

ĐỐI TƯỢNG MẢNG VÀ HÀM

A – LÝ THUYẾT

I. ĐỐI TƯỢNG MẢNG

1. Định nghĩa

Mảng là một đối tượng (hay có thể coi nó là một kiểu dữ liệu) được sử dụng để lưu trữ nhiều giá trị trong một biến duy nhất.

2. Khai báo một mảng

Để khai báo một mảng chúng ta có thể sử dụng một trong các cách sau

Cách 1

```
var tenMang = new Array();
```

```
tenMang[0] = value1;
```

```
tenMang[1] = value2;
```

```
...
```

```
tenMang[N-1] = valueN;
```

Trong đó

- ✓ **Key**: chỉ số thứ tự của phần tử trong mảng. Đánh số thứ tự bắt đầu từ 0 với phần tử đầu tiên
- ✓ **Value**: giá trị của phần tử trong mảng

```
<script language="javascript" type="text/javascript">  
    var mang = new Array();  
    mang[0]="Nguyễn văn A";  
    mang[1]=23;  
    mang[2]="Thái Nguyên";  
</script>
```

Cách 2

var tenMang = new Array(value1, value2, ... valueN);

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
</script>
```

3. Thêm một phần tử vào mảng

Cú pháp

tenMang[Key] = Value;

Trong đó

- ✓ Giá trị của Key chính là vị trí của phần tử được thêm vào trong mảng

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mang[3] = "sinh viên";
</script>
```

4. Xuất giá trị của phần tử mảng

Cú pháp

document.write(tenMang[Key]);

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    document.write(mang[0]);
</script>
```

5. Duyệt mảng

Để duyệt qua tất cả các phần trong một mảng chúng ta sử dụng cú pháp FOR...IN

Cú pháp

```
for(x in tenMang){  
  
    document.write(tenMang[x]);  
  
}
```

```
<script language="javascript" type="text/javascript">  
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");  
    for(x in mang) {  
        document.write(mang[x]);  
    }  
</script>
```

II. MỘT SỐ PHƯƠNG THỨC LÀM VIỆC VỚI MẢNG

1. Phương thức Length

Phương thức Length dùng để đếm số phần tử có trong mảng

Cú pháp

```
mang.length;
```

```
<script language="javascript" type="text/javascript">  
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");  
    $soPhanTu = mang.length;  
    document.write($soPhanTu);  
</script>
```

2. Phương thức Concat()

Phương thức concat() dùng để gom tất cả các phần tử của nhiều mảng riêng rẽ tập trung vào một mảng

Cú pháp

mang.concat();

```
<script language="javascript" type="text/javascript">
    var mang1 = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    var mang2 = new Array("Nguyễn Thị B", 20, "Hà Nội");
    mang = mang1.concat(mang2);
    document.write(mang);
</script>
```

3. Phương thức Join()

Phương thức join() giúp chúng ta tạo ra một chuỗi từ các phần tử trong một mảng và được ngăn cách bằng một ký tự bất kỳ

Cú pháp

mang.join();

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mangjoin=mang.join('--');
    document.write(mangjoin);
</script>
```

4. Phương thức Pop()

Phương thức pop() dùng để xóa phần tử cuối cùng trong một mảng

Cú pháp

mang.pop();

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mang.pop();
    document.write(mang);
</script>
```

5. Phương thức Push()

Phương thức push() dùng để thêm một hoặc nhiều phần tử vào cuối mảng

Cú pháp

mang.push();

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mang.push("sinh viên");
    document.write(mang);
</script>
```

6. Phương thức Splice()

Phương thức splice() sử dụng để kết hợp việc xóa phần tử cũ khỏi mảng đồng thời thêm một phần tử mới tiếp sau các phần tử còn lại trong mảng

Cú pháp

mang.splice(index, count, element1, ...elementN);

Trong đó

- ✓ **index:** vị trí phần tử muốn loại
- ✓ **count:** Số phần tử muốn loại tính từ vị trí index
- ✓ **element1, ...element** là các phần tử mới thêm vào

