

计算机语言编程规范

版本：**v0.8**

Crifan Li

摘要

本文主要介绍了各种计算机语言编程规范的通用知识和某个特定语言的编程规范。



本文提供多种格式供：

在线阅读	HTML ¹	HTMLs ²	PDF ³	CHM ⁴	TXT ⁵	RTF ⁶	WEBHELP ⁷
下载（7zip压缩包）	HTML ⁸	HTMLs ⁹	PDF ¹⁰	CHM ¹¹	TXT ¹²	RTF ¹³	WEBHELP ¹⁴

HTML版本的在线地址为：

http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/html/lan_coding_rule.xml.html

有任何意见，建议，提交bug等，都欢迎去讨论组发帖讨论：

http://www.crifan.com/bbs/categories/lan_coding_rule.xml/

修订历史

修订 0.8

2015-03-27

crl

1. 介绍各种计算机语言的通用的编程规范的来由和特性
2. 列举不同语言的编程规范

¹ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/html/lan_coding_rule.xml.html

² http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/htmls/index.html

³ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/pdf/lan_coding_rule.xml.pdf

⁴ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/chm/lan_coding_rule.xml.chm

⁵ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/txt/lan_coding_rule.xml.txt

⁶ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/rtf/lan_coding_rule.xml.rtf

⁷ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/webhelp/index.html

⁸ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/html/lan_coding_rule.xml.html.7z

⁹ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/htmls/index.html.7z

¹⁰ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/pdf/lan_coding_rule.xml.pdf.7z

¹¹ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/chm/lan_coding_rule.xml.chm.7z

¹² http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/txt/lan_coding_rule.xml.txt.7z

¹³ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/rtf/lan_coding_rule.xml.rtf.7z

¹⁴ http://www.crifan.com/files/doc/docbook/lan_coding_rule.xml/release/webhelp/lan_coding_rule.xml.webhelp.7z

计算机语言编程规范:

Crifan Li

版本 : **v0.8**

出版日期 2015-03-27

版权 © 2015 Crifan, <http://crifan.com>

本文章遵从 : [署名-非商业性使用 2.5 中国大陆\(CC BY-NC 2.5\)](#)¹⁵

¹⁵ http://www.crifan.com/files/doc/docbook/soft_dev_basic/release/html/soft_dev_basic.html#cc_by_nc

目录

正文之前	iv
1. 目的	iv
2. 声明	iv
1. 什么是编程规范以及为何要有编程规范	1
1.1. 编程规范指的是什么	1
1.2. 为何会有编程规范	1
2. 通用的编程规范	3
2.1. 编程规范之风格	3
2.1.1. 风格统一	3
2.2. 编程规范之注释	3
2.2.1. 添加注释的原则	3
2.2.2. 注释的语法	4
2.3. 编程规范之命名	4
2.3.1. 命名的原则	4
2.3.1.1. 见名知意	4
2.3.2. 命名方法	4
2.3.2.1. 驼峰（式）命名法	4
2.3.2.2. 匈牙利命名法	5
2.3.3. 命名规范中的一些细节	5
2.3.3.1. 常量大写	5
2.3.3.2. 根据情况给名字加前缀	5
2.4. 编程规范之设计原则	5
2.4.1. 高内聚，低耦合	6
2.4.2. 使用频率高的代码，尽量做成函数和类库	6
2.4.3. 嵌套层次不要太深	6
2.5. 不同语言自己的特定的编程规范	6
2.6. 不同公司或组织的编程规范	7
2.6.1. Google的编程规范	7
2.6.2. 华为的编程规范	7
3. C语言编程规范	8
4. C++语言编程规范	9
5. Python语言编程规范	10
6. GO语言编程规范	11
7. Java语言编程规范	12
7.1. Java语言编程规范	12
7.2. Android编程规范	12
8. C#语言编程规范	13
9. PHP语言编程规范	14
10. JavaScript语言编程规范	15
11. 其他语言的编程规范	16
11.1. HTML编程规范	16
11.2. CSS编程规范	16
参考书目	17

正文之前

1. 目的

本文旨在介绍不同编程语言所通用的编程规范，以及每个特定的语言所自己特有的编程规范。

以便编程人员重视编程规范，学会遵守合理的编程规范，写出更高质量的代码。

2. 声明

任何问题、意见、建议等，都欢迎发邮件一起探讨：admin (at) crifan.com。

第 1 章 什么是编程规范以及为何要有编程规范

1.1. 编程规范指的是什么

编程规范，指的是一组规则或指导意见，建议你，要求你，在编程时，用某种计算机语言写代码时，如何写，要遵循什么意见或建议。

一般来说，不是强制的，但是是多数人都遵守的一些规范。

不过，很多公司，倒是强制性的，使用统一的某种规范，此时，此规范或约定，就是要强制性的实行了。

在讨论编程规范时，另外还有几个常提到的概念，在此处可以理解基本上是同一个意思：

编程约定Coding Convention(s)

