

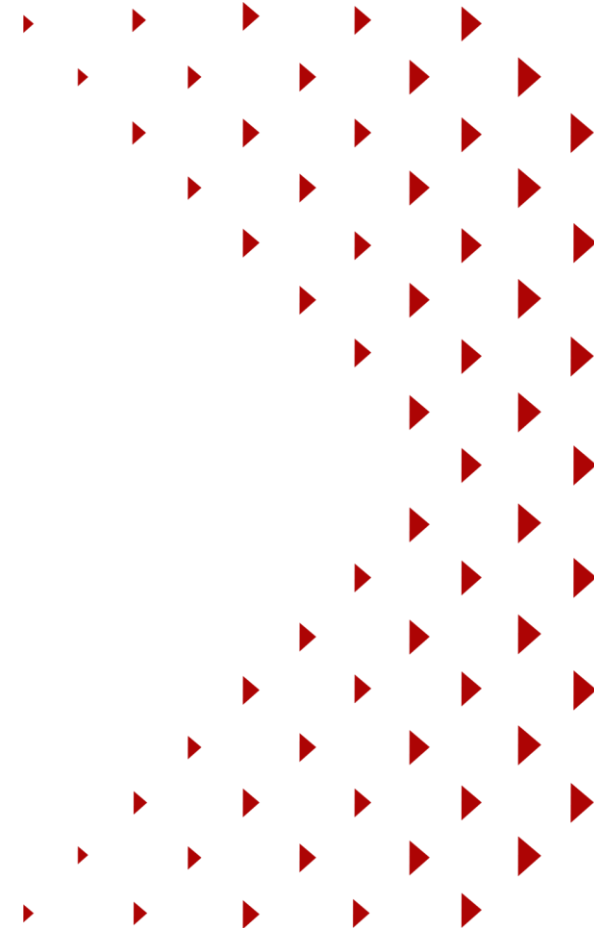


SESSION 10:

DECISION MAKING AND LOOP

Module 1: Web Front-end Fundamental

Version: 2.0



1. Tổng quan câu lệnh điều khiển

2. Câu lệnh điều kiện

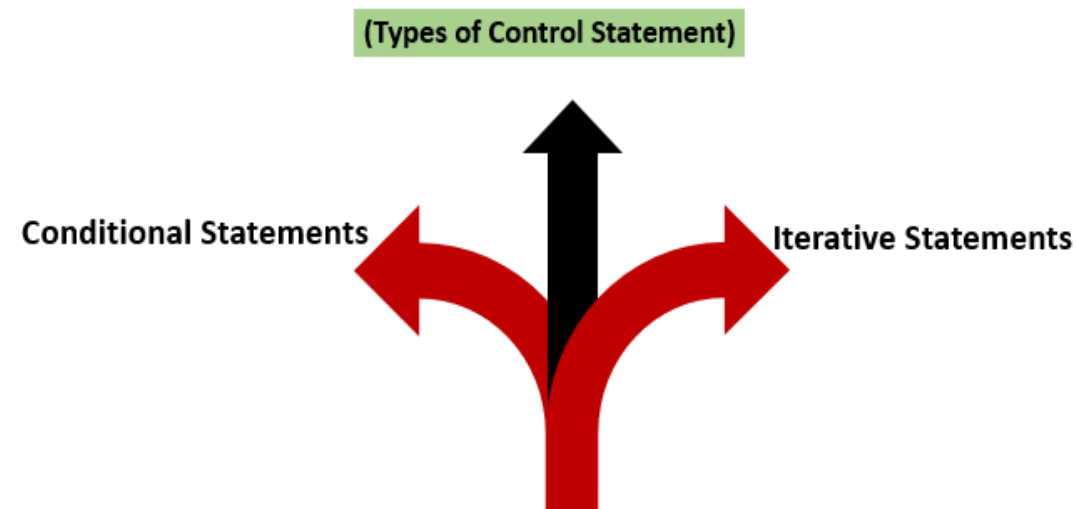
3. Câu lệnh lặp

4. Câu lệnh nhảy

TỔNG QUAN CÂU LỆNH ĐIỀU KHIỂN

- **Control Statement – Câu lệnh điều khiển**

- Câu lệnh có quyền thay đổi luồng thực thi của chương trình
- Quyết định các câu lệnh khác được thực thi và thực thi khi nào
- Các câu lệnh điều khiển:
 - **Condition Statement** (Decision Marking) – Câu lệnh điều kiện
 - **Iterative Statement** (Loop) – Câu lệnh lặp
 - **Jump Statement** – Câu lệnh nhảy



