

# Bài 7

# Câu lệnh lặp

Module: BOOTCAMP PREPARATION

# Kiểm tra bài trước

Hỏi và trao đổi về các khó khăn gặp phải trong bài “Cấu trúc điều kiện”

Tóm tắt lại các phần đã học từ bài “Cấu trúc điều kiện”

# Mục tiêu

---

- Giải thích được khái niệm lặp
- Mô tả được cú pháp của vòng lặp for, while, do/while
- Giải thích được cách sử dụng lệnh break, continue
- Trình bày được vòng lặp lồng nhau
- Sử dụng được vòng lặp for, while, do/while
- Sử dụng được lệnh break, continue
- So sánh được các loại vòng lặp
- Sử dụng được vòng lặp lồng nhau

# Thảo luận

Vòng lặp for

# Vòng lặp (loop)

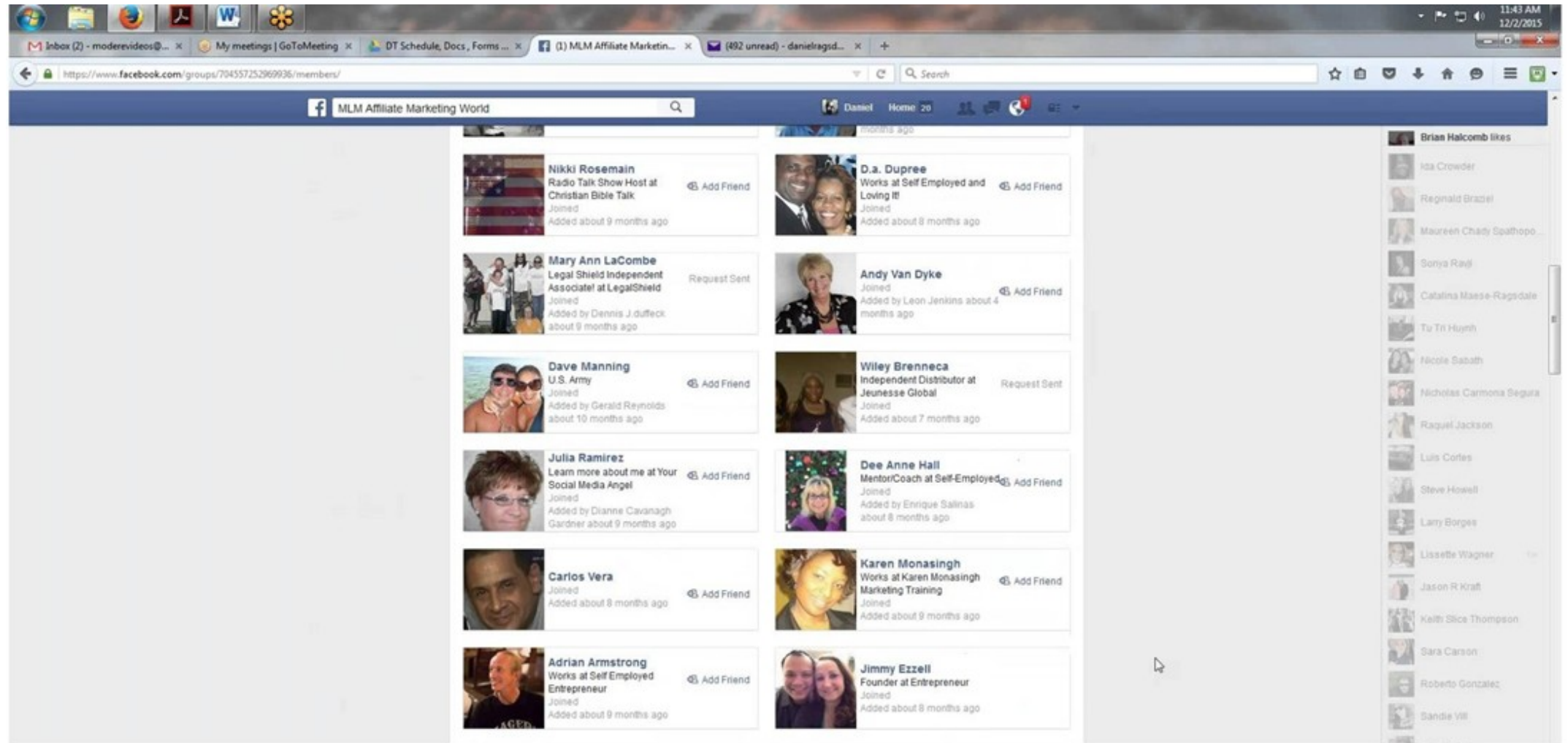
---

- Vòng lặp cho phép tự động thực hiện một khối lệnh lặp đi lặp lại nhiều lần dựa vào một điều kiện cho trước
- Vòng lặp giúp cho lập trình viên viết được các mã nguồn ngắn gọn hơn so với việc phải viết lặp lại những dòng mã tương tự nhau
- Vòng lặp bao gồm:
  - Một điều kiện để đánh giá (lúc nào thì lặp?)
  - Một khối lệnh để thực thi (lặp cái gì?)
- Các vòng lặp hỗ trợ bởi JavaScript: *for*, *while* và *do-while*

# Vòng lặp: Ví dụ

Customers who bought this product ▲							
Copy	Print	Excel	CSV	PDF	Reset column order	Show 10 orders	Search <input type="text"/>
Date	Order	Billing First name	Billing Last name	Billing E-mail	Billing Phone	Billing Address 1	
