

# Bài: Biến toàn cục và biến cục bộ trong C#

Xem bài học trên website để ủng hộ Kteam: [Biến toàn cục và biến cục bộ trong C#](#)

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 các bài học trước, chúng ta đã cùng nhau tìm hiểu về [CẤU TRÚC CƠ BẢN CỦA MỘT HÀM](#). Chúng ta đã từng nghe đến biến cục bộ và biến toàn cục.

Vậy **biến cục bộ** và **biến toàn cục** là gì. Chúng ta cùng tìm hiểu ở bài này nhé!

## Nội dung

Để đọc hiểu bài này tốt nhất các bạn nên có kiến thức cơ bản về các phần:

- [BIẾN](#) và [KIỂU DỮ LIỆU](#) trong C#
- [TOÁN TỬ TRONG C#](#)
- [CÂU ĐIỀU KIỆN TRONG C#](#)
- [CẤU TRÚC CƠ BẢN VÒNG LẶP](#)
- [CẤU TRÚC CƠ BẢN CỦA HÀM](#)

Trong bài học này, chúng ta sẽ cùng tìm hiểu các vấn đề:

- Biến toàn cục và biến cục bộ trong C#

## Biến toàn cục và biến cục bộ trong C#

**Biến toàn cục** là biến được khai báo ở phân cấp cao hơn vị trí đang xác định.

**Biến cục bộ** là biến được khai báo ở cùng phân cấp tại vị trí đang xác định.

Vòng đời của biến toàn cục và biến cục bộ bắt đầu khi khối lệnh chứa nó bắt đầu (khối lệnh bắt đầu bằng dấu "{") và kết thúc khi khối lệnh chứa nó kết thúc (khối lệnh kết thúc bằng dấu "}").

**Biến cục bộ** được ưu tiên sử dụng hơn **biến toàn cục** trong trường hợp 2 biến này trùng tên.

## Ví dụ:

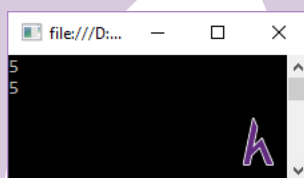
Ví dụ về biến toàn cục

C#:

```
class Program
{
    // biến toàn cục của các hàm nằm trong class Program
    // biến cục bộ của class Program
    static int value = 5;
    static void Main(string[] args)
    {
        // in ra màn hình biến toàn cục
        Console.WriteLine(value);
        PrintSomething();
        Console.ReadKey();
    }

    static void PrintSomething()
    {
        // in ra màn hình biến toán cục
        Console.WriteLine(value);
    }
}
```

Kết quả khi chạy chương trình trên:



Chúng ta thử thay đổi giá trị của biến `value` ngay trước khi gọi hàm `PrintSomething` nhé.

**C#:**

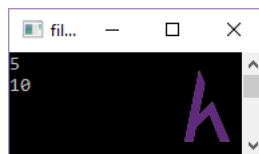
```
class Program
{
    // biến toàn cục của các hàm nằm trong class Program
    // biến cục bộ của class Program
    static int value = 5;
    static void Main(string[] args)
    {
        // in ra màn hình biến toàn cục
        Console.WriteLine(value);

        // thay đổi giá trị của value
        value = 10;

        // kết quả gọi hàm này sẽ in ra màn hình giá trị là 10
        PrintSomething();
        Console.ReadKey();
    }

    static void PrintSomething()
    {
        // in ra màn hình biến toàn cục
        Console.WriteLine(value);
    }
}
```

Kết quả là màn hình in ra giá trị 5 và 10. Do giá trị của biến **value** đã bị thay đổi trước khi gọi hàm **PrintSomething**.



Và giờ chúng ta thử tạo một biến cục bộ ngay bên trong hàm **PrintSomething**.

**C#:**

```
class Program
{
    // biến toàn cục của các hàm nằm trong class Program
    // biến cục bộ của class Program
    static int value = 5;

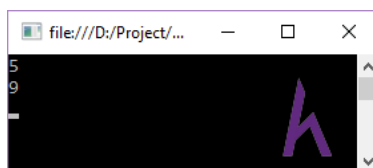
    static void Main(string[] args)
    {
        // in ra màn hình biến toàn cục
        Console.WriteLine(value);

        // thay đổi giá trị của value
        value = 10;

        // kết quả gọi hàm này sẽ không thay đổi vì ưu tiên biến cục bộ hơn
        PrintSomething();
        Console.ReadKey();
    }

    static void PrintSomething()
    {
        int value = 9;
        // in ra màn hình biến toàn cục
        Console.WriteLine(value);
    }
}
```

Kết quả màn hình in ra giá trị 5 và 9. Không in ra giá trị 10 vì **biến cục bộ** được ưu tiên sử dụng hơn. Ở hàm **Main**, thay đổi giá trị của biến **value** thì biến **value** này đang là **biến toàn cục**. Không làm thay đổi giá trị của **biến cục bộ** bên trong hàm **PrintSomething**.



**Lưu ý:**

- Parameter chính là một biến cục bộ.
- **Biến cục bộ** có phạm vi sử dụng bên trong cặp dấu ngoặc nhọn {}.

**C#:**

```
static void Main(string[] args)
{
    bool isTrue = true;

    if (isTrue)
    {
        int value = 5;

        Console.WriteLine(value);
    }
    else
    {
        // không thể sử dụng biến value vì hiện thời biến value là biến cục bộ của câu if true
        // vòng đời của biến cục bộ value chỉ nằm bên trong cặp dấu ngoặc nhọn {}

        Console.WriteLine(value);
    }

    Console.ReadKey();
}
```

## Kết luận

Qua bài này chúng ta đã nắm được biến toàn cục và biến cục bộ là gì.

Bài sau chúng ta sẽ tìm hiểu về [TỪ KHÓA REF VÀ OUT TRONG C#](#)

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**".