CÂU LỆNH ĐIỀU KIỆN - 1



- **Condition Statement – Câu lệnh điều kiện**
 - Cho phép thay đổi luồng thực thi của chương trình
 - Dựa vào **điều kiện** một câu lệnh hoặc một chuỗi câu lệnh có thể được thực hiện hoặc không
 - Có 2 loại câu lệnh điều kiện
 - Câu điều kiện if
 - Câu điều kiện switch

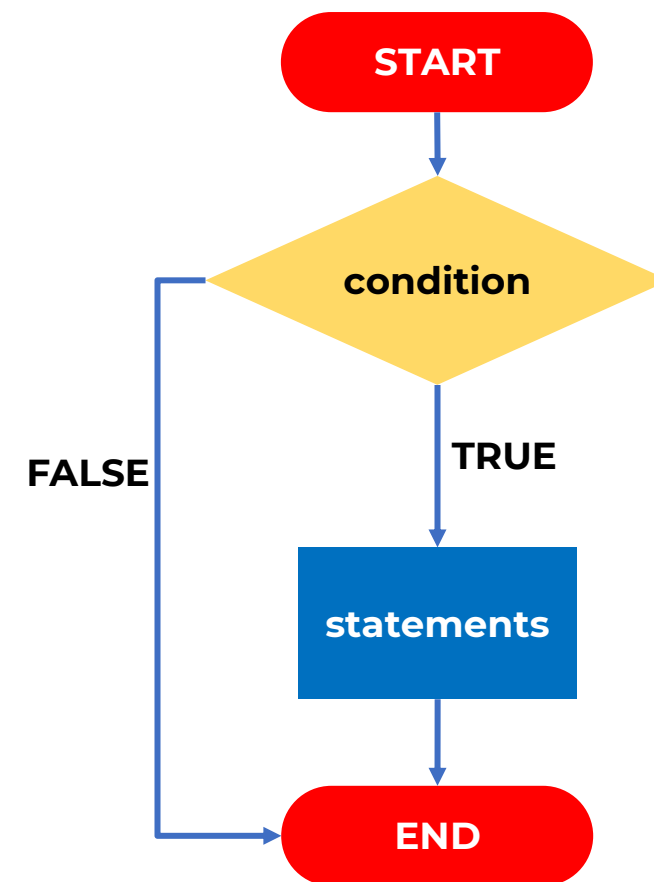
CÂU LỆNH ĐIỀU KIỆN - 2

- **Câu lệnh if**

- Đánh giá một điều kiện và đưa ra lựa chọn thực thi một khối lệnh dựa trên điều kiện đó
- Nếu điều kiện có giá trị đúng (true) thì khối lệnh bên trong if sẽ được thực thi
- Nếu điều kiện có giá trị sai (false) thì khối lệnh bên trong if sẽ được bỏ qua.
- Luồng thực thi sẽ chuyển xuống ngay sau khối lệnh if.
- **Syntax:**

```
if(condition) {  
    // one or more statements  
}
```

- Trong đó
 - condition: là biểu thức trả về giá trị kiểu boolean
 - statements: Các câu lệnh được thực thi nếu điều kiện trả về true