2017-05-14	<a href="#">2712</a>	John	Doe	<a href="mailto:john@johndoe.com">john@johndoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	20 John avenue	
2017-05-06	<a href="#">2645</a>	Jane	Doe	<a href="mailto:jane@janedoe.com">jane@janedoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	72 Jane street	
2017-05-06	<a href="#">2645</a>	James	Doe	<a href="mailto:james@jamesdoe.com">james@jamesdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	433 James plaza	
2017-07-21	<a href="#">2919</a>	Jeremy	Doe	<a href="mailto:jeremy@jeremydoe.com">jeremy@jeremydoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	154 Jeremy lane	Creel
2017-07-21	<a href="#">2919</a>	Janis	Doe	<a href="mailto:janis@janisdoe.com">janis@janisdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	747 Janis road	Creel
2017-06-22	<a href="#">2905</a>	Jolene	Doe	<a href="mailto:janis@janisdoe.com">janis@janisdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	4830 Jolene circle	
2017-05-14	<a href="#">2710</a>	Jasper	Doe	<a href="mailto:jasper@jasperdoe.com">jasper@jasperdoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	765 Jasper route	
2017-05-13	<a href="#">2709</a>	Jackie	Doe	<a href="mailto:jackie@jackiedoe.com">jackie@jackiedoe.com</a>	<a href="tel:555-555-5555">555-555-5555</a>	7373 Jackie crescent	406
Showing 1 to 8 of 8 entries 1 row selected						« < > »	

