

# 让我们用中文编程（二）之C++篇 - 自己做中文编程语言，你也行

---

让我们用中文编程（二）之C++篇 - 自己做中文编程语言，你也行

作者：Dancefire (<http://www.dancefire.org/>) (Dancefire [at] gmail)

日期：2005年11月11日

由于CSDN的Blog好像对表格支持不好，我贴了几次，都导致了重复的内容。所以，为了能够贴到这里，我把表格都改变了，所以，推荐大家访问没有改过格式的连接：<http://www.dancefire.org/Project/SinoProgram/> (<http://www.dancefires.com/Project/SinoProgram/>)

## 引言

上回书说到，用C#来进行中文编程，最后的主程序上很是有趣，没看过的客官，可以看看第一篇《让我们用中文编程 (<http://blog.csdn.net/DanceFire/archive/2004/10/04/124805.aspx>)》。看到有些客官，在后面说到，如果连那些关键字也能够中文化就好了。当时心中无奈，唉，谁叫C#不支持#define来替换关键字呢？在这风云变幻的年代，每一分钟都有奇迹的发生，曾经的“不可能”，很有可能就变成了“想当然”。当初的无奈，现在还是无奈么？请看下文。

## 中文编程的神化

随着11月7日，微软Visual C++ 2005的发布，我们关注到，VC8.0中增加了大量新的特性，包括最重要的，引入了C++/CLI，可以说是给C/C++语言历史上掀起了新的一页。作为一个方言的出现C++/CLI会有什么样的作为，却是值得大家拭目以待，是功是过，由历史来评说。但是我们也注意到，VC8.0引入了一个非常有趣的支持，VC8.0开始支持Unicode的源代码了。如同C#和Java一样。如今，我们要写C++程序，已经完全允许在代码中出现Unicode编码的变量名、函数名、类名等等，并且#define也支持Unicode替换了。

虽然是个小变化，但是，中文编程的世界一下就变了。这使我立刻就想到，国内有些公司，在致力于炒作中文编程，细看他们的语言，无非是对现有语言把关键字替换成了中文而已，其他的改进太小了。有的是把Forth给用中文关键字包装了一下，就推出来的；还有的是把C语言用中文的关键字替换一下就推出来的。这样的语言，真的是自己的么？这样的语言真的有价值么？“决定一切的不是语言而是思想！”，这是已经人人共知的思想，国内的某些专家和公司竟然还在为这种“伪中文”编程而津津乐道，甚至有些获得了国家项目的资助。如果真的仅仅是用中文关键字来进行编程就可以改变我们国家的IT面貌的话，那大可不必投资数十万甚至上百万来做，用我下面的一个头文件就够了。

关键字使用中文，很简单的技术；实现一个别人已经发明的语言，也不是什么太复杂的事情（当然如果做一个100%兼容的C++这样的复杂语言的编译器还是具有相当的难度的）。真正难的应该是理解这种语言的创新思想，发现或者创造一种方法，以解决现有语言在程序设计上的存在的种种问题，这样更新或者创造出来的才是真正有用的经典之作。不用细说，大家都知道，C、C++、Java、C#、Ruby、Python的诞生都分别解决了什么难题。他们用漂亮的办法解决了一直困扰程序员的问题，他们就成了经典。而国内那些做中文编程的公司的产品呢？呵呵，也许仅仅是茶余饭后而笑柄而已。

不好意思，一时感慨颇多，啰里啰唆了那么多废话。下面言归正传。。。

## 没有什么不可以

再细说之前，需要说一下编译器的事情。

目前我不知道还有哪个编译器能够对Unicode有良好的支持，所以，目前只有Visual C++ 2005才能够编译下面的代码，但是很幸运，原本打算卖US\$49的Visual C++ 2005 Express，从现在开始，一年内是免费的，而且微软没有一年以后的收费计划，所以也许我们可以一直合法的用下去，呵呵。大家可以从微软网站直接下载，需要下载请访问下面的连接：

<http://msdn.microsoft.com/vstudio/express/visualc/> (<http://msdn.microsoft.com/vstudio/express/visualc/>)