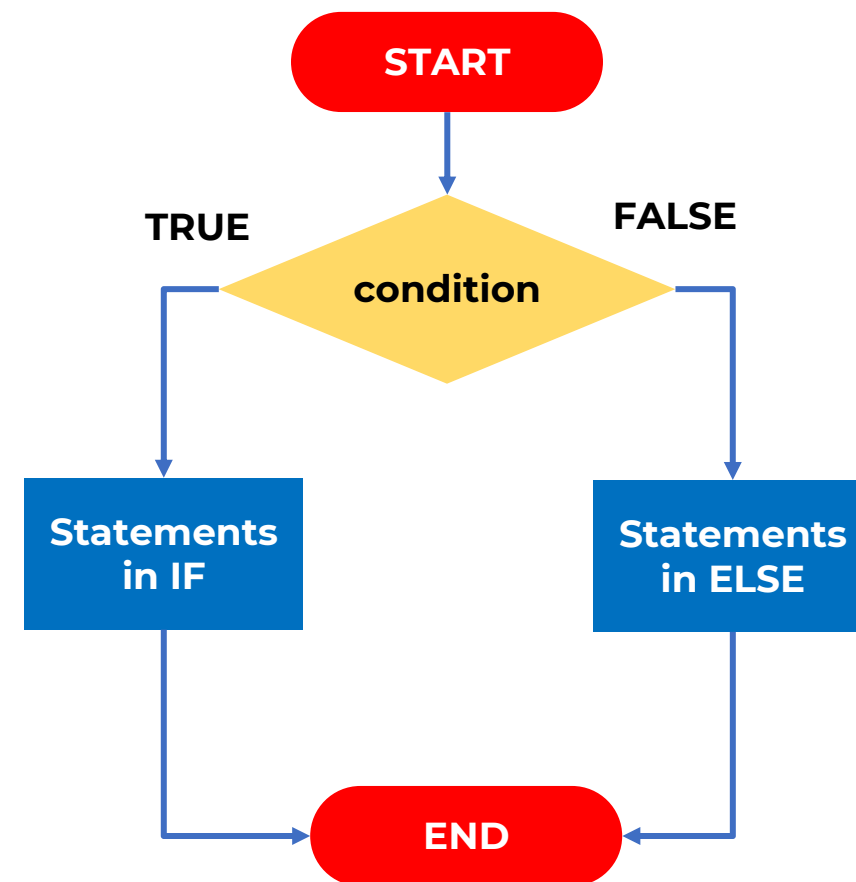


CÂU LỆNH ĐIỀU KIỆN - 3

- **Câu lệnh if...else**

- Đánh giá một điều kiện và đưa ra lựa chọn thực thi một khối lệnh dựa trên điều kiện đó
- Nếu điều kiện có giá trị đúng (true) thì khối lệnh bên trong **if** sẽ được thực thi
- Nếu điều kiện có giá trị sai (false) thì khối lệnh bên trong **else** sẽ được thực thi
- Luồng thực thi sẽ chuyển xuống ngay sau khối lệnh if
- **Syntax:**

```
if(condition) {  
    // one or more statements in if  
} else {  
    // one or more statements in else  
}
```