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mang.splice(2,1,"Hà Nội");
    document.write(mang);
</script>
```

7. Phương thức Sort()

Phương thức sort() dùng để sắp xếp mảng theo thứ tự tăng dần

Cú pháp

mang.sort();

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mang.sort();
    document.write(mang);
</script>
```

8. Phương thức Reverse()

Phương thức reverse() trả lại một mảng với thứ tự các phần tử bị đảo ngược

Cú pháp

mang.reverse();

```
<script language="javascript" type="text/javascript">
    var mang = new Array("Nguyễn Văn A", 23, "Thái Nguyên");
    mang.reverse();
    document.write(mang);
</script>
```

III. HÀM TRONG JAVASCRIPT

1. Định nghĩa Hàm

Hàm hay còn được gọi là *Function* được sử dụng để đóng gói nhóm mã xây dựng nên một chức năng nhất định

2. Hàm tự định nghĩa không có tham số

Cú pháp

```
function tenHam() {  
  
    /* Các lệnh thực thi Hàm */  
  
    }
```

Trong đó

- ✓ Tên Hàm được đặt giống tên của Biến
- ✓ Thực thi hàm bằng cách gọi tới tên hàm: tenHam();

```
<script language="javascript" type="text/javascript">  
    function chao () {  
        document.write ("xin chào!");  
    }  
    chao ();  
</script>
```

3. Hàm tự định nghĩa có tham số

Cú pháp

```
function tenHam (x1, x2, ... xN){  
  
    /* Các lệnh thực thi Hàm */  
  
    }
```

```
<script language="javascript" type="text/javascript">
    function chao(x) {
        document.write("xin chào:" + x);
    }
    chao("Nguyễn Văn A");
</script>
```

4. Hàm tự định nghĩa có giá trị trả về

Cú pháp

```
function tenHam(){
    /* Các lệnh thực thi Hàm */
    return ketQua;
}
```

```
<script language="javascript" type="text/javascript">
    function chao(x) {
        var hello = "xin chào:" + x;
        return hello;
    }
    var loichao = chao("Nguyễn văn A");
    document.write(loichao);
</script>
```

5. Biến toàn cục và Biến cục bộ đối với Hàm

Biến cục bộ là Biến chỉ có tác dụng (Tồn tại) bên trong Hàm

Ngược lại thì Biến Toàn cục là biến có tác dụng (Tồn tại) cả bên trong và bên ngoài Hàm

IV. HÀM XÂY DỰNG SẴN TRONG JAVASCRIPT

1. Hàm isNaN()

Hàm `isNaN()` kiểm tra giá trị. Nếu không phải số trả về `true`, ngược lại thì nếu là số thì trả về `false`

Cú pháp

isNaN();

```
<script language="javascript" type="text/javascript">
    var x = 10;
    if(isNaN(x) == false){
        document.write("x là một số.");
    }
    else {
        document.write("x không là một số.");
    }
</script>
```

2. Hàm Eval()

Hàm `eval()` biến chuỗi thành biểu thức tính toán được hoặc biến chuỗi thành lệnh thì hành được như là mã lệnh của JS

Cú pháp

Eval();

```
<script language="javascript" type="text/javascript">
    var bieuthuc = "10 - 5";
    document.write(bieuthuc);
    var hieu = eval(bieuthuc);
    document.write(" = "+hieu);
</script>
```

3. Hàm SetTimeout()

Hàm `SetTimeout()` có tác dụng gọi một hàm sau thời gian định trước, và chỉ thực thi một lần duy nhất

Cú pháp

`setTimeout("code", Delaytime);`

Trong đó

- ✓ **code**: có thể là một hàm hoặc một câu lệnh
- ✓ **Delaytime**: thời gian truyền vào được tính theo mili giây

```
<script language="javascript" type="text/javascript">
    setTimeout(function() {
        alert("welcome to CNTT");
    }, 3000);
</script>
```

4. Hàm setInterval()

Hàm `SetInterval()` có tác dụng gọi một hàm sau thời gian định trước, được thực thi trong suốt quá trình chạy file hiện hành