下面的代码都可以在Visual C++ 2005 Express 版上编译通过。

好了，真的言归正传了。我用了一系列的#define宏和typedef，作了中文和英文关键字的替换。比如：

```
typedef int    整型;
typedef char   字符型;

#define 如果   if
#define 否则   else
#define 返回   return
#define 常量   const

#define 标准输出 std::cout
#define 行结束   std::endl
#define 标准输入 std::cin
#define 字符串   std::string

#define 主程序  main
```

经过这样处理后我们就可以写出一个最简单的中文程序：

整型 主程序( 整型 参数数量, 常量 字符型\*\* 参数列表 )

```
{
    字符串 路人甲;
    整型  年龄;

    标准输出 << "请输入姓名：" << 行结束;
    标准输入 >> 路人甲;
    标准输出 << "请输入年龄：" << 行结束;
    标准输入 >> 年龄;

    如果(年龄 > 18){
        标准输出 << 路人甲 << "先生/女士，您好！祝您今天工作愉快哦。" << 行结束;
    }否则{
        标准输出 << 路人甲 << "小朋友，您好！祝你今天玩得开心。" << 行结束;
    }

    返回 0;
}
```

程序很简单，执行后，询问你的姓名和年龄，如果你大于18岁呢，它就要对你说“Dancefire先生/女士，您好！祝您今天工作愉快哦。”，如果还不到18岁呢，就会说“Dancefire小朋友，您好！祝你今天玩得开心。”

这个程序似乎不够有趣。那么我再来一个有趣一些的，用到了面向对象的继承的概念，新瓶装旧酒了，前一篇文章中提到的，生物圈的例子，（既然是中文嘛，就来点中国文化特色，呵呵），

哦，多说一句，下面的程序一共由3个文件组成，其中2个是头文件，另外一个为源文件。三个文件都可以从我的网站下载：

<http://www.dancefire.org/file/SinoCPP.zip> (<http://www.dancefire.org/file/SinoCPP.zip>)

如果大家想自己写个中文的类玩玩，只需要#include第一个Sino.h就可以了。我已经在里面替换了大量的关键字，玩玩够了，如果还想替换其他关键字，你们自己加进去就行了，需要注意的是，这几个文件都是受GPL协议保护的，简单而言就是，大家可以随便改，只要继续保持开放就行了。

下面是主程序：

```
//
// 作者: Dancefire (http://www.dancefire.org/) (Dancefire[at] gmail)
// 日期: 公历 二零零五年十一月十一日
//      农历 乙酉年 丁亥月 己亥日 巳时
//      宜 解除 修饰垣墙 冠笄 出行 馊事勿取
//      忌 开市 动土 破土 嫁娶 安葬
// 项目: 中文C++编程测试
// 版本: 甲子
//

#include <vector>
#include <iostream>
#include <string>
#include "sino.h"
#include "SinoTest.h"
```

使用 名字空间 标准库;

整型 主程序( 整型 参数数量, 常量 字符型\*\* 参数列表 )

```
{
    标准输出 << "世界你好! " << 行结束 << 行结束;
```

使用 名字空间 中文编程;

**矢量**<生物\*> 动物们;

```
动物们.排到后面( 新建 鸟());
动物们.排到后面( 新建 鱼());
动物们.排到后面( 新建 马());
动物们.排到后面( 新建 动物());
动物们.排到后面( 新建 生物());
```

循环(无符号整型 甲 = 0 ; 甲 < 动物们.多少() ; ++甲)

```
{
    生物* 小家伙 = 动物们[甲];
    小家伙->自我介绍();

    动物* 小动物 = 动态转型<动物*>(小家伙);
    如果( 小动物 ){
        小动物->移动();
    }
    标准输出 << 行结束;
}
返回 0;
}
```

主程序很简单，先建立了一个**动物们**的容器，然后一个一个新的动物加入到这个容器（队列）中了。（当然严格来说，应该是**生物们**，但是听起来好怪，就叫**动物们**好了。）这些动物们排好队后，开始一个接着一个的自我介绍，如果是**动物**呢，就动一动，如果不是**动物**就算了。

