

# Bài: Vòng lặp While trong JavaScript

Xem bài học trên website để ủng hộ Kteam: [Vòng lặp While trong JavaScript](#)

Mọi vấn đề về lỗi website làm ảnh hưởng đến bạn hoặc thắc mắc, mong muốn khóa học mới, nhằm hỗ trợ cải thiện Website. Các bạn vui lòng phản hồi đến Fanpage [How Kteam](#) nhé!

## Dẫn nhập

Ở bài trước, chúng ta đã cùng nhau tìm hiểu về các [câu lệnh điều kiện trong JavaScript](#)

Trong bài này, Kteam sẽ giới thiệu đến các bạn một loại câu lệnh mới: **câu lệnh lặp While**

## Nội dung

Để nắm vững nội dung bài này, bạn cần có kiến thức về

- Biến trong JavaScript
- Các kiểu dữ liệu cơ bản trong JavaScript
- Các toán tử cơ bản trong JavaScript

Những nội dung mà chúng ta sẽ cùng tìm hiểu:

- Đặt vấn đề
- Vòng lặp while
- Vòng lặp do – while
- Lệnh break và continue

## Đặt vấn đề

Vào một ngày đẹp trời không mây, Toàn có một trò chơi nho nhỏ: Thả xúc xắc (súc sắc) cho đến khi xuất hiện 6 hoặc 7 nút.

Trong trường hợp này, Toàn không thể biết trước số lần mình gieo để đạt được mục tiêu mong muốn. Và việc thả xúc xắc sẽ được diễn ra mãi cho đến khi xuất hiện số 6 hoặc 7.

Đối với những công việc được lặp đi lặp lại để thỏa mãn một điều kiện nào đó, thì trong lập trình, nó được gọi là vòng lặp

## Vòng lặp while

Cú pháp:

```
while(<condition>) <statement>
```

Trong đó:

- **<condition>** là một điều kiện, sẽ được đối chiếu trước khi thực hiện **<statement>**, và vòng lặp sẽ tiếp diễn mãi cho đến khi **<condition> = falsy**.
- **<statement>**: là các lệnh (khối lệnh) sẽ được diễn ra khi **<condition> = truthy**.

Kteam xin nhấn mạnh: Vòng lặp sẽ xảy ra mãi cho đến khi **<condition> = falsy**. Khi đó, các lệnh tiếp theo sau vòng lặp sẽ được thực hiện.

Dưới đây là một ví dụ cụ thể: dùng hàm **random** để tạo ngẫu nhiên một con số nằm trong khoảng [0..10], và làm liên tục cho đến khi có một con số nào đó lớn hơn 7 xuất hiện.

**Javascript:**

```
number = 0
// 0
while (number <= 7) {
  console.log(number);
  number = Math.floor(Math.random() * 10); // Sử dụng hàm Math.random
}
// 0
// 0
// 1
// 9
console.log(number);
// 9
// undefined
```

**Lưu ý:** vì chúng ta đang random, nên với mỗi lần chạy, kết quả thu được sẽ khác nhau.

Dưới đây là một số ví dụ khác có dùng while

- **Ví dụ 1:** In ra các số chẵn trong khoảng [0..10]

**Javascript:**

```
var t = 2;
while(t <= 10) {
  console.log(t);
  t += 2;
}
```

- **Ví dụ 2:** Tính tổng các số nguyên nằm trong khoảng [0..100]

**Javascript:**

```
var t = 0,
    Sum = 0;
while (t <= 100) {
  Sum += t;
  t += 1;
}
console.log(Sum);
```

- **Ví dụ 3:** In ra các kí tự có trong một chuỗi

**Javascript:**

```
str = "howKteam"
// 'howKteam'
t = 0
// 0
while(t <= str.length) {
  console.log(str[t])
  t++
}
// h
// o
// w
// K
// t
// e
// a
// m
// undefined
// 8
```

## Vòng lặp do-while

Cú pháp:

```
do <statement> while(<condition>)
```

Trong đó, **<statement>** và **<condition>** giống như đối với **while**, chỉ khác là đối với **<condition>**, việc kiểm tra sẽ diễn ra sau khi các lệnh được thực hiện.

Đối với vòng lặp này, thì các lệnh **<statement>** sẽ được thực hiện ít nhất một lần trước khi kiểm tra **<condition>**.

Về mặt cơ bản, chúng là như nhau. Dưới đây là các ví dụ bên trên khi được viết bằng do-while

- **Ví dụ 1:** In ra các số chẵn trong khoảng [2..10]

JavaScript:

```
var t = 0;
do {
  t += 2;
  console.log(t)
} while(t <= 10)
```

- **Ví dụ 2:** Tính tổng các số nguyên nằm trong khoảng [0..100]

JavaScript:

```
var t = 0, Sum = 0;
do {
  Sum += t;
  t++;
} while(t <= 100)
console.log(Sum)
```

- **Ví dụ 3:** In ra các kí tự trong một chuỗi

JavaScript:

```
str = 'howKteam'  
// 'howKteam'  
t = 0  
// 0  
do {  
  console.log(str[t])  
  t++  
} while(t <= str.length)  
// h  
// o  
// w  
// K  
// t  
// e  
// a  
// m  
// undefined  
// 8
```

## Lệnh break và continue

### Lệnh break

Đối với **break**, thì Kteam đã giải thích tương đối đầy đủ ở bài [Câu lệnh điều kiện trong JavaScript](#), các bạn có thể xem lại.

Trong một vòng lặp, khi đang chạy mà gặp lệnh **break**, thì chương trình sẽ kết thúc luôn vòng lặp mà không cần xem xét điều kiện.

- **Ví dụ:** với chương trình sau, ta chỉ tính được tổng các số trong khoảng [1..50]:

JavaScript:

```
var t = 0, Sum = 0;  
while(t <= 100)  
{  
  if(t > 50) break;  
  Sum += t;  
  t++;  
}  
console.log(Sum) // 1275
```

### Lệnh continue

Khác với break, lệnh **continue** trong vòng lặp while sẽ khiến chương trình bỏ qua các lệnh sau nó, và trực tiếp chuyển đến kiểm tra điều kiện.

- **Ví dụ:** Tính tổng các số trong khoảng [0..100], bỏ qua các số trong khoảng [50..60]:

JavaScript:

```
var t = 0, Sum = 0;  
while(t < 100)  
{  
  t++;  
  if(t >= 50 && t <= 60) continue;  
  Sum += t;  
  console.log(t);  
}  
console.log(Sum);
```

## Kết luận

Qua bài này, các bạn đã được tìm hiểu về vòng lặp với while trong JavaScript

Ở bài sau, các bạn sẽ được tìm hiểu về một dạng vòng lặp khác: [vòng lặp với for](#)

Cảm ơn các bạn đã theo dõi bài viết. Hãy để lại bình luận hoặc góp ý của mình để phát triển bài viết tốt hơn. Đừng quên **"Luyện tập – Thử thách – Không ngại khó"**.