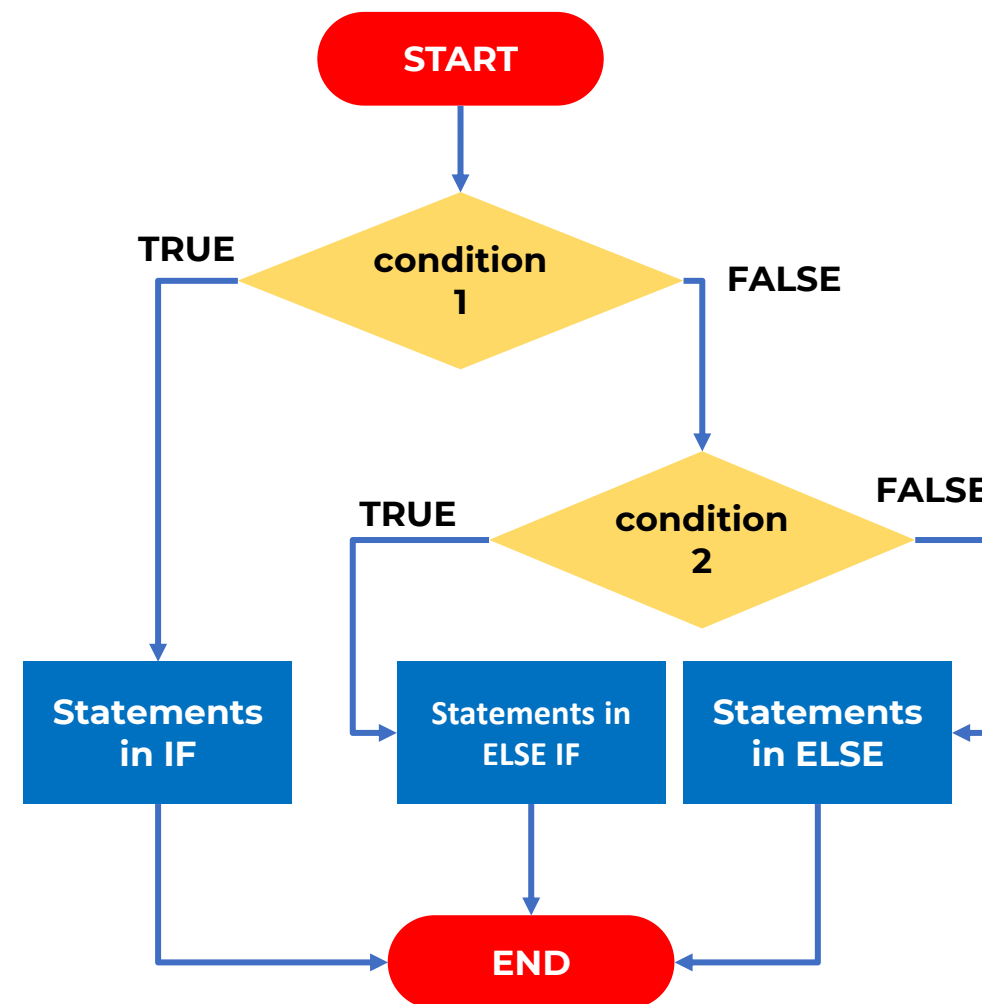
CÂU LỆNH ĐIỀU KIỆN - 4

- **Câu lệnh if...else if...else**

- Đánh giá từng điều kiện và đưa ra lựa chọn thực thi một khối lệnh dựa trên từng điều kiện đó
- Nếu điều kiện đầu tiên có giá trị đúng (true) thì khối lệnh bên trong **if** sẽ được thực thi
- Nếu điều kiện có giá trị sai (false) thì sẽ kiểm tra tiếp điều kiện 2 của else if. Nếu điều kiện đó có giá trị đúng (true) thì khối lệnh bên trong **else if** sẽ được thực thi, nếu giá trị sai (false) thì sẽ thực thi khối lệnh trong **else**
- Luồng thực thi sẽ chuyển xuống ngay sau khối lệnh **if**.

- **Syntax:**

```
if(condition 1) {  
    // one or more statements in if  
} else if(condition 2) {  
    // one or more statements in else if  
} else {  
    // one or more statements in else  
}
```



CÂU LỆNH ĐIỀU KIỆN - 5

- **Câu lệnh switch...case**

- Chọn câu lệnh để thực thi khi giá trị trùng khớp với kết quả
- Giá trị của biểu thức sẽ được so sánh với từng trường hợp (case)
- Nếu có trường hợp bằng nhau thì khối lệnh tương ứng sẽ được thực thi
- Nếu gặp câu lệnh break thì sẽ kết thúc thực thi khối switch-case
- Nếu gặp trường hợp bằng nhau, nhưng sau đó không có câu lệnh break thì tất cả những khối lệnh phía sau cũng được thực thi
- Nếu không có trường hợp nào bằng nhau thì khối lệnh trong default (nếu có) sẽ được thực thi

```
switch (expression) {  
    // Kiểm tra result === value  
    case value1:  
        // Thực thi khối lệnh 1  
        break;  
    case value2:  
        // Thực thi khối lệnh 2  
        break;  
    case valueN:  
        // Thực thi khối lệnh n  
        break;  
    default:  
        // Thực thi khối lệnh khi tất cả các case  
        không được thực hiện  
}
```


CÂU LỆNH KIỂU KIỆN - 6

- So sánh câu lệnh if và switch

	IF ... ELSE	SWITCH
Biểu thức kiểm tra	Có thể kiểm tra đồng thời nhiều điều kiện hoặc nhiều giá trị	Kiểm tra so sánh giữa biểu thức và giá trị
Trình tự code	Thực thi câu lệnh này hoặc câu lệnh khác	Thực thi các case đến khi kết thúc hoặc gặp break

- Lưu ý

- Khi gặp **break**; switch case sẽ lập tức dừng
- Nếu không gặp break hoặc hoặc tất cả case đều không thực hiện khối lệnh bên trong, thì khối lệnh trong **default**(nếu có) sẽ được thực hiện
- Có thể chuyển đổi giữa if else và switch case tùy thuộc vào mục đích sử dụng