执行的结果是这样的：

世界你好！

我属于动物, 我是一只可爱的鸟  
这只鸟在自由自在的飞翔

我属于动物, 我是一条可爱的鱼  
这条鱼在自由自在的游来游去

我属于动物, 我是一匹可爱的马  
这匹马在自由自在的奔跑

我属于生物, 我是一只可爱的动物  
这只动物在自由自在的移动

我是...咦？我是什么？我也不知道我是什么，难道我是一个外星的生物

大家可以在程序中感受到C++的OO的概念。这里面有5个类，**动物**类继承了**生物**类能够自我介绍的能力，**鸟**、**鱼**、**马**继承了**动物**类能够移动的能力，而他们在继承自我介绍的时候，用自己的方式替代(重写)了原始**生物**的自我介绍方式，此处自然用到了虚函数，既然是表现继承，怎么能没了虚函数呢，是吧。在判断是否是动物的时候，用到了C++标准的动态转型，这是在运行期，而不是编译期，进行转型的。类的定义在头文件SinoTest.h中，完整的这个头文件如下：

```
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 作者: Dancefire (http://www.dancefires.com/) (Dancefire [at] gmail)
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 日期: 公历 二零零五年十一月十一日
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 农历 乙酉年 丁亥月 己亥日 巳时
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 宜 解除 修饰垣墙 冠笄 出行 馊事勿取
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 忌 开市 动土 破土 嫁娶 安葬
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 项目: 中文C++编程测试
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif) // 版本: 甲子
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/None.gif)名字空间 中文编程
□ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedBlock.gif){
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) // 生物类
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 类 生物
□ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)公有:
□ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) 生物() {
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 类别 = "生物";
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 祖先的类别 = "游离态";
└ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)
□ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) 虚~生物(){}
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 虚 字符串 我的类别()
□ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 返回 类别;
└ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 虚 字符串 我祖先的类别()
□ (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
```

```

| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 返回 祖先的类别;
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 虚 无值 自我介绍()
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 标准输出 << "我是...噢? 我是什么? 我也不知道我是什么, 难道我是一个外星的" << 我的类别() << 行结束;
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)受保护的:
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 字符串 类别;
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 字符串 祖先的类别;
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) };
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) // 动物类, 动物当然是生物啦
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 类 动物: 公有 生物
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)公有:
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) 动物() {
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 移动方式 = "移动";
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 量词 = "只";
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 类别 = "动物";
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 祖先的类别 = "生物";
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) 虚 无值 移动(){
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 标准输出 << "这" << 量词 << 我的类别() << "在自由自在的" << 移动方式 << 行结束;
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) 虚 无值 自我介绍() {
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 标准输出 << "我属于" << 我祖先的类别() << ", 我是一" << 量词 << "可爱的" << 我的类别() << 行结束;
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)受保护的:
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 字符串 移动方式;
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 字符串 量词;
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) };
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) // 鸟类, 鸟是动物咯。
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 类 鸟: 公有 动物
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)公有:
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoft.com/Images/OutliningIndicators/ContractedSubBlock.gif) 鸟(){
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 移动方式 = "飞翔";
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 量词 = "只";
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 类别 = "鸟";
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 祖先的类别 = "动物";
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) }
└ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif) };
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) // 鱼也是动物咯
| (http://www.actiprosoft.com/Images/OutliningIndicators/InBlock.gif) 类 鱼: 公有 动物
┌ (http://www.actiprosoft.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)

```

```
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)公有:
␣(http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif)    鱼(){
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)        移动方式 = "游来游去";
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)        量词 = "条";
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)        类别 = "鱼";
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)        祖先的类别 = "动物";
| (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif)    }
| (http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockEnd.gif)    };
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) // 马也是动物
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif) 类 马 : 公有 动物
␣(http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif)
(http://www.actiprosoftware.com/Images/OutliningIndicators/ContractedSubBlock.gif) {
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)公有:
| (http://www.actiprosoftware.com/Images/OutliningIndicators/InBlock.gif)    马()
␣(http://www.actiprosoftware.com/Images/OutliningIndicators/ExpandedSubBlockStart.gif) (undefined) 版权声明：本文来自CSDN，感谢博主原创文章，
遵循 CC 4.0 by-sa 版权协议，转载请附上原文出处链接和本声明。
原文链接：https://blog.csdn.net/DanceFire/article/details/527736
站方申明：本站部分内容来自社区用户分享，若涉及侵权，请联系站方删除。
```

🕒 发表于 2020-04-19 10:44:05    阅读 ( 544 )    分类： (https://www.coder55.com/articles/)

