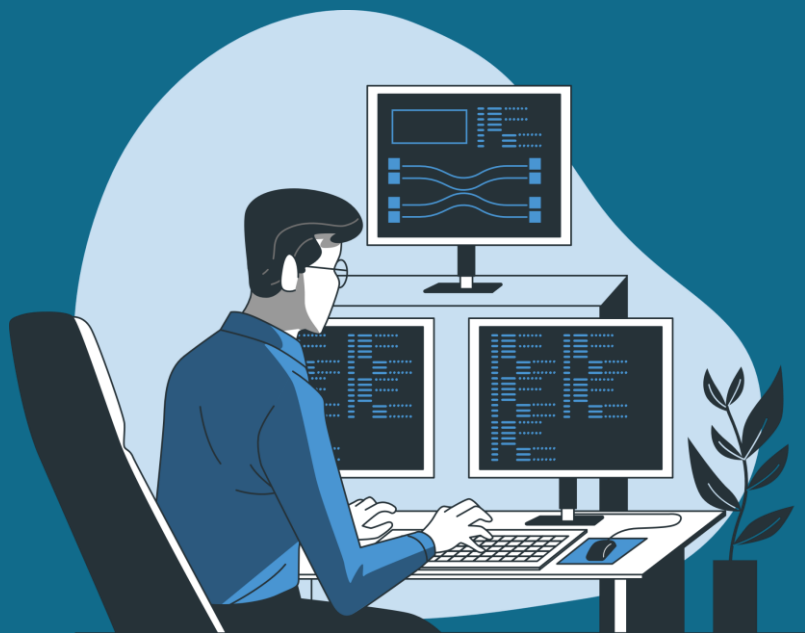




28TECH

Become A Better Developer



VÒNG LẶP (LOOP)





/PROBLEM

Hãy viết chương trình in ra
1000 dòng “Hello World!”.

1000 lệnh
printf





1. Vòng lặp for:

CÚ PHÁP

```
for( [Câu lệnh khởi tạo] ; [Điều kiện lặp]; [Câu lệnh cập nhật]){  
    //Câu lệnh bên trong vòng lặp for  
}
```

Vòng lặp for gồm **3 phần chính**, các phần phân cách nhau bằng **dấu chấm phẩy ;**

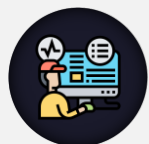
- 1** Câu lệnh khởi tạo thường là khai báo và khởi tạo một biến có nhiệm vụ làm biến đếm cho vòng for (i, j, k).
- 2** Điều kiện lặp, chừng nào giá trị của điều kiện lặp còn có giá trị true thì vòng for còn lặp.
- 3** Cập nhật biến đếm, thường phần này sẽ thay đổi giá trị của biến đếm sau khi các câu lệnh ở bên trong vòng lặp for được thực hiện xong.



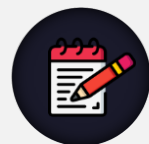


1. Vòng lặp for:

SỬ DỤNG FOR



Vòng lặp for được sử dụng nhiều nhất, và thường được sử dụng khi biết trước số vòng lặp cần thực hiện.

