
BÀI 06: LÀM VIỆC VỚI JAVASCRIPT

Lê Quang Lợi

Bài 06: Làm việc với javascript

Nội dung

- » Giới thiệu
- » Một số ví dụ
- » Cú pháp
- » Hàm cơ bản

6.1 Giới thiệu

- » Ngôn ngữ kịch bản chạy trên trình duyệt
- » Javascript cho phép tương(đọc/ghi) tác với tài liệu HTML
- » Hỗ trợ tương tác với người dùng qua giao diện(**sự kiện**)
- » **Ngôn ngữ thông dịch**(mã nguồn => thành chương trình)
- » Cú pháp tương tự **C,C++, java**

```
<script type="text/javascript">  
    document.write("Hello World!");  
</script>
```

6.2 Tạo Javascript trong tài liệu HTML

- » Thuộc tính Onclick: viết lệnh trên thẻ HTML

```
<h1 onClick="alert('Chào các bạn!');">Hello</h1>
```

- » Thẻ **script**: chứa mã javascript trên trang HTML

```
<script language="javascript" >  
    alert('Chào các bạn!');  
</script>
```

- » Nhúng file script: nhiều file ".js" nhúng vào trong HTML

```
<script language="javascript" src="URL"></script>
```

6.3 Cơ bản về cú pháp HTML

- » Câu lệnh: một câu lệnh kết thúc bởi dấu “;”
- » Chú thích: miêu tả câu lệnh, khi dịch được bỏ qua
 - dòng: **//** Nội dung
 - khối: **/*** Nội dung chú thích ***/**
- » Quy định về xâu: cặp “ và ” hoặc cặp ‘ và ’
“Chào các bạn”
- » Biến **không cần phải khai báo trước**, có thể gán dữ liệu bất kì
- » Từ khóa **var**: cho phép khai báo biến
var a; var b=20; alert(b);

6.3.1 Một số đối tượng/hàm cơ bản

» **Alert**: thông báo lời nhắn ra ngoài màn hình

```
alert('Chào các bạn');
```

» **Write** và **writeline**: viết ra tài liệu HTML một dòng

```
document.write("<h1>Chào các bạn </h1>");
```

» Hàm **NaN**: kiểm tra giá trị không phải là số

```
NaN("ab123");//=> true; NaN("1213");// => false
```

» Hàm **Eval**: Chuyển đổi giá trị sang kiểu số

```
var a= Eval("12bc"); var b= Eval("1213");// b = 1213
```

6.3.2 Kiểu dữ liệu

- » Kiểu đối tượng: đối tượng chứa dữ liệu bất kì
- » Kiểu boolean: kiểu đúng sai (**true/false**)
- » Kiểu mảng: thể hiện mảng các phần tử => buổi khác
- » Một số hằng số:
 - * **null**: hằng trống của một chuỗi **var myString=null;**
 - * **true/false**: hằng đúng hoặc sai của kiểu trả về (hàm, biểu thức, đối tượng ..)

6.3.2 Kiểu dữ liệu

- » Kiểu số (số nguyên và số thực): 10, 10.5
- » Đối tượng math: Chứa các hàm hỗ trợ tính toán với các con số

Math.sin(x); Math.abs(X); Math.PI, Math.sqrt(x)...

Ví dụ: **var a = Math.sqrt(10);**

- » Kiểu chuỗi (mảng các ký tự): ký hiệu “ ” hoặc ‘ ’

- » Thao tác với chuỗi:

```
var myLen = s.length;
```

```
last_char = s.charAt(s.length - 1);
```

```
sub = s.substring(1,4);
```

```
i = s.indexOf('a');
```


6.3.3 Biểu thức và toán tử

- » Toán tử toán học: `+, -, *, %, ++, --`
- » Toán tử gán: `=, +=, -=, *=, /=`
- » Toán tử quan hệ: `>, >=, <, <=, ==, !=`
- » Toán tử logic: `and(&&), or(||), not(!)`
- » Toán tử thao tác chuỗi: `=, +=, +`
- » Biểu thức: kết hợp logic giữa các toán tử và toán hạng