0 推荐

收藏

你可能感兴趣的文章

精选的优质文章

- 也许 Go 开发可以更简单！ (https://www.coder55.com/article/5948?f=featur) 7424 浏览
- 如何使用 Golang 日志监控你的应用程序？ (https://www.coder55.com/article/1079?f=featur) 8074 浏览
- 从Go语言实现模板设计模式浅谈Go的抽象能力 (https://www.coder55.com/article/23350?f=featur) 9406 浏览
- 阿里云基于 Go 的微服务架构分享 (https://www.coder55.com/article/3380?f=featur) 20906 浏览
- java是否会被取代？ Go会否给Java带来冲击？ (https://www.coder55.com/article/495?f=featur) 25676 浏览
- 千万级规模高性能、高并发的网络架构经验分享 (https://www.coder55.com/article/4469?f=featur) 27095 浏览
- 阿里部分面试题汇总,对想进阿里的同学非常实用 (https://www.coder55.com/article/1?f=featur) 58983 浏览
- 实用好文：知乎实时数仓架构实践及演进 (https://www.coder55.com/article/3117?f=featur) 27586 浏览
- 支撑马蜂窝「双11」营销大战背后的技术架构 (https://www.coder55.com/article/15712?f=featur) 226165 浏览
- 想进大厂？ 50个多线程面试题，你会多少？（一） (https://www.coder55.com/article/3071?f=featur) 19221 浏览

0 条评论

请先 登录 (https://coder55.com/login) 后评论

官方社群



关注公众号

—— 加入社区微信群 ——

→ 「Go语言教程」领取 (http://xueyuan.coder55.com/read/go-senior-learn/go-senior-learn-index)

▼ 1. Go语言简介
▼ 2. Go语言基本语法
▼ 3. Go语言数据结构
▼ 4. Go语言流程控制
▼ 5. Go语言函数
▼ 6. Go语言结构体
▼ 7. Go语言接口
▼ 8. Go语言包
▼ 9. Go语言并发
▼ 10. Go语言反射
▼ 11. Go语言文件处理
▼ 12. Go语言编译和工具链
▼ 13. Go语言进阶
▼ 14. 常见面试题

推荐文章

猜你喜欢

---

## 随便看看

JavaScript面试的完美指南(开发者视角) ([https://coder55.com/article/17972?f=r\\_rand](https://coder55.com/article/17972?f=r_rand))

VUE CLI 3.0 安装及创建项目 ([https://coder55.com/article/17971?f=r\\_rand](https://coder55.com/article/17971?f=r_rand))

JavaScript中的Array对象 ([https://coder55.com/article/17970?f=r\\_rand](https://coder55.com/article/17970?f=r_rand))

使用PixiJS做一个小游戏 ([https://coder55.com/article/17969?f=r\\_rand](https://coder55.com/article/17969?f=r_rand))

[VUE ERROR] Invalid options in vue.config.js: "publicPath" is not allowed ([https://coder55.com/article/17968?f=r\\_rand](https://coder55.com/article/17968?f=r_rand))

VUE CLI 3.0 项目引入 ElementUI ([https://coder55.com/article/17967?f=r\\_rand](https://coder55.com/article/17967?f=r_rand))

VUE CLI 3.0 项目引入 Mock.js ([https://coder55.com/article/17966?f=r\\_rand](https://coder55.com/article/17966?f=r_rand))

go.js remove 特定part ([https://coder55.com/article/17965?f=r\\_rand](https://coder55.com/article/17965?f=r_rand))

Node 框架接入 ELK 实践总结 ([https://coder55.com/article/17964?f=r\\_rand](https://coder55.com/article/17964?f=r_rand))

js提交表单错误: document.form.submit() is not a function ([https://coder55.com/article/17963?f=r\\_rand](https://coder55.com/article/17963?f=r_rand))





Go语言中文社区 (<https://www.coder55.com>) | 联系我们 (<https://www.coder55.com/article/33237>) | 官方标准库  
(<https://studygolang.com/static/pkgdoc/main.html>) | 京ICP备18055895号-2 (<https://beian.miit.gov.cn>)

欢迎关注Go语言社区官方微信公众号，商务合作联系QQ：124096702

Powered By Release 20180601 ©2009-2022 coder55.com