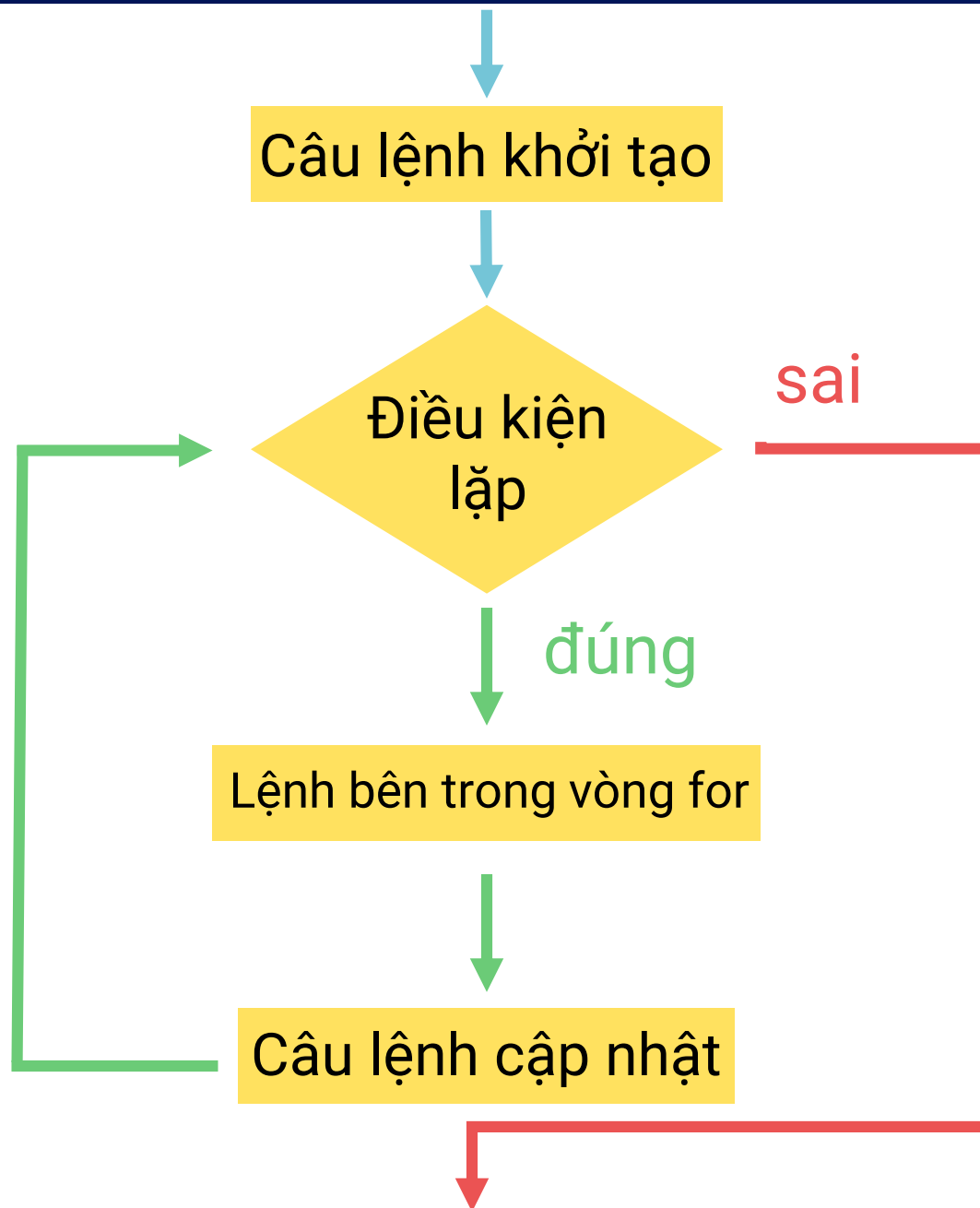


Vòng for không nhất thiết phải có đủ cả 3 phần nhưng phải luôn có đủ 2 dấu ; để phân cách giữa 3 phần.



1. Vòng lặp for:

❖ Sơ đồ khối:



1. Vòng lặp for:

❖ Ví dụ:

Code	Output
<pre>for(int i = 1; i <= 1000; i++){ System.out.println(i + " "); }</pre>	1 2 ... 1000
<pre>for(int i = 1; i <= 10; i +=2){ System.out.println(i + " "); }</pre>	1 3 5 7 9
<pre>for(int i = 1; ; i++){ System.out.println(i + " "); }</pre>	Vòng lặp vĩnh viễn in ra các tự nhiên bắt đầu từ 1

1. Vòng lặp for:

❖ Ví dụ:

Code	Output
<pre>for(; i <= 10; i++){ System.out.println(i + " "); }</pre>	Compilation error
<pre>for(int i = 1; i <= 10;){ System.out.println(i + " "); }</pre>	Vòng lặp in ra vô hạn các số 1
<pre>int i = 1; for(; i <= 100; i++){ } System.out.println(i + " ");</pre>	101

/ BREAK VÀ CONTINUE



Khi muốn dừng vòng lặp for với một điều kiện cho trước ta dùng câu lệnh break.

