

Câu lệnh điều kiện và vòng lặp trong JavaScript

Câu lệnh điều kiện if, else if và các vòng lặp như vòng lặp for, vòng lặp while, vòng lặp do while, lệnh rẽ nhánh switch, break, continue

Trong phần tìm hiểu về các cấu trúc điều khiển lệnh với Javascript như: **Khối lệnh** **Lệnh if** **Lệnh switch** **vòng lặp for** **vòng lặp while** **vòng lặp do ... while** **lệnh continue, break**

Khối lệnh trong Javascript

Nhiều câu lệnh Javascript có thể nhóm với nhau tạo ra khối lệnh, các khối lệnh được nhóm bằng cặp dấu ngoặc nhọn {}, các khối thường sử dụng trong các câu lệnh điều khiển rẽ nhánh, vòng lặp ...

```
{  
    //Các dòng lệnh trong khối  
}
```

Khối lệnh cũng có thể dán nhãn cho nó, để sau này từ vị trí khác, bạn có thể nhảy đến thi hành các lệnh trong khối (với lệnh **continue**, **break**)

```
labelexamp : {  
    //Các dòng lệnh trong khối có nhãn labelexamp  
}
```

Chú ý khai báo biến với **var** thì nó không bị giới hạn trong khối, khai báo biến với **let** thì phạm vi hiệu lực chỉ trong khối khai báo

```
var a = 5;  
{  
    var a = 2;  
}  
console.log(a);  
//Xuất ra là 2 => biến a trong và ngoài khối là 1  
  
let b = 10;  
{  
    let b = 20;  
}  
//Xuất ra là 10 => biến b trong và ngoài khối là khác nhau  
  
alert(b);
```

Lệnh if, if ... else

Lệnh if : nếu điều kiện là đúng (**true**) thì thi hành các lệnh trong khối - nếu điều kiện sai **false** thì khối lệnh sau nó bị bỏ qua, cú pháp là:

```
if (điều_kiện) {  
    //Các dòng lệnh trong khối
```

```
}
```

Ví dụ:

```
var myNum1 = 7;  
var myNum2 = 10;  
if (myNum1 < myNum2) {  
    alert("JavaScript rất dễ học.");  
}
```

Thi hành đoạn mã trên sẽ hiện thị hộp thoại thông báo: JavaScript rất dễ học. vì điều kiện là `true` nên lệnh `alert` thi hành

if ... else ...

Nếu biểu thức logic là `true` thi hành các lệnh trong khối `if`, nếu `false` thì thi hành khối lệnh `else`

```
if (expression) {  
    // thi hành các lệnh ở đây (khối if) nếu true  
}  
else {  
    // thi hành các lệnh (khối else) nếu điều kiện false  
}
```

Ví dụ

```
var myNum1 = 7;  
var myNum2 = 10;  
if (myNum1 > myNum2) {  
    alert("Đây là khối lệnh 1");  
}  
else {  
    alert("Đây là khối lệnh 2");  
}
```

Chạy code trên hiện thị thông báo: Đây là khối lệnh 2 vì `myNum1 > myNum2` là `false`

khối lệnh else if

`else if` sẽ tạo ra câu lệnh điều kiện `if` mới nếu điều kiện trước đó `false`

```
var course = 1;  
if (course == 1) {  
    document.write("HTML Tutorial");  
} else if (course == 2) {  
    document.write("CSS Tutorial");  
} else {  
    document.write("JavaScript Tutorial");  
}
```

Chạy code trên, trình duyệt sẽ hiện thị `HTML Tutorial`, nếu bạn gán biến `course` thì hiện thị `CSS Tutorial`, nếu gán `course` khác với 1 và 2 thì sẽ hiện thị `JavaScript Tutorial`

Lệnh switch

Trong trường hợp bạn có rẽ nhánh (nhiều điều kiện) khác nhau thay vì sử dụng nhiều `else if` hãy dùng `switch` với cú pháp

```
switch (expression) {  
  case n1:  
    //.. thi hành nếu expression bằng n1  
    break;  
  case n2:  
    //.. thi hành nếu expression bằng n2  
    break;  
  default:  
    //.. mặc định thi hành nếu expression không bằng giá trị nào ở trên  
}
```

Ví dụ:

```
var day = 2;  
switch (day) {  
  case 1:  
    document.write("Monday");  
    break;  
  case 2:  
    document.write("Tuesday");  
    break;  
  case 3:  
    document.write("Wednesday");  
    break;  
  default:  
    document.write("Another day");  
}  
// Outputs "Tuesday"
```

Lưu ý từ khóa `break`; để điều hướng ra khỏi khối. Thường bạn cần có `break`; ở cuối mỗi khối của lệnh `switch`, nếu thiếu `break` thì sẽ không thoát lệnh mà sẽ thi hành khối tiếp theo. Ở ví dụ trên, nếu bỏ đi `break`, thì script sẽ chạy từ điểm rẽ nhánh `case 2`; cho đến cuối (xuất ra: Tuesday, Wednesday, Another day)

Từ khóa `default`: định nghĩa khối mặc định, khối này thi hành nếu tất cả các điều kiện rẽ nhánh không thỏa mãn.