Ví dụ: `a = 10 * 2 + 23; b = a / 4;`

6.4 Cấu trúc điều khiển

- » Javascript hỗ trợ các cấu trúc điều khiển: if, if else, for, while
- » Cho phép điều khiển chương trình
- » Xây dựng cấu trúc lặp
 - Lựa chọn đúng cấu trúc
 - Xây dựng biểu thức điều kiện
 - Xây dựng nội dung phù hợp

6.4.1 Cấu trúc điểm khiển

Cấu trúc rẽ nhánh (if)

```
» if(btđk){
```

```
    // khối lệnh 01
```

```
}
```

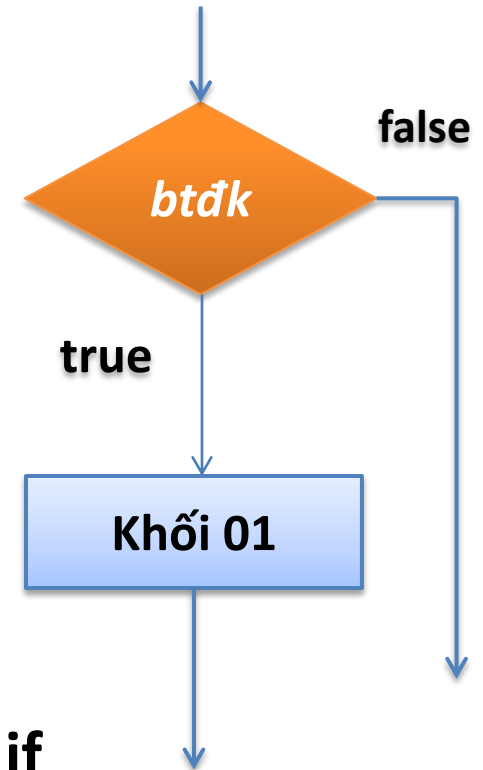
» **btđk**: là biểu thức trả về giá trị logic

