



CẤU TRÚC ĐIỀU KHIỂN



Mục tiêu bài học

1. Nắm được khái niệm về cấu trúc điều khiển lệnh trong lập trình
2. Biết cách sử dụng JavaScript triển khai các lệnh để điều khiển cấu trúc

CẤU TRÚC LẶP

while và do...while

Lệnh while

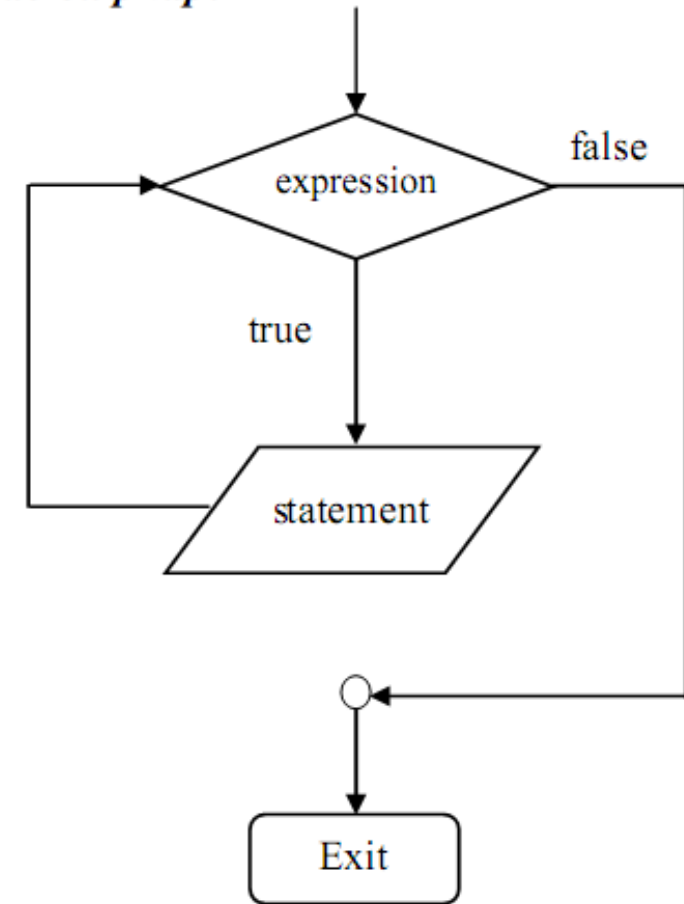
➤ Cú pháp:

```
while (expression) {  
    statement;  
}
```

➤ Ý nghĩa:

1. Xem xét điều kiện **expression**.
2. Nếu **expression** là **true** thì thực hiện **statement** và quay lại bước 1
3. Nếu **expression** là **false** thì thoát khỏi vòng lặp.

Lưu đồ cú pháp:



Ví dụ

- ❖ Viết chương trình tính tổng các số nguyên từ 1 đến n

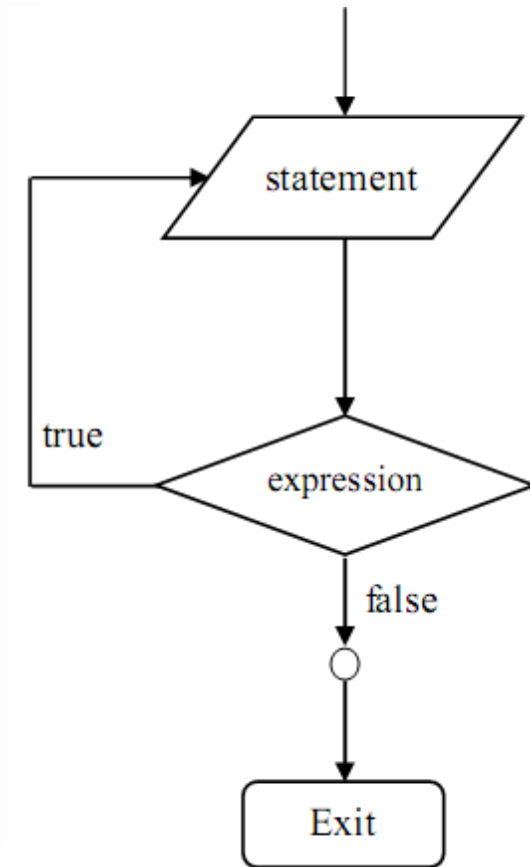
Lệnh do...while

➤ Cú pháp:

```
do {  
    statement;  
}while (expression);
```

➤ Ý nghĩa:

1. **statement** được thực hiện
2. Dựa trên giá trị **expression**.
3. Nếu **expression** là **true** thì quay lại bước 1
4. Nếu **expression** là **false** thì thoát khỏi vòng lặp.



