

# Buổi 2: Kiểu dữ liệu, biến, toán tử trong C

## Kiểu dữ liệu trong C:

Trong lập trình, một **kiểu dữ liệu** là một cách phân loại dữ liệu cho chương trình hiểu các lập trình viên muốn sử dụng dữ liệu gì.

Ví dụ bạn muốn quản lý tuổi của một người trong chương trình, tuổi sẽ là kiểu **số nguyên** - trong ngôn ngữ lập trình sẽ có kiểu dữ liệu tương ứng để làm việc với số nguyên.

Hầu hết các ngôn ngữ lập trình hỗ trợ nhiều kiểu dữ liệu khác nhau, như số nguyên, số thập phân... và C cũng vậy.

Một kiểu dữ liệu chỉ định kích thước và loại giá trị. Có 6 kiểu dữ liệu được sử dụng nhiều nhất trong C:

Kiểu dữ liệu	Kích thước	Mô tả
<code>int</code>	4 bytes	Lưu trữ toàn bộ số nguyên từ -2,147,483,648 to 2,147,483,647
<code>long</code>	8 bytes	Lưu trữ toàn bộ số nguyên từ -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
<code>float</code>	4 bytes	Lưu trữ số thập phân. Đủ để lưu trữ 6 đến 7 chữ số thập phân sau dấu phẩy
<code>double</code>	8 bytes	Lưu trữ số thập phân. Đủ để lưu trữ 15 chữ số thập phân sau dấu phẩy
<code>char</code>	2 bytes	Lưu trữ một ký tự/chữ cái hoặc giá trị trong bảng mã <u>ASCII</u>

## Biến trong C là gì?

**Biến** nơi lưu trữ tạm thời dữ liệu với một kiểu dữ liệu xác định. Biến có thể thay đổi giá trị trong quá trình thực thi chương trình, do đó được gọi là **biến**.

Trong ngôn ngữ C, một biến sẽ có 3 thành phần:

- Kiểu dữ liệu: chính là phần bên trên chúng ta đã tìm hiểu qua.
- Tên gọi: tên gọi này sẽ giúp ta làm việc với biến đó trong chương trình, ví dụ gọi biến đó ra, tính toán biến đó...
- Giá trị: đó là giá trị mà biến đó nó chứa, ví dụ như `10`.

### **Cách đặt tên biến trong C:**

Trong ngôn ngữ C, mỗi biến có một tên gọi, nhưng không phải muốn đặt tên thế nào tùy theo ý thích của bạn cũng được. Dưới đây là một số nguyên tắc khi đặt tên cho biến:

- Tên biến chỉ có thể chứa chữ cái (viết hoa hoặc viết thường), chữ số và ký tự `_` (gạch dưới).
- Tên biến phải bắt đầu bằng một chữ cái.
- Không được sử dụng khoảng trắng hoặc các ký tự đặc biệt như `!`, `@`, `#`, ...
- Bạn cũng không được phép sử dụng chữ cái mang dấu trọng âm. (ví dụ `é` `ê`...).
- Tên biến không được trùng với những từ khóa trong C.
- Và một điều hết sức quan trọng mà bạn cần phải nhớ đó là trong ngôn ngữ C có sự khác nhau giữa chữ thường và chữ in hoa: `chieu_rong`, `CHIEU_RONG` và `Chieu_RoNg` là tên của 3 biến khác nhau trong ngôn ngữ C.

### **Vậy đặt tên biến như thế nào cho đúng?**

- Hãy sử dụng tiếng anh để đặt tên cho biến sẽ làm cho code dễ đọc và hiểu.
- Tên của biến phải bắt đầu bằng một chữ cái viết thường, nếu tên biến có 2 từ trở lên, hãy viết hoa chữ cái đầu tiên của từ tiếp theo hoặc sử dụng `.` Ví dụ: `phoneNumber` hoặc `phone_number` ...
- Tên biến cần tường minh, có nghĩa là tên biến phải thể hiện được ý nghĩa rõ ràng đúng với mục đích nó được khai báo. Ví dụ: `name` thay vì chỉ `n`, `age` thay vì chỉ `a` ...

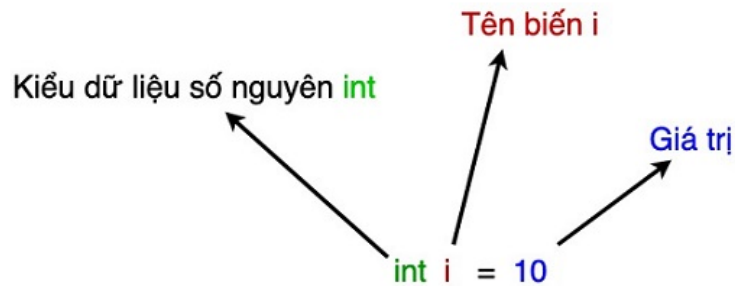
### **Khai báo và khởi tạo biến trong C:**

Cú pháp để khai báo 1 biến như sau:

```
kiểu dữ liệu  tên biến;
```

hoặc chúng ta có thể vừa khai báo vừa gán luôn giá trị cho biến như sau:

kiểu dữ liệu   tên biến   =   giá trị;