Khi gặp câu lệnh break, **vòng for sẽ kết thúc ngay lập tức**, thông thường break sẽ đi kèm với câu lệnh if để kiểm tra điều kiện dừng.

```
for ( [Câu lệnh khởi tạo] ; [Điều kiện lặp] ; [Câu lệnh cập nhật] ){  
    //Các câu lệnh trước break  
    if (condition){  
        break; // Câu lệnh sẽ được thực thi khi condition true  
    }  
    //Nhóm câu lệnh sau break  
}
```



/ BREAK VÀ CONTINUE



Khi muốn bỏ qua phần còn lại của vòng lặp hiện tại và chuyển sang vòng lặp mới luôn, ta sử dụng continue. Trong code bên dưới nếu continue được thực hiện, nhóm câu lệnh sau continue sẽ không được thực hiện

```
for ( [Câu lệnh khởi tạo] ; [Điều kiện lặp] ; [Câu lệnh cập nhật] ){  
    //Các câu lệnh trước continue  
    if (condition){  
        continue;  
    }  
    //Nhóm câu lệnh sau continue  
}
```



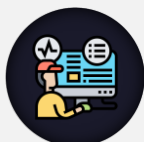


2. Vòng lặp while:

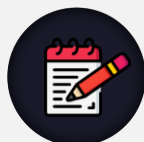
CÚ PHÁP

```
while([Điều kiện lặp]){  
    //Các câu lệnh của vòng lặp  
}
```

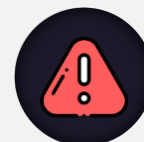
SỬ DỤNG WHILE



Khi điều kiện lặp còn đúng thì vòng while còn tiếp tục hoạt động và các câu lệnh bên trong vòng while sẽ tiếp tục được thực hiện.



Vòng lặp while được sử dụng khi chưa xác định được số vòng lặp cần thực hiện.



Lỗi thường gặp khi sử dụng while : Vòng lặp bị lặp vĩnh viễn.