# Vòng lặp: Ví dụ



# Vòng lặp: Ví dụ

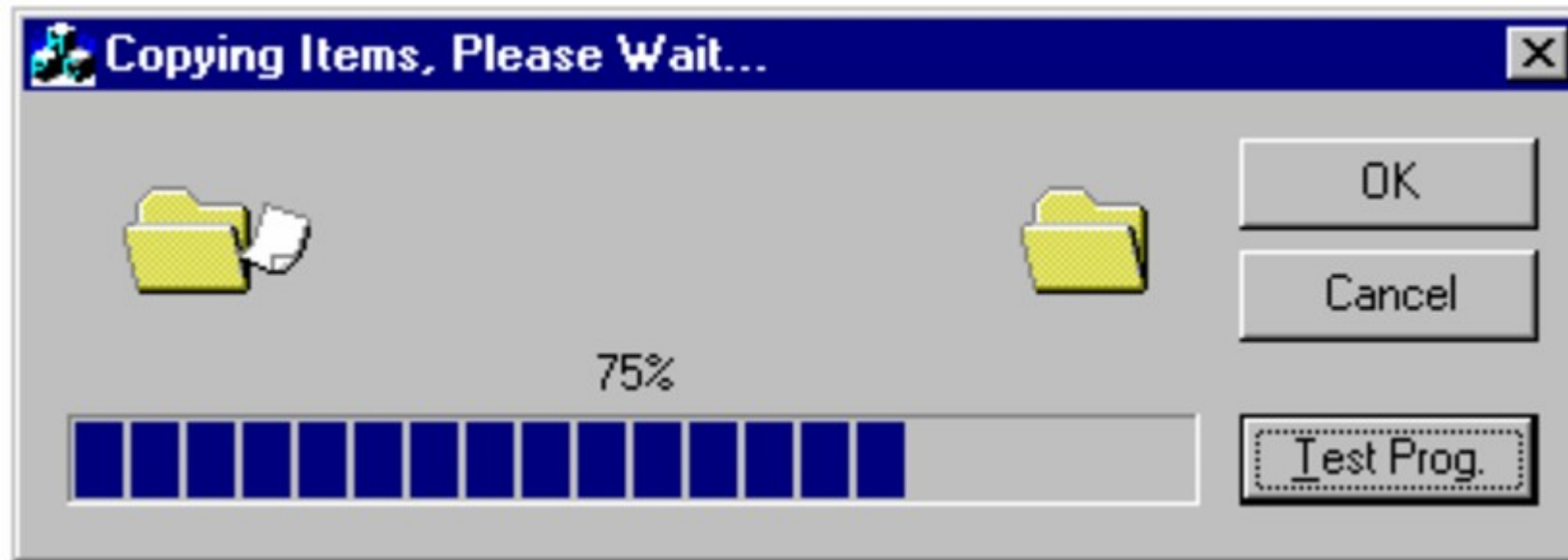
---





# Vòng lặp: Ví dụ

---



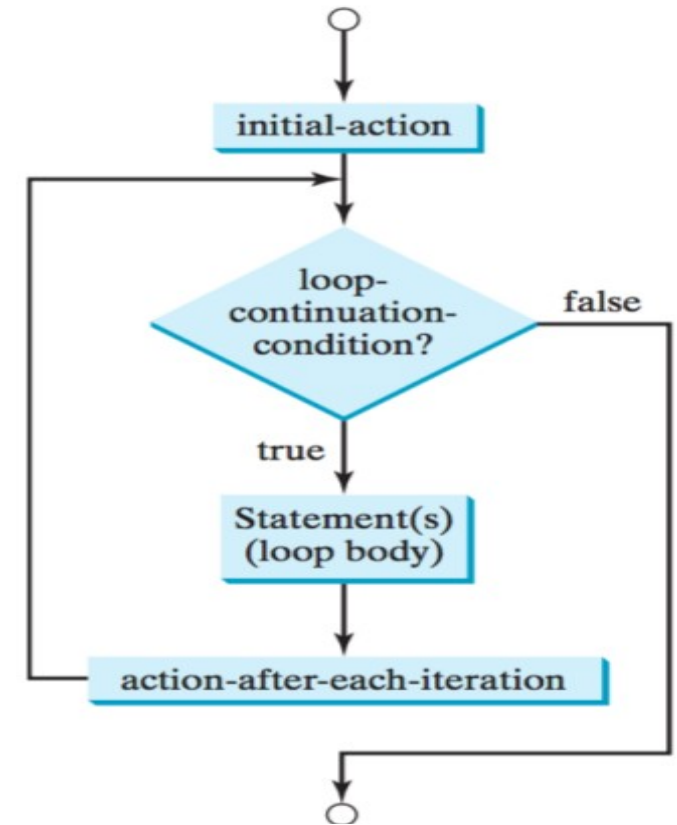
# Vòng lặp for

- Cú pháp:

```
for (initial-action; loop-continuation-condition; action-after-each-iteration) {  
    statement(s);  
}
```

Trong đó:

- *initial-action*: là các câu lệnh được thực thi một lần duy nhất khi vòng lặp bắt đầu chạy
- *loop-continuation-condition*: là biểu thức điều kiện để xác định xem vòng lặp có được tiếp tục hay không
- *statement(s)*: là khối lệnh sẽ được thực thi trong mỗi lần lặp
- *action-after-each-iteration*: là các câu lệnh được thực thi sau mỗi lần lặp



# Vòng lặp for: Luồng thực thi

```
for (1initial-action; 2loop-continuation-condition; 4action-after-each-iteration) {  
    3statement(s);  
}
```

1. Các câu lệnh *initial-action* được thực thi
2. Biểu thức điều kiện được đánh giá. Nếu điều kiện đúng thì khối lệnh bên trong thân vòng lặp được thực thi. Nếu điều kiện sai thì vòng lặp kết thúc
3. Khối lệnh bên trong thân vòng lặp được thực thi
4. Các câu lệnh *action-after-each-iteration* được thực thi
5. Bắt đầu vòng lặp mới từ bước 2 – 3 - 4

# Vòng lặp for: Ví dụ

*initial-action*      *loop-continuation-condition*      *action-after-each-iteration*

```
for (var i = 1; i <= 10; i++){  
    var product = 5 * i;  
    document.write("5 x " + i + " = " + product);  
}
```

*statement(s)*

5 x 1 = 5  
5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50

**Lưu ý:** Biến *i* ở vòng lặp trên còn được gọi là biến điều khiển (control variable). Phạm vi của biến *i* là bên trong vòng lặp for.

# Vòng lặp for: Ví dụ

- Hiển thị bảng cửu chương của 5

$$5 \times 1 = 5$$

```

        true  2
        ↓    ↓    ↓
    for (var i = 1; i <= 10; i++){
5  →  var product = 5 * i;
    →  document.write("5 x " + i + " = " + product);
    }
    
```

# Vòng lặp for: Ví dụ

- Hiển thị bảng cửu chương của 5

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

```

      true      3
      ↓        ↓
for (var i = 1; i <= 10; i++){
10 → var product = 5 * i;
    → document.write("5 x " + i + " = " + product);
}

```

# Vòng lặp for: Ví dụ

- Hiển thị bảng cửu chương của 5

```

      false  11
      ↓      ↓
for (var i = 1; i <= 10; i++){
    var product = 5 * i;
    document.write("5 x " + i + " = " + product);
}
  
```

$5 \times 1 = 5$   
 $5 \times 2 = 10$   
 $5 \times 3 = 15$   
 $5 \times 4 = 20$   
 $5 \times 5 = 25$   
 $5 \times 6 = 30$   
 $5 \times 7 = 35$   
 $5 \times 8 = 40$   
 $5 \times 9 = 45$   
 $5 \times 10 = 50$

# Demo

Vòng lặp for



# Sử dụng dấu phẩy trong vòng lặp for

- Có thể sử dụng nhiều câu lệnh bên trong các phần *initial-action* và *action-after-each-iteration* của vòng lặp for, các câu lệnh này cách nhau bởi dấu phẩy

- Ví dụ:

```
for (var i = 0, j = 10; i <= 10; i++, j--) {
    int sum = i + j;
    document.write(i + " + " + j + " = " + sum);
}
```