==编程规范Coding Rule(s)

==编程风格Coding Style(s)

==编码风格Programming Style(s)

==编码标准Coding Standard(s)

1.2. 为何会有编程规范

TODO：抽空把那张照片的内容，整理过来：

TODO：另外再顺带搜一下，该照片中的内容的原始出处。

软件，生命周期，代码可读性和可维护性，其中软件的生命周期的大部分，都是别人负责看的代码，维护你的代码，扩展你的代码。

即，代码写出来主要是给别人看的，而不仅仅是给自己看的。

注意此处的“主要”，意味着，你写了代码，是为了后来人，维护此软件的人，看的。

可以总结成简单的一句话：

你写出来的代码是给别人看的

背后的逻辑是：

->所以你要确保你自己写出来的代码，不仅仅是自己容易看懂，且也要确保别人也容易看懂

->对于每个写代码的人，如果都按照自己的喜好和习惯去写代码，那么写出来的代码的风格肯定大相径庭，风格迥异

->所以要有有一个，相对来说通用的，合理的，大家所习惯遵守的风格，总结和定义出来，让大家去遵守

->每个人都在写代码时都遵守这个规范

->这个描述了该如何写代码的规范，就叫做：编程规范

解释了道理后，再贴上一些大牛们的观点：

Programs are meant to be read by humans and only incidentally for computers to execute.

— H. Abelson and G. Sussman (in "Structure and Interpretation of Computer Programs")

其中：

- H. Abelson and G. Sussman
是个牛人，维基百科中的解释是：
“哈尔·阿伯尔森（英语：Hal Abelson，1947年4月26日 - ）是美国麻省理工学院电子工程和计算机科学部门的一名著名计算机科学教授，电气电子工程师学会院士，创作共用，自由软件基金会，麻省理工网络开放课程，媒体实验室 Center for Mobile Learning，MIT App Inventor的创始领导人。”
- "Structure and Interpretation of Computer Programs"
是本书，“这本书被广泛地认为是计算机科学的经典教材，在计算机科学的教学领域产生了深刻的影响。”

总之：大牛们的观点都已经是：代码是写给别人看的，顺带只是用来被计算机执行的。

所以，你要是写的代码，让别人看不懂，那么就好好学习编程规范，写出让别人也易懂的代码吧。

第 2 章 通用的编程规范

此处介绍，不论是任何计算机语言，相对来说都，通用的，具有普适性的一些编程规范。

2.1. 编程规范之风格

2.1.1. 风格统一

整个项目遵守同一种风格。

包括但不限于：

- 命名方法
要么都是驼峰法命名，要么都是匈牙利命名法，不要两者混用。
- 括号的位置
花括号{}的位置，要是同一行，要是换行对齐，不要混用。
- 缩进
缩进是用空格还是TAB键
如果用空格缩进，空格是用2个还是4个
- 代码中合适的位置添加空格以便于增加代码可读性
 - 二元操作符的左右添加单个空格
比如，加减乘除等操作，大于号小于号等等
 -
 -
 -
- 代码中合适的地方要加上必要的括号
即使是，语法上，可以不用加括号，但是为了程序的可读性，以及减少笔误带来的出错机会，也要加对应的括号

2.2. 编程规范之注释

2.2.1. 添加注释的原则

不同语言都要加上必要的注释

必要的注释，指的是：

- 用注释说明代码背后的事情
如果代码本身不能说明全部问题，那么则一定要加上对应的注释，去说明代码背后的问题
- 不要多余的注释
在已经保证代码可读性足够好的情况下，无需加上多余的注释