CÂU LỆNH LẶP - 1

- **Vòng lặp for**

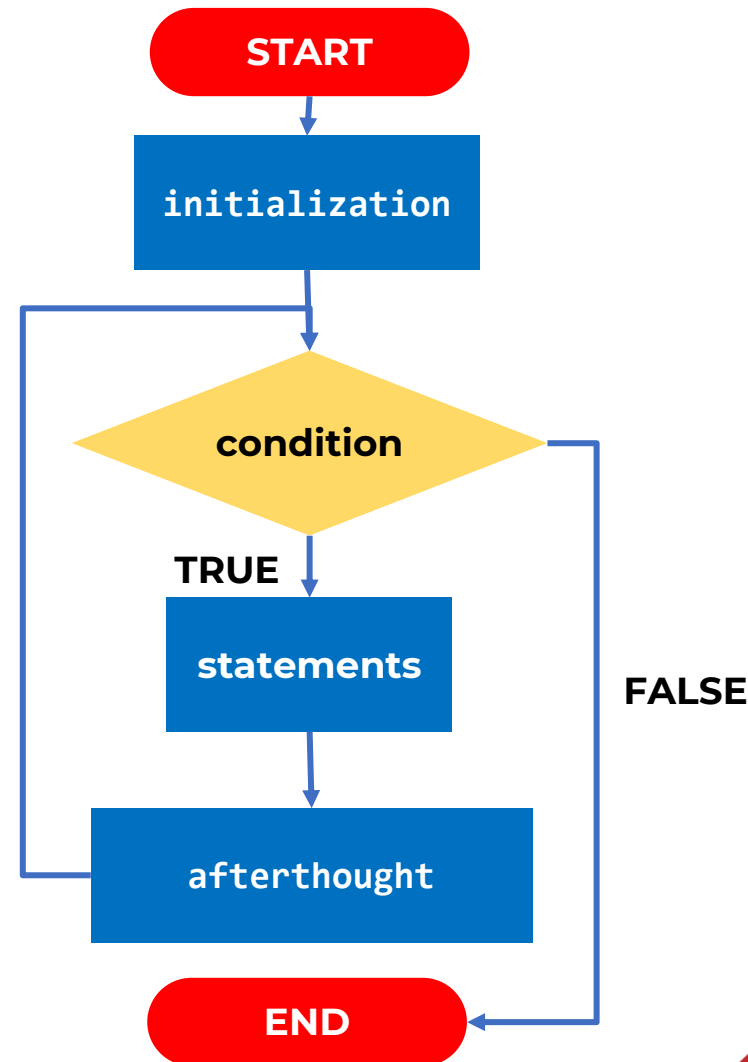
- Thực thi khối lệnh trong vòng lặp **khi điều kiện đúng** (số lần lặp được biết trước)
- **Syntax:**

```
for (initialization; condition; afterthought) {  
    | statement  
}
```