$$0 + 10 = 10$$

$$1 + 9 = 10$$

$$2 + 8 = 10$$

$$3 + 7 = 10$$

$$4 + 6 = 10$$

$$5 + 5 = 10$$

$$6 + 4 = 10$$

$$7 + 3 = 10$$

$$8 + 2 = 10$$

$$9 + 1 = 10$$

$$10 + 0 = 10$$

# Thảo luận

Vòng lặp while

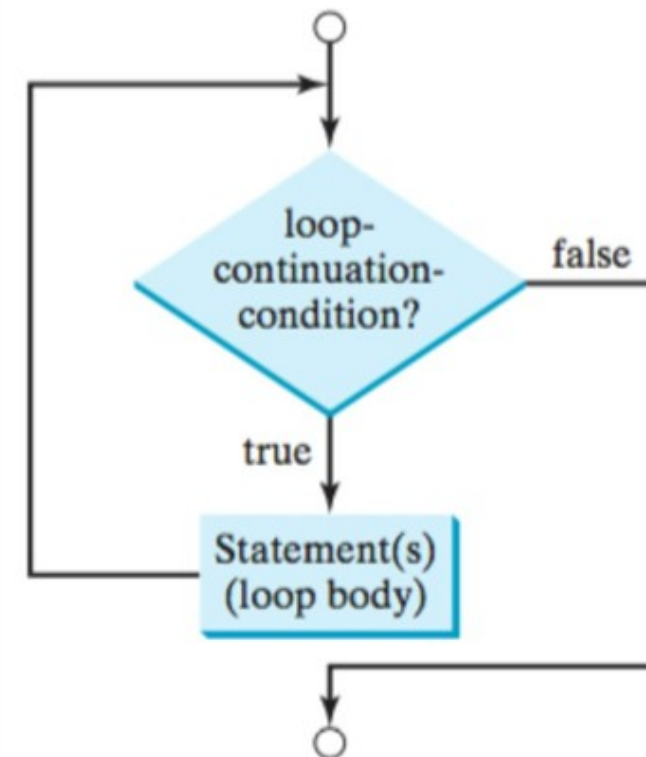
# Vòng lặp while

- Vòng lặp *while* thực thi lặp lại một khối lệnh nếu biểu thức điều kiện trả về giá trị đúng
- Cú pháp:

```
while (loop-continuation-condition) {  
    statement(s);  
}
```

Trong đó:

- *loop-continuation-condition* : là biểu thức điều kiện
- *statement(s)*: là các câu lệnh được thực thi trong mỗi lần lặp



# While: Ví dụ

- Nhập vào một chuỗi để in ra trang web
- Nhập vào ký tự "q" để thoát vòng lặp

```
var text = 'Hello';  
  
while (text !== 'q') {  
    document.write(text + '<br/>');  
    text = prompt('Enter a text to print. Enter "q" to quit.');
```

```
}
```

An embedded page on this page says

Enter a text to print. Enter "q" to quit.

Cancel OK

# Lặp với điều kiện phức tạp

- Nhập vào một chuỗi để in ra trang web
- Nhập vào ký tự "q" hoặc ký tự rỗng để thoát vòng lặp

```
var text = 'Hello';  
  
while (text !== 'q' && text !== '') {  
    document.write(text + '<br/>');  
    text = prompt('Enter a text to print. Enter "q" or empty to quit.');
```

```
}
```

An embedded page on this page says

Enter a text to print. Enter "q" or empty to quit.

Cancel OK

# Lặp vô hạn với while

```
while(true) {  
}
```

```
while(1) {  
}
```

**Lưu ý:** Khi vòng lặp thực thi vô tận thì có thể sẽ hao tốn rất nhiều tài nguyên. Cần thiết kế *điều kiện dừng* cẩn thận để tránh xảy ra trường hợp này.

# Demo

Vòng lặp while

# Lựa chọn while hay for?

---

- Các loại vòng lặp có thể được sử dụng thay thế cho nhau
- Việc lựa chọn vòng lặp cho phù hợp với từng tình huống là quan trọng
- Khác biệt thông thường nhất giữa for và while:
  - for thường được sử dụng khi biết trước số lần lặp
  - while thường được sử dụng khi không biết số lần lặp



