



C++ 语法糖

By 吴羽晗.

简介

语法糖 (Syntactic sugar) 指在编程语言中添加的便于使用的语法。它不改变语言功能，但能提高代码可读性，让程序员少写 bug。

- 部分功能需要开启 C++11。 [See how to enable support for C++11.](#)

一、 `auto` 关键字

声明变量时用 `auto` 代替类型名，让编译器自动推导变量类型。

示例 1

```
auto x = 10;    // 鉴定为 int
auto y = 3.14;  // 鉴定为 double
auto z = 'A'    // 鉴定为 char
```

示例 2

```
std::string s;

auto p = s.begin();    // 鉴定为 std::string::iterator
```

示例 3

```
float solve() {
    ...
}

int main() {
```

```
    auto result = solve();  
    // 若 solve() 的类型发生变更, 此处不必跟着变更  
  
    ...  
}
```

二、范围 for 循环

范围 for 循环 (Range-based for Loop) 可以更简单地遍历顺序容器中的每一个元素。

支持数组, `vector`, `deque`, `list`, `string` 等支持顺序访问的容器。

示例 1

```
#include <iostream>  
  
int main() {  
  
    int a[] = {1, 1, 4, 5, 1, 4};  
  
    for (int el : a) {  
        std::cout << el << ' ';  
    }  
  
    return 0;  
}
```

1 1 4 5 1 4

示例 2

将数组写进 for 循环, 再配合 `auto` 关键字, 美滋滋。

```
#include <iostream>  
  
int main() {
```

```

    for (auto el : {1, 1, 4, 5, 1, 4}) {
        std::cout << el << ' ';
    }

    return 0;
}

```

```
1 1 4 5 1 4
```

示例 3

```

#include <iostream>
#include <string>

std::string toUpperCase(std::string s) {

    for (auto& ch : s) // 不能漏加「&」，否则变更不会保存到 s 中
        ch = toupper(ch);

    return s;
}

int main() {

    std::cout << toUpperCase("hello, world");

    return 0;
}

```

```
HELLO, WORLD
```

三、使用 `using` 定义别名

看上去比 `typedef` 更优美。

基础用法

```
using TypeB = TypeA;
```

示例 1

```
using int32 = long;
using int64 = long long;
using uint32 = unsigned long;
using uint64 = unsigned long long;

const int64 Mod = 998244353;

...
```

示例 2

通过 `using` 可以更方便地定义模板类的别名。

```
#include <map>

template <typename T>
using intTo = std::map<int, T>;

int main() {

    intTo<int> map1;
    intTo<char> map2;
    intTo<double> map3;

    ...

}
```

四、Lambda 表达式

Lambda 表达式允许我们在任何地方定义函数，甚至是在函数中定义函数。

基础用法

```
auto funcName = [&](Type1 x1, Type2 x2, ...) -> returnType {  
    ...  
};
```

- Lambda 表达式是一个语句，不能漏掉最后的分号；
- 当采用 `[&]` 时，Lambda 函数可以修改外部的变量，采用 `[=]` 时则不行；
- `-> returnType` 可省略。

示例 1

```
#include <iostream>  
#include <string>  
  
int main() {  
  
    auto toUpperCase = [=](std::string s) -> std::string {  
        for (auto& ch : s)  
            ch = toupper(ch);  
        return s;  
    };  
  
    std::cout << toUpperCase("hello, world");  
  
    return 0;  
}
```

HELLO, WORLD

示例 2

```
#include <iostream>  
#include <vector>  
#include <algorithm>  
  
int main() {  
  
    std::vector<int> vec = {1, 1, 4, 5, 1, 4};
```

```
std::sort(vec.begin(), vec.end(), [&](int l, int r) {  
    return l > r;  
});  
  
for (auto el : vec) {  
    std::cout << el << ' ';  
}  
  
return 0;  
}
```

5 4 4 1 1 1