Ví dụ

- ❖ Viết chương trình in dãy số nguyên liên tiếp từ a đến b.
- ❖ Ví dụ: $a = 2$, $b = 8$, kết quả sẽ là

Các số nguyên liên tiếp từ 2 đến 8 gồm:

2

3

4

5

6

7

8

CẤU TRÚC LẬP *for*

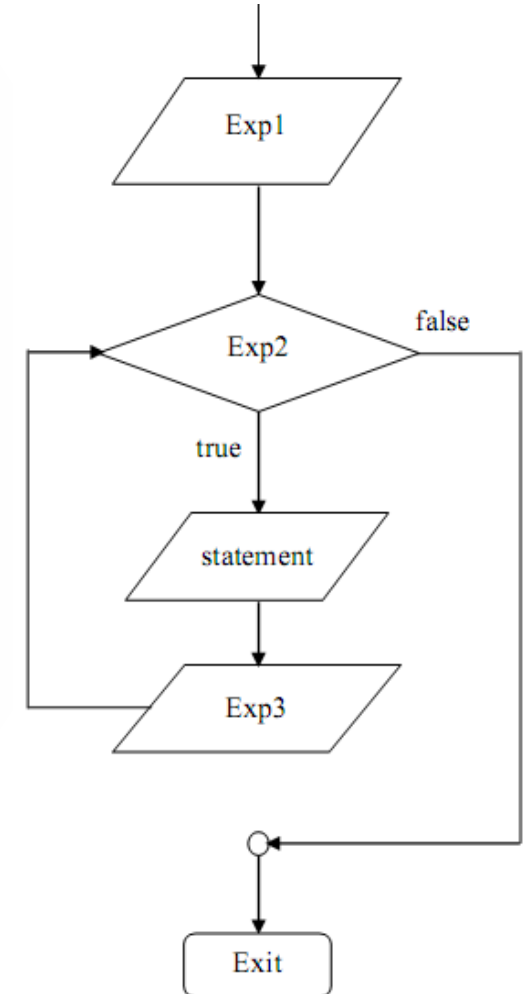
Hoạt động cấu trúc “for”

➤ Cú pháp:

for (**exp1**; **exp2**; **exp3**)
statement;

➤ Ý nghĩa:

1. **exp1**: là biểu thức **khởi tạo** được thực hiện.
2. **exp2**: là biểu thức điều kiện **dừng**
3. **exp3**: biểu thức điều khiển **lặp**



Ví dụ

- ❖ Viết chương trình tính tổng các số nguyên từ 1 đến n .

❖ Cho phép `exp1` là một định nghĩa biến

```
for (var i=1; i<=n; i++) {  
    ...  
}
```

❖ Xóa tất cả các biểu thức trong vòng lặp `for` sẽ cho một vòng lặp vô tận.

```
for (;;) {  
    statement;  
}
```

Dùng for hay while, do..while

- ✍ Đối với **for**: lệnh được lặp lại với số lần lặp mà ta biết trước.
- ✍ Đối với **while, do...while**: Lệnh được lặp lại khi thỏa điều kiện nào đó và ta không biết trước số lần lặp.

Lệnh ngắt vòng lặp “break”

Ý nghĩa việc dùng lệnh “break”

- ✍ Khi gặp lệnh **break** thì lập tức **thoát** khỏi một cấu trúc điều khiển mà không chờ đến biểu thức điều kiện được định trị.
- ✍ Chương trình tự động nhảy đến lệnh tiếp theo.
- ✍ Trong vòng lặp câu lệnh **break** thường đi kèm với câu lệnh **if**.

Ví dụ

- ❖ Lặp lại 10 lần việc hiển thị giá trị của biến number.
- ❖ Sau mỗi lần hiển thị cho giá trị của biến number tăng thêm một.
- ❖ Tuy nhiên, nếu giá trị của biến number lớn hơn 50 thì kết thúc vòng lặp.

Lệnh “**continue**”

Ý nghĩa việc dùng lệnh “continue”

- ➡ Lệnh **continue** dùng để kết thúc sớm vòng lặp hiện tại và quay về đầu vòng lặp.
- ➡ Lệnh **continue** chỉ được dùng trong thân các cấu trúc lặp như **for, while, do...while**.
- ➡ Câu lệnh **continue** thường đi kèm với câu lệnh **if**.

Ví dụ

- ❖ Hiển thị dãy số từ một đến mười (*ngoại trừ các số 2, 5, 9*)
- ❖ Tính tổng các số chia hết cho 3 trong dãy số từ 1 đến 100

1. Các cấu trúc điều khiển qui định thứ tự thực thi các lệnh của chương trình.
2. Có 3 loại cấu trúc điều khiển:
 - i. *Cấu trúc tuần tự* (sequence).
 - ii. *Cấu trúc rẽ nhánh/ lựa chọn* (selection): **if, switch**
 - iii. *Cấu trúc lặp* (loop): **for, while, do...while**

Việc dùng lệnh break, continue?

- ✓ Lệnh **break**: dùng để thoát khỏi một cấu trúc lệnh hiện tại và thực hiện **lệnh** tiếp theo.
- ✓ Lệnh **continue**: khi gặp lệnh **continue** sẽ quay về đầu vòng lặp và thực hiện **lần lặp** tiếp theo.
- ✓ Trong cấu trúc lặp (**for, while, do...while**) thì lệnh **break, continue** thường nằm trong cấu trúc lệnh **if**

Q&A