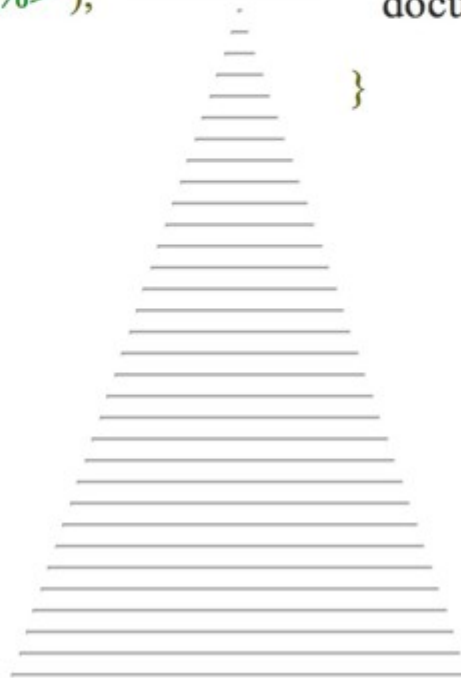
# Chuyển đổi giữa for và while

```
var count = 0;
while (count < 100){
    document.write("<hr width= " + count + "%>");
    count++;
}
```

**while**

```
var count;
for(count = 0; count < 100; count++){
    document.write("<hr width= " + count + "%>");
}
```

**for**



# Lỗi thường gặp #1: Vòng lặp vô tận

- Ví dụ:

```
for(var i = 0; i >= 0; i++){  
    document.write("i = " + i + "<br/>");  
}
```

- Hoặc:

```
var i = 0;  
  
while(i >= 0){  
    document.write("i = " + i + "<br/>");  
    i++;  
}
```

# Lỗi thường gặp #2: Số lần lặp không đúng

---

- Ví dụ, khối lệnh sau sẽ lặp 101 lần thay vì 100 lần:

```
var count = 0;
while (count <= 100) {
    document("Welcome to JavaScript!<br/>");
    count++;
}
```

# Thảo luận

Vòng lặp do-while

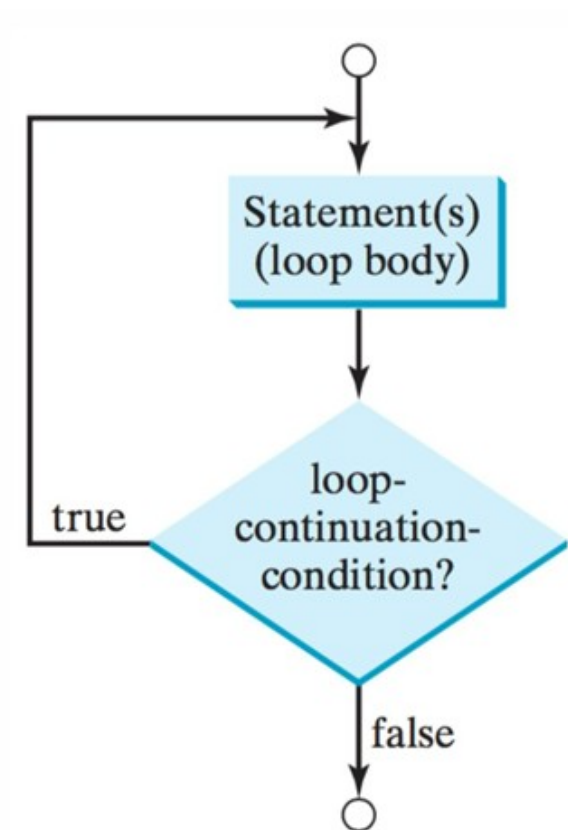
# Vòng lặp do-while

- Cú pháp:

```
do {
    statements(s);
} while (loop-continuation-condition);
```

Trong đó:

- statement(s): Các câu lệnh được thực thi trong mỗi lần lặp
- loop-continuation-condition: Biểu thức điều kiện. Nếu biểu thức điều kiện trả về giá trị true thì vòng lặp sẽ tiếp tục thực thi. Nếu biểu thức điều kiện trả về false thì vòng lặp kết thúc
- Điểm khác biệt giữa **while** và **do-while**: Thân của vòng lặp do-while được thực hiện ít nhất là một lần. Kể cả khi biểu thức điều kiện trả về sai.