Vòng lặp for

Cú pháp:

```
for (statement1; statement2; statement3) {  
  Khối lệnh thi hành  
}
```

- `statement1`: lệnh thi hành trước khi vòng lặp `for` bắt đầu

- **statement2** : điều kiện kiểm tra trước mỗi lần thi hành khối lệnh **for** (**true** thì khối lệnh sẽ thi hành, **false** sẽ khối for sẽ không thi hành - thoát lặp)
- **statement3** : thi hành sau mỗi lần một vòng hoàn thành

Ví dụ:

```
for (i=1; i<=5; i++) {
    document.write(i + "<br >");
}
//In ra
1
2
3
4
5
```

Có thể bỏ qua **statement1** (vẫn giữ lại dấu ;)

```
var i = 1;
for (; i<=5; i++) {
    document.write(i + "<br >");
}
//In ra
1
2
3
4
5
```

Tương tự bạn có thể bỏ qua **statement3** và **statement2** (vẫn giữ ;), lưu ý bạn cũng có thể sử dụng lệnh **break**; để thoát vòng lặp.

```
var i = 0;
for (;i+=2) {
    if (i>10) break;           //Thoát vòng Lặp
    document.write(i + "<br >");
}
```

for ... in

for ... in duyệt qua các key của đối tượng đếm được (không dùng cho mảng)

```
//Cú pháp
for (let yourvar in object) {
}

//Ví dụ
let ob = {key1: 1, key2: 2};
for (let k in ob) {
    console.log(k);
}
//In ra: key1, key2
```

for ... of

Lệnh **for ... of** duyệt qua các đối tượng.

```
//Cú pháp
for (let yourvar in objects) {
}

//Ví dụ
let a = ["A", "B", "C"];
for (let e in a) {
    console.log(e);
}
//In ra A,B,C
```

Vòng lặp while

Thi hành khối lệnh khi mà điều kiện kiểm tra vẫn là **true**

```
while (điều-kiện) {
    //Khối lệnh
}
```

Đầu tiên nó kiểm tra điều kiện, nếu **true** sẽ thi hành khối lệnh. Đến cuối khối lại kiểm tra điều kiện, nếu điều kiện vẫn là **true** thì lại tiếp tục thi hành vòng mới của khối lệnh.

Ví dụ

```
var i=0;
while (i<=5) {
    document.write(i + "<br >");
    i++;
}
//In ra
0
1
2
3
4
5
```

Lưu ý về việc sau một số vòng thì điều kiện phải là **false** nếu không vòng lặp sẽ lặp lại vô tận.

Vòng lặp do while

Giống với vòng lặp **while** nhưng khối lệnh thi hành luôn mà không kiểm tra điều kiện trước, khi khối lệnh thi hành xong mới kiểm tra điều kiện để xem có lặp lại hay không

Cú pháp

```
do {
    //Khối lệnh
}
while (condition);
```

Ví dụ

```
var i=20;
do {
    document.write(i + "<br >");
    i++;
}
while (i<=25);

//In ra
20
21
22
23
24
25
```

Vòng lặp **do ... while** khối lệnh luôn được thi hành ít nhất một lần

Lệnh continue và break

Trong vòng lặp khi gặp **continue**; nó sẽ bỏ qua các lệnh còn lại và khởi tạo vòng lặp mới luôn. Còn nếu gặp **break**; thì bỏ qua các lệnh còn lại đồng thời thoát khỏi vòng lặp.

```
for (i = 0; i <= 70000; i++) {
    if (i == 5) {
        continue;                //Khởi tạo vòng lặp mới luôn
    }
    document.write(i + "<br >");
    if (i >=7) {
        break;                    //Thoát lặp nếu i >=7
    }
}
```

//In ra các số: (bỏ qua 5)

```
0
1
2
3
4
6
7
```

Lệnh **continue** còn dùng để nhảy đến một khối lệnh có nhãn bằng cú pháp

```
continue nhãn_khối_lệnh;
```

Lệnh **break** còn dùng để hủy thi hành khối lệnh bên ngoài có nhãn, với cú pháp:

```
break nhãn_khối_lệnh_ngoài;
```

Phần nói về các đối tượng có kiểu liệt kê được (ví dụ như mảng, danh sách ...), còn có các lệnh duyệt qua từng phần tử liệt kê được đó với các lệnh **for ... in**, **for ... of**.