Cú pháp

`SetInterval("code", Delaytime);`

Trong đó

- ✓ **code**: có thể là một hàm hoặc một câu lệnh
- ✓ **Delaytime**: thời gian truyền vào được tính theo mili giây

```
<script language="javascript" type="text/javascript">
    setInterval(function() {
        alert("welcome to CNTT");
    }, 3000);
</script>
```

V. CÁC PHƯƠNG THỨC (HÀM) CỦA ĐỐI TƯỢNG MATH

1. Bảng thống kê các phương thức phổ biến trong Math

STT	Phương Thức	Chức Năng	Ví Dụ
1	Math.round(x)	Làm tròn	Math.round(5.6) = 6
2	Math.floor(x)	Làm tròn xuống	Math.floor(4.5) = 4
3	Math.random()	Lấy số ngẫu nhiên $0 < x < 1$	Math.random() = 0.73...
4	Math.max(x,y,z,...N)	Lấy số có giá trị lớn nhất trong chuỗi số	Math.max(1,5,3,7) = 7
5	Math.min(x,y,z,...N)	Lấy số có giá trị nhỏ nhất trong chuỗi số	Math.min(1,5,3,7) = 1
6	Math.abs(x)	Trả về giá trị số dương	Math.abs(-9.5) = 9.5
7	Math.PI	Hằng số PI	Math.PI = 3.14...
8	Math.sqrt(x)	Lấy căn bậc 2 của một số	Math.sqrt(9) = 3

B – BÀI TẬP

I. BÀI TẬP THỰC HÀNH

Bài 1:

Áp dụng kiến thức về Mảng & Hàm để Xây dựng Hàm cho phép người dùng sử dụng để đếm số phần tử có Key chẵn và số phần tử có Key lẻ của một mảng bất kỳ

Bài 2:

Áp dụng kiến thức về Hàm để Viết Ứng dụng sau:

- ✓ Chạy Ứng dụng thì xuất hiện hộp thoại Prompt yêu cầu nhập vào Số thứ nhất (Số bất kỳ)
- ✓ Sau khi bấm OK thì xuất hiện hộp thoại Prompt tiếp theo yêu cầu nhập vào một trong 5 ký tự sau (+, -, x, :, %)
- ✓ Sau khi bấm OK thì xuất hiện hộp thoại Prompt tiếp theo yêu cầu nhập vào Số thứ hai (Số bất kỳ)
- ✓ Sau lần bấm OK này thì xuất hiện hộp thoại Prompt thông báo kết quả của Số thứ nhất phép toán với Số thứ hai

VD: Kết quả của 10 x 5 bằng 50

II. BÀI TẬP VỀ NHÀ

Bài 3:

Viết Ứng dụng tạo mảng tự động với các Giá trị của các phần tử mảng do người dùng tự nhập vào từ hộp thoại Prompt. Người dùng muốn kết thúc việc nhập giá trị các phần tử mảng thì chỉ cần nhập giá trị của phần tử mảng bằng 0. Sau khi kết thúc sẽ thấy thông báo tất cả các giá trị của tất cả các phần tử trong mảng vừa được người dùng nhập vào

Bài 4:

Viết Ứng dụng yêu cầu người dùng nhập vào 10 Số Nguyên Dương bất kỳ, mỗi Số Nguyên Dương được nhập vào từ hộp thoại Prompt. Sau khi người dùng nhập đầy đủ và bấm OK thì hiển thị tất cả giá trị của các phần tử thuộc mảng Lẻ (Bao gồm chính các Số Nguyên Dương lẻ mà người dùng nhập vào ở trên) và tất cả giá trị của các phần tử thuộc mảng Chẵn (Bao gồm chính các Số Nguyên Dương chẵn mà người dùng nhập vào ở trên)

Bài 5:

Viết Ứng dụng hiển thị hình ảnh ngẫu nhiên, mà cứ mỗi lần người dùng tải lại trang web thì sẽ lại được hiển thị một bức ảnh khác.

Bài 6:

Viết Ứng dụng cho phép người dùng tạo Bảng và nhập Dữ liệu trực tiếp cho Bảng thông qua hộp thoại Prompt

Chạy file bài giải để hiểu rõ chương trình