» Nếu **btđk** đúng thì thực thi khối lệnh 01

» Nếu **btđk** sai: thực thi lệnh nằm ngay sau if

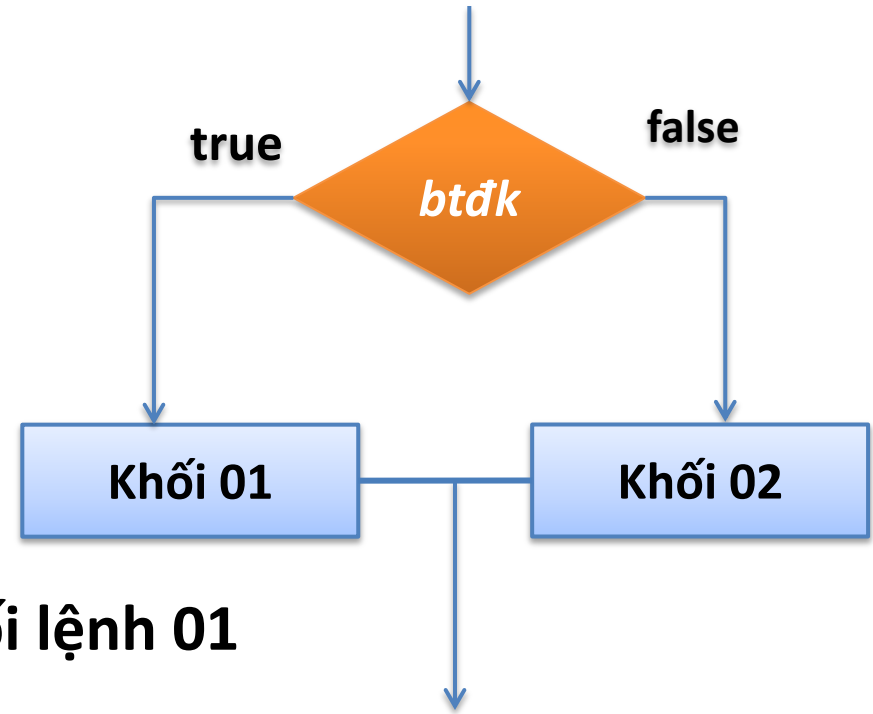
Ví dụ:

```
if(a==5){alert(a);}
```



6.4.1 Cơ bản về cú pháp HTML

```
» if(btđk){  
  // khối lệnh 01  
} else{  
  // khối lệnh 02  
}
```



- » Nếu **btđk** đúng thì thực thi khối lệnh 01
- » Nếu **btđk** sai: thực thi khối lệnh 02

Ví dụ:

```
if(a==5){alert(a);} else{ alert(a + " không phải là số 05 ");}
```

6.4.2 Cấu trúc lặp

```
Var sum =0;  
Var i=0; // khởi gán  
while( i<20){ // điều kiện  
    sum +=i; // lệnh  
    i++;  
}
```

```
Var sum =0;  
Var i=0; // khởi gán  
do{  
    sum +=i; // lệnh  
    i++;  
} while( i<20); // điều kiện
```

6.4.2 Cấu trúc lặp

Vòng lặp for: thực hiện các công việc lặp

```
for(bt01; bt02; bt03){ // khối lệnh}
```

+ **bt01**: tạo giá trị bắt đầu

+ **bt02**: biểu thức điều kiện dừng (**false**)

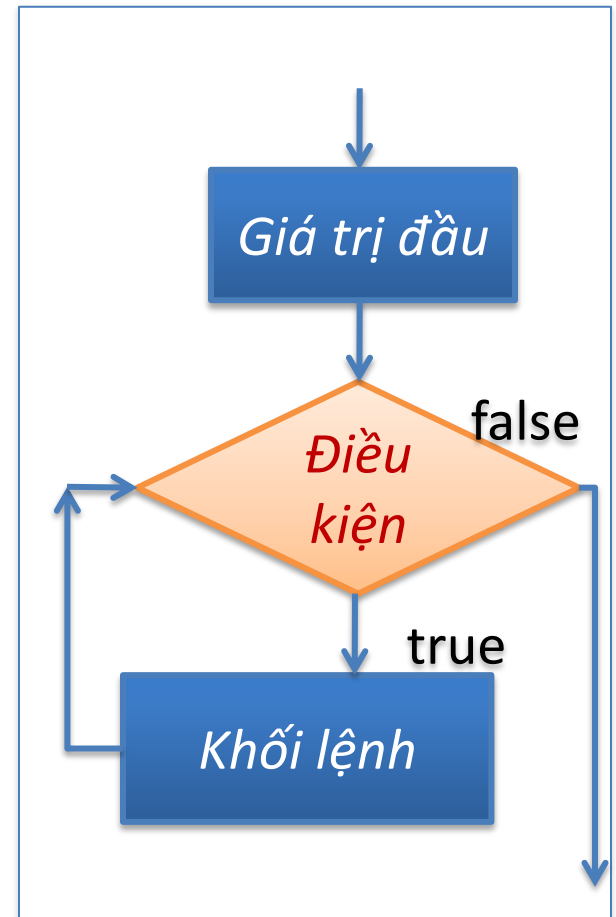
+ **bt03**: biểu thức điều khiển bt02 về **false**

```
var sum =0;
```

```
for( var i=0; i<10; i++) {
```

```
    sum+=0; // lặp lại 10 lần
```

```
}
```



6.4.2 Cấu trúc lặp

Vòng lặp while: lặp không các định

```
while( btđk){ // khối lệnh }
```

```
do{// khối lệnh}while(btđk);
```

+ **btđk**: biểu thức **logic** điều kiện vòng lặp

While nếu **true** thì tiếp tục, **false** dừng

+ khối lệnh: nội dung lặp lại

Chú ý: khối lệnh luôn có câu lệnh làm

Thay đổi giá trị của btđk về **false**