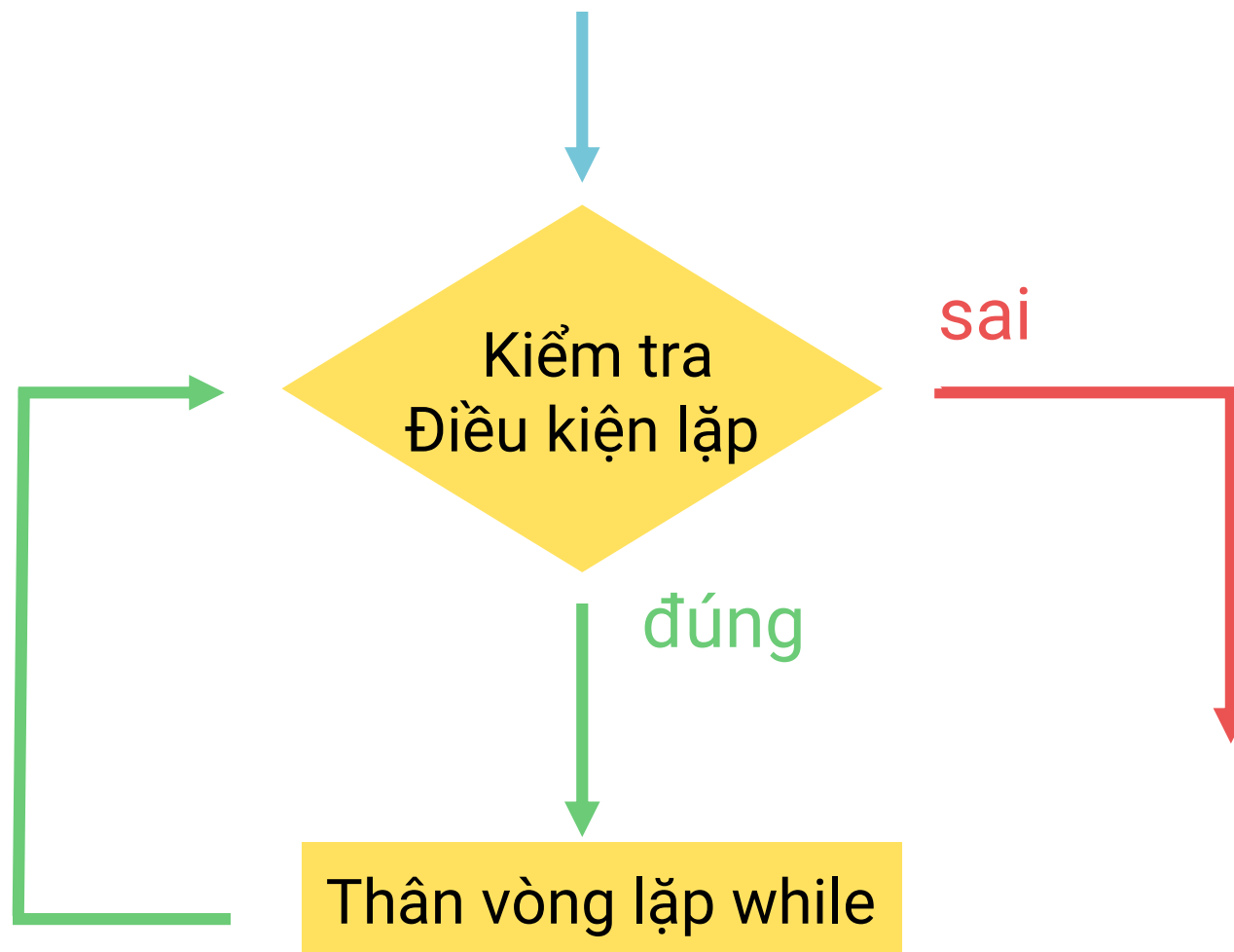




2. Vòng lặp while:

❖ Sơ đồ khối:

Bắt đầu vào vòng lặp while



2. Vòng lặp while:

❖ Ví dụ:

Code	Output
<pre>int i = 1; while(i <= 5){ System.out.println(i + " "); ++i; }</pre>	1 2 3 4 5
<pre>while(true){ System.out.println("28tech"); }</pre>	In ra vô hạn dòng 28tech
<pre>int i = 1; while(i <= 5){ System.out.println(i + " "); }</pre>	In ra vô hạn số 1

2. Vòng lặp while:

❖ Ví dụ:

Code	Output
<pre>int i = 1; while(i <= 5){ i += 2; } System.out.println(i + " ");</pre>	7
<pre>int n = 1234; while(n != 0){ System.out.println(n % 10); n /= 10; // làm mất số hàng đơn vị }</pre>	4 3 2 1
<pre>int n = 1234; int sum = 0; while(n != 0){ sum += n % 10; n /= 10; // làm mất số hàng đơn vị } System.out.println(sum);</pre>	10

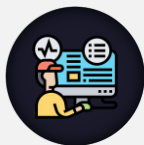


3. Vòng lặp do while:

CÚ PHÁP

```
do{  
    //code bên trong vòng lặp  
}while([Điều kiện lặp]);
```

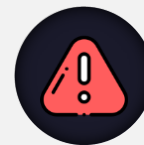
SỬ DỤNG DO WHILE



Hoạt động: Tương tự như vòng while nhưng có một chút sự khác biệt đó là Do while luôn thực thi code bên trong vòng lặp 1 lần đầu tiên, sau đó mới kiểm tra điều kiện lặp.



Vòng lặp do while được sử dụng khi chưa xác định được số vòng lặp cần thực hiện.

