ví dụ, tôi muốn khi người dùng  thì câu lệnh  sẽ được thực thi.

# SỰ KIỆN TRONG JAVASCRIPT

## 3. Bắt sự kiện trong Javascript là gì ?

- Khi người dùng di chuyển con trỏ vào phần tử này thì màu nền của phần tử này sẽ chuyển sang màu hồng (tức là câu lệnh

```
this.style.backgroundColor='pink' được thực thi)
```

- Khi người dùng di chuyển con trỏ ra khỏi phần tử này thì màu nền của phần tử này sẽ chuyển sang màu trắng (tức là câu lệnh

```
this.style.backgroundColor='white' được thực thi)
```

- Khi người dùng di chuyển con trỏ vào phần tử này thì màu nền của phần tử này sẽ chuyển sang màu hồng (tức là câu lệnh

```
this.style.backgroundColor='pink' được thực thi)
```

- Khi người dùng di chuyển con trỏ ra khỏi phần tử này thì màu nền của phần tử này sẽ chuyển sang màu trắng (tức là câu lệnh

```
this.style.backgroundColor='white' được thực thi)
```

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

- Trước khi bắt sự kiện thì ta cần phải xác định rõ ba thành phần:

- (1) Phần tử dùng để xảy ra sự kiện.
- (2) Sự kiện sẽ xảy ra.
- (3) Đoạn mã sẽ được thực thi khi sự kiện xảy ra.

- Dưới đây là cú pháp dùng để bắt sự kiện:

```
<Tên-phần-tử Tên-sự-kiện="đoạn mã sẽ được thực thi khi sự kiện xảy ra">
```

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

### ❑ Ví dụ 1:

Khi người dùng *click kép* chuột vào nút "*Xem kết quả*" thì đoạn mã:

```
var a = 100;  
var b = 50;  
var result = (a + b)*2;  
alert('Kết quả của biểu thức là: ' + result)
```

 sẽ được thực thi.

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

```
<body>  
  
  <button onclick="var a = 100;var b = 50;var result = (a + b)*2;alert('Kết quả của biểu thức là: ' +  
  result)">Xem kết quả</button>  
  
</body>
```

Xem kết quả

webcoban.vn cho biết  
Kết quả của biểu thức là: 300

OK

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

### ❑ Ví dụ 2:

Khi người dùng click chuột vào nút "Xin chào" thì hàm `hello()` sẽ được thực thi

```
<button onclick="hello()">Xin chào</button>

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

<script>
  function hello(){
    var name = "Nguyễn Thành Nhân";
    var year = 1993;
    var str = "Tôi tên " + name + " sinh năm " + year;
    document.getElementById("demo").innerHTML = str;
  }
</script>
```

Xin chào

Tôi tên Nguyễn Thành Nhân sinh năm 1993

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

- Lưu ý: Với cùng một phần tử, ta có thể bắt cho nó nhiều sự kiện.
- Phần tử <div> có id là demo bên dưới được bắt hai sự kiện:
  - Khi người dùng di chuyển con trỏ vào nó thì sự kiện **onmouseenter** xảy ra và câu lệnh **this.style.backgroundColor='yellow'** sẽ được thực thi.
  - Khi người dùng di chuyển con trỏ vào nó thì sự kiện **onmouseleave** xảy ra và câu lệnh **this.style.backgroundColor='gray'** sẽ được thực thi.

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

### ❑ Ví dụ 3:

```
<head>
  <style type="text/css">
    #demo{
      width:300px;
      height:200px;
      background-color:gray;
    }
  </style>
</head>
<body>
  <div id="demo" onmouseenter="this.style.backgroundColor='yellow'"
onmouseleave="this.style.backgroundColor='gray'"></div>
</body>
```





# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

- Lưu ý:

- Nếu đoạn mã được viết bên trong bên trong cặp dấu nháy kép thì trong đoạn mã tuyệt đối không được chứa ký tự là dấu nháy kép.
- Nếu đoạn mã được viết bên trong bên trong cặp dấu nháy đơn thì trong đoạn mã tuyệt đối không được chứa ký tự là dấu nháy đơn.

```
<button onclick="alert('Hello')">Xin chào</button> <!-- SAI -->  
<button onclick='alert("Hello")'>Xin chào</button> <!-- SAI -->  
<button onclick="alert('Hello')">Xin chào</button> <!-- ĐÚNG -->  
<button onclick='alert("Hello")'>Xin chào</button> <!-- ĐÚNG -->
```

# SỰ KIỆN TRONG JAVASCRIPT

## 4. Cách bắt sự kiện trong Javascript

### ❑ Ví dụ 4:

```
<body>  
  <input type="text" onkeydown="this.style.backgroundColor='pink'" onkeyup="this.style.backgroundColor='yellow'"  
  placeholder="gõ ký tự vào đây">  
</body>
```

gõ ký tự vào đây

abc

# SỰ KIỆN TRONG JAVASCRIPT

Tài liệu hướng dẫn học lập trình [web](#) Tài liệu hướng dẫn học lập trình [web](#)

## 4. Cách bắt sự kiện trong Javascript

### ❑ Ví dụ 5:

```
<body>
  <textarea cols="50" rows="15" oncopy="alert('Bạn vừa sao chép nội dung của phần tử này')" oncut="alert('Bạn vừa cắt nội dung của phần tử này')" onpaste="alert('Bạn vừa dán nội dung vào phần tử này')">Tài liệu hướng dẫn học lập trình web</textarea>
</body>
```

webcoban.vn cho biết

Bạn vừa sao chép nội dung của phần tử này

OK

webcoban.vn cho biết

Bạn vừa dán nội dung vào phần tử này

OK

webcoban.vn cho biết

Bạn vừa cắt nội dung của phần tử này

OK



Thank you !