initialization: khối lệnh thực thi 1 lần khi vòng lặp được khởi tạo

condition: Điều kiện kiểm tra mỗi lần lặp trước khi thực thi khối lệnh

afterthought: Câu lệnh thực thi khi kết thúc mỗi lần lặp

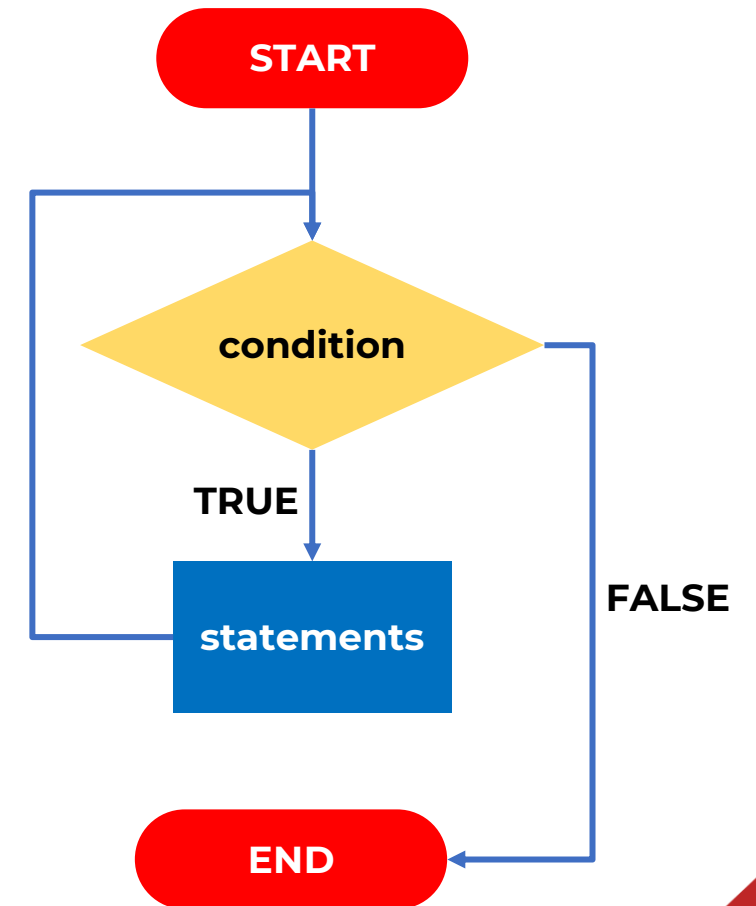


CÂU LỆNH LẶP - 2

- **Vòng lặp while**

- Thực thi khối lệnh trong vòng lặp **khi điều kiện đúng** (số lần chưa được xác định)
- **Syntax:**

```
while (condition) {  
    // statements  
}
```

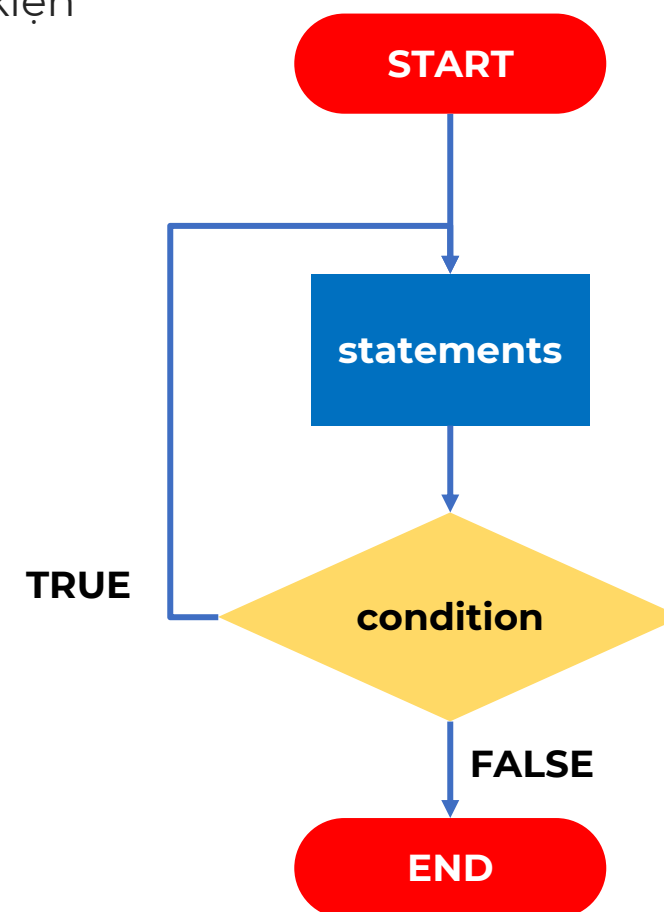


CÂU LỆNH LẶP - 3

- **Vòng lặp do...while**

- Thực thi khối lệnh trong **DO** ít nhất 1 lần, sau đó kiểm tra điều kiện
- Nếu điều kiện đúng, tiếp tục thực thi khối lệnh trong **DO**
- Nếu điều kiện sai, kết thúc vòng lặp
- **Syntax:**

```
do {  
    //Statements  
}  
while (condition)
```



CÂU LỆNH LẶP - 4

- So sánh for, while, do...while

FOR	WHILE	DO...WHILE
Thường dùng khi biết trước số lần lặp	Thường dùng khi chưa biết trước số lần lặp	
Khi điều kiện ban đầu là false thì không thực hiện lần nào		Điều kiện ban đầu là false thực hiện 1 lần
Điều kiện không đúng vòng lặp vô tận		

CÂU LỆNH NHẢY

- Câu lệnh break thoát khỏi switch case và vòng lặp gần nhất
- Câu lệnh continue: dừng vòng lặp tại đó và nhảy sang vòng lặp tiếp theo

```
for (let i = 0; i < 10; i++) {  
  if (i === 3) { break; }  
  text += "The number is " + i + "<br>";  
}
```

```
for (let i = 0; i < 10; i++) {  
  if (i === 3) { continue; }  
  text += "The number is " + i + "<br>";  
}
```



KẾT THÚC

HỌC VIỆN ĐÀO TẠO LẬP TRÌNH CHẤT LƯỢNG NHẬT BẢN