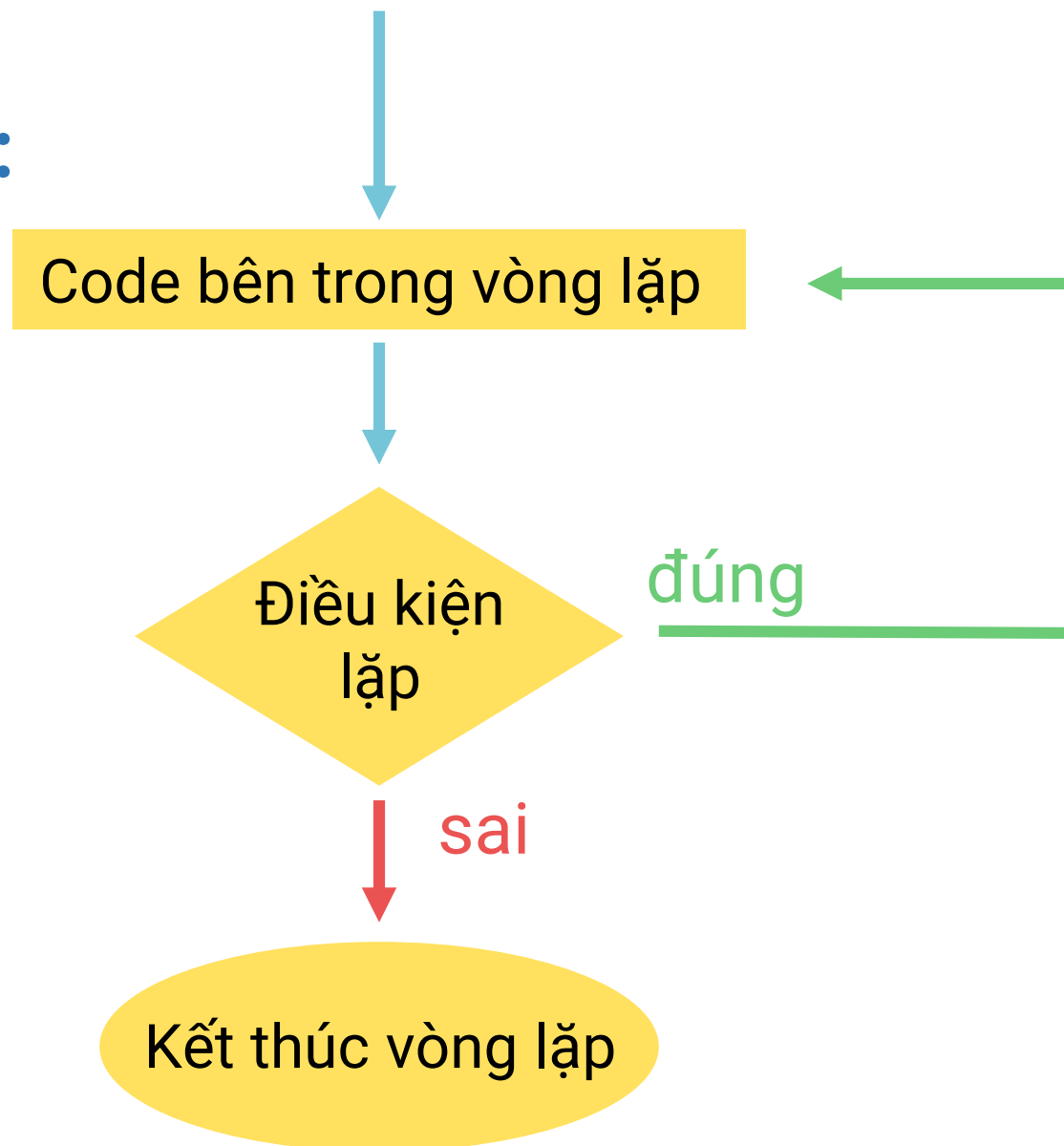


Lỗi thường gặp khi sử dụng do while: Thiếu dấu ; đằng sau while.



3. Vòng lặp do while:

❖ Sơ đồ khối:



3. Vòng lặp do while:

❖ Ví dụ:

Code	Output
<pre>int i = 100; do{ System.out.println(i); }while(i < 100);</pre>	100
<pre>int i = 1; do{ System.out.println(i); }while(i < 5);</pre>	1 2 3 4 5