

Const修飾子

@arbuztw

先來個例子

- 寫一個函數計算陣列的和

```
int sum(int arr[], int n) {  
    for (int i = 1; i < n; i++)  
        arr[i] += arr[i-1];  
    return arr[n-1];  
}
```

先來個例子

- 寫一個函數計算陣列的和

```
int sum(int arr[], int n) {  
    for (int i = 1; i < n; i++)  
        arr[i] += arr[i-1];  
    return arr[n-1];  
}
```

- 如果之後還要用到arr的值

```
int arr[5] = {1, 2, 3, 4, 5};  
int s = sum(arr, 5);  
  
// ... do something ...  
  
std::cout << arr[4] << std::endl;    // 15
```

先來個例子

- 寫一個函數計算陣列的和

```
int sum(int arr[], int n) {  
    for (int i = 1; i < n; i++)  
        arr[i] += arr[i-1];  
    return arr[n-1];  
}
```

- 如果之後還要用到arr的值

```
int arr[5] = {1, 2, 3, 4, 5};  
int s = sum(arr, 5);  
  
// ... do something ...  
  
std::cout << arr[4] << std::endl;    // 15
```

- 悲劇!

使用const

- 當code變的複雜的時候該如何debug?

使用const

- 當code變的複雜的時候該如何debug?
- 讓編譯器幫我們檢查

```
int sum(const int arr[], int n) {  
    //      ^^^^^  
    for (int i = 1; i < n; i++)  
        arr[i] += arr[i-1];  
    return arr[n-1];  
}
```

const語法

- const代表constant(常數)
- 放在變數宣告代表之後不能再改它的值

```
const double PI = 3.1415926535;  
double const PI = 3.1415926535;
```

- 放在函數參數代表不能在函數中改變它的值

```
int func(const int arr[], ...);
```

- `const int arr[]` 可以讀作 array of const int.

const使用注意

- const 變數一定要給初始值
- const 變數名稱常使用全大寫