### Số nguyên:

Đối với kiểu số nguyên thì không có gì đặc biệt, các bạn hãy xem qua ví dụ sau:

```
#include <stdio.h>
int main() {
    int a = 2; // gán giá trị cho biến tại thời điểm khai báo
    int b;
    b = 5;     // gán giá trị sau khi khai báo biến
}
```

### Số thập phân

Đối với kiểu số thập phân ( `float` và `double` ) cần lưu ý khi khai báo và gán giá trị cho biết, trong lập trình chúng ta sẽ sử dụng dấu chấm `.` thay vì dấu phẩy `,` - chúng ta hay dùng cụm từ `dấu chấm động` để nói đến các số thập phân.

```
#include <stdio.h>
int main() {
    float a = 2.75; // dùng dấu .
    float b = 5,55; // dùng dấu , sẽ bị báo lỗi
}
```

### Kiểu ký tự

Đối với kiểu ký tự, giá trị cần được đặt trong cặp dấu **nháy đơn** `' '` để chương trình hiểu được đó là một ký tự.

```
#include <stdio.h>
int main() {
```

```
char c = 'A';  
}
```

Nếu kiểu ký tự bạn cố tình khai báo nhiều hơn 1 ký tự trong `char`, biến sẽ chỉ nhận 1 ký tự cuối cùng.

```
#include <stdio.h>  
int main() {  
    char c = 'ABC';  
    printf("%c", c);  
}
```

Kết quả

C

Khi bạn khai báo mà không gán giá trị cho biến, lúc này biến sẽ chứa một giá trị không xác định.

### Hằng số trong C

Nếu bạn muốn khởi tạo một biến có giá trị **không thể thay đổi**, bạn có thể sử dụng 2 cách sau:

- Sử dụng bộ tiền xử lý `#define`.
- Sử dụng từ khóa `const`.

Điều này sẽ tạo ra một **hằng số**.

Ví dụ với bộ tiền xử lý `#define`:

```
#include <stdio.h>  
#define PI 3.14  
int main() {  
    printf("%f", PI);  
}
```

Ví dụ với từ khóa `const`:

```
#include <stdio.h>
int main() {
    const double PI = 3.14;
    PI = 4.5; // dòng này sẽ bị lỗi
    printf("%f", PI);
}
```

- Khi bạn gán lại giá trị `PI = 4.5;` thì dòng này sẽ bị lỗi vì `PI` đã được khai báo với từ khóa `const`, lúc này `PI` được xem là một hằng số, mà một hằng số thì không thể thay đổi được giá trị (hay gán lại một giá trị mới).
- Nên dùng hằng số khi bạn muốn một biến không thể thay đổi giá trị do ngẫu nhiên hay sơ ý của bản thân hoặc không mong muốn người khác sơ ý hay cố ý thay đổi giá trị biến của bạn, điều này có thể dẫn đến chương trình chạy sai hoặc bị lỗi.
- Quy tắc đặt tên biến hằng số hơi khác với quy tắc đặt tên biến thông thường đó là sẽ viết hoa tất cả ký tự tên biến, nếu tên biến hằng số có nhiều hơn 1 từ, thì mỗi từ nên được ngăn cách bằng dấu gạch dưới `_`. Ví dụ: `PI`, `NEW_LINE` ...

## ***Phạm vi biến trong C***

Trong ngôn ngữ lập trình C, mọi biến khi khai báo đều có 2 thuộc tính: kiểu dữ liệu (type) và lớp lưu trữ (storage class) của nó. Lớp lưu trữ ở đây chính là thuộc tính thể hiện phạm vi của biến - nơi nào có thể dùng biến đó và biến đó tồn tại trong bao lâu. Có 2 loại phạm vi chính cần quan tâm như sau:

- Global variables (biến toàn cục)
- Local variables (biến cục bộ)

## ***Biến toàn cục trong C***

Các biến được khai báo ở bên ngoài tất cả các khối code của hàm main thì được gọi là biến toàn cục (external or global variable). Các biến toàn cục có thể sử dụng ở mọi hàm - hay mọi nơi trong chương trình. Biến toàn cục cũng tồn tại cho tới khi chương trình kết thúc.

Ví dụ:

```
#include <stdio.h>
int n = 7; // global variable
```

```
int main() {  
    printf("%d", n);  
    return 0;  
}
```

Kết quả

7

## Biến cục bộ trong C

Các biến được khai báo trong thân một hàm thì được gọi là các biến cục bộ (local variables). Các biến cục bộ này chỉ tồn tại và chỉ có thể sử dụng bên trong hàm nó được khai báo.

Thân hay body của một hàm được xác định nằm trong cặp dấu ngoặc nhọn `{ }`.

Ví dụ:

```
#include <stdio.h>  
int main() {  
    int a;  
    double b;  
    return 0;  
}
```

## Độ ưu tiên phạm vi biến

Nếu chúng ta sử dụng cùng một tên biến `global variable` và `local variables` thì biến `local variables` sẽ được ưu tiên hơn.

Ví dụ:

```
#include <stdio.h>  
int a = 5;    // global variable  
int main() {  
    int a = 1; // local variables  
    printf("%d", a);  
}
```

Kết quả

1