# do-while: Ví dụ

- Tính tổng các số nguyên:
  - Tính tổng của các số được nhập vào từ bàn phím
  - Nếu nhập vào số 0 thì kết thúc chương trình

```
var sum = 0;
var number;

do {
    number = parseInt(prompt('Enter a number'));
    sum += number;
} while (number != 0);

document.write("The sum is: " + sum);
```

# Demo

Vòng lặp do-while

# Thảo luận

Lệnh break

Lệnh continue



# Câu lệnh break

- Lệnh break dùng để thoát khỏi vòng lặp mà không đi tới điểm cuối
- Khi gặp câu lệnh này trong vòng lặp, chương trình sẽ thoát ra khỏi vòng lặp và chỉ đến câu lệnh liền sau nó
- Nếu nhiều vòng lặp thì break sẽ thoát ra khỏi vòng lặp gần nhất.
- Ngoài ra, break còn được dùng trong cấu trúc lựa chọn switch

```
var text = ""
var i;
for (i = 0; i < 5; i++) {
    if (i === 3) {
        break;
    }
    text += "The number is " + i + "<br>";
}
```



The result of *text* will be:

```
The number is 0
The number is 1
The number is 2
```

# Câu lệnh continue

- Lệnh continue để nhảy qua một phần của khối lệnh và bắt đầu dòng tiếp theo.
- Khi gặp lệnh này trong các vòng lặp, chương trình sẽ bỏ qua phần còn lại trong vòng lặp và tiếp tục thực hiện lần lặp tiếp theo.

```
var text = ""
var i;
for (i = 0; i < 5; i++) {
    if (i === 3) {
        continue;
    }
    text += "The number is " + i + "<br>";
}
```



The result of *text* will be:

```
The number is 0
The number is 1
The number is 2
The number is 4
```

# Lặp vô hạn

---

- Vòng lặp vô hạn là chuỗi các câu lệnh được thực hiện lặp đi lặp lại không có điểm dừng.
- Nguyên nhân có thể là:
  - Vòng lặp chưa có điều kết thúc vòng lặp
  - Vòng lặp có điều điều kiện để kết thúc vòng lặp nhưng điều kiện này không bao giờ được đáp ứng.
  - Hoặc có thể một logic nào đó dẫn đến lặp vô hạn
- Chương trình gặp vòng lặp vô hạn,
  - Treo máy hoặc bị crash
- Do vậy, cần tránh để xảy ra vòng lặp vô hạn trong chương trình.

# Demo

Lệnh break

Lệnh continue

# Thảo luận

Vòng lặp lồng nhau

# Vòng lặp lồng nhau (nested loops)

- Một vòng lặp có thể được đặt bên trong một vòng lặp khác
- Không hạn chế số lượng vòng lặp lồng nhau

**Lưu ý:** Sử dụng quá nhiều vòng lặp lồng nhau sẽ làm tăng độ phức tạp của mã nguồn lên rất nhiều.

# Ví dụ vòng lặp lồng nhau

- Hiển thị hình chữ nhật:

```
for (int i = 0; i < 5; i++){  
    for (int j = 0; j < 10; j++){  
        console.log("*");  
    }  
    console.log();  
}
```

```
* * * * *  
* * * * *  
* * * * *  
* * * * *  
* * * * *
```

# Demo

Vòng lặp lồng nhau



# Tóm tắt bài học

---

- Vòng lặp cho phép thực hiện các câu lệnh lặp đi lặp lại nhiều lần
- Các vòng lặp được JavaScript hỗ trợ: for, for-in, while
- Vòng lặp for thường được sử dụng khi đã biết trước số lần lặp
- Vòng lặp while thường được sử dụng khi không biết trước số lần lặp
- Vòng lặp for-each (còn được gọi là enhanced for) được sử dụng khi muốn duyệt qua tất cả các phần tử của một collection
- Vòng lặp do-while thực thi phần thân ít nhất là một lần
- Câu lệnh break sẽ kết thúc vòng lặp
- Câu lệnh continue sẽ bỏ qua lần lặp hiện tại
- Nhiều câu lệnh lặp lồng nhau sẽ làm tăng độ phức tạp của thuật toán

# Hướng dẫn

Hướng dẫn làm bài thực hành và bài tập

Chuẩn bị bài tiếp theo: *Mảng*