2.2.2. 注释的语法

语言不同，注释的语法不同

2.3. 编程规范之命名

不论是用哪种语言去写代码，都会遇到，对于一个东西，给其起一个名字，即给东西命名

这个东西，包括但不限于：

- 变量variable
- 函数function
- 结构体structure

下面就来介绍一下，关于命名方面的规范。

2.3.1. 命名的原则

2.3.1.1. 见名知意

否则容易：让看代码的人就容易混淆

2.3.2. 命名方法

知道了命名时要遵循的原则后，就涉及到，如何给具体的变量、函数等内容去命名了。

命名这个事情，很早就有人专门总结了不同的，常见的方法：

2.3.2.1. 驼峰（式）命名法



帕斯卡 (pascal) 命名法

此处只是简要的提醒一下：

和驼峰式命名法类似的另外还有种方法，叫做：帕斯卡 (pascal) 命名法

2.3.2.2. 匈牙利命名法

2.3.3. 命名规范中的一些细节

知道了命名的方法和原则后，再来介绍一些，在命名的细节，即命名时的一些常见做法：

2.3.3.1. 常量大写

一般来说，常量，都是用大写。

2.3.3.2. 根据情况给名字加前缀

有些编程规范中，预定中，会针对于如下的一些情况，给变量加一些前缀：

- 模块内部变量加m前缀

m，表示module，模块，模块内部的变量

- 全局的外部的变量加g前缀

g，指的是global，全局的，外部的变量，相对于当前函数内部来说，是外部的全局的变量

- 常量加const前缀

这个做法，和之前介绍的，常量一般用大写，是互斥的，

两种做法，根据自己需要，采用其中一种即可。

2.4. 编程规范之设计原则

在你写代码时，往往也伴随着设计对应的函数，类，模块，甚至项目架构，而其中所涉及到的很多原则，也是属于编程规范的范畴。

下面就来介绍各种常见的编程规范的设计原则。

2.4.1. 高内聚，低耦合

不论是设计函数、类，模块，项目架构，等内容时，都要遵循的一个原则，那就是：高内聚，低耦合
简单解释就是：

- 高内聚：和当前功能上有逻辑关系的，放在一起。
- 低耦合：不要把和当前功能无关的东西牵扯进来

2.4.2. 使用频率高的代码，尽量做成函数和类库

将那些，使用频率高的，相对重合度很高的代码，整合到对应的，公共的函数中，甚至做成单独的类库

这样的好处是，随着你的编码时间和经验的提高，你的，或者你们公司的，项目的，公用的代码，函数，类库，会逐渐增加，为你以后的编程的质量和效率的提升，大有帮助。

比如，之前我在编程期间，就慢慢地积累了一些相对通用的函数，整理成自己的库函数了：

[crifanLib – Crifan' s Library \(C#/Python/PHP/C/...\) | 在路上](#)¹

有需要的可以去看看。

2.4.3. 嵌套层次不要太深

写代码涉及到嵌套时，比如for循环语句，不要嵌套太多层

一般不应该超过7层，否则就应该要回头审视你的设计的架构是否合理，或者是函数本身是否过于复杂以至于是否该把函数拆分了。

2.5. 不同语言自己的特定的编程规范

各种不同的计算机语言，基本上都有自己的编程规范

详见后面各个章节的介绍：

- [第 3 章 C语言编程规范](#)
- [第 4 章 C++语言编程规范](#)
- [第 5 章 Python语言编程规范](#)
- [第 6 章 GO语言编程规范](#)
- [第 7 章 Java语言编程规范](#)
- [第 8 章 C#语言编程规范](#)
- [第 9 章 PHP语言编程规范](#)
- [第 10 章 JavaScript语言编程规范](#)
- [第 11 章 其他语言的编程规范](#)

¹ http://www.crifan.com/crifan_released_all/crifanlib/

2.6. 不同公司或组织的编程规范

除了语言本身的编程规范之外，其他很多，相对比较正规以及重视技术积累的公司或组织，针对某些常用的语言，也制定了自己的一套编程规范，比如：

2.6.1. Google的编程规范

Google的R语言的编程规范：

[Google's R Style Guide](https://google-styleguide.googlecode.com/svn/trunk/Rguide.xml) ²

2.6.2. 华为的编程规范

华为的编程规范：

华为技术有限公司c语言编程规范_百度文库

[华为技术有限公司c语言编程规范_百度文库](http://wenku.baidu.com/view/40f60ec74028915f804dc2d7.html) ³

² <https://google-styleguide.googlecode.com/svn/trunk/Rguide.xml>

³ <http://wenku.baidu.com/view/40f60ec74028915f804dc2d7.html>

第 3 章 C语言编程规范

第 4 章 C++ 语言编程规范

第 5 章 Python语言编程规范

第 6 章 GO语言编程规范

网络上已有的一些整理好的GO语言的编程规范，可供参考：

- [The Go Programming Language Specification - The Go Programming Language](http://golang.org/ref/spec)¹

对应已有中文版：

[Go编程语言规范](http://zh-golang.appspot.com/ref/spec)²

- [Golang编码规范 - SegmentFault](http://segmentfault.com/blog/btc/1190000000464394)³

¹ <http://golang.org/ref/spec>

² <http://zh-golang.appspot.com/ref/spec>

³ <http://segmentfault.com/blog/btc/1190000000464394>

第 7 章 Java语言编程规范

7.1. Java语言编程规范

7.2. Android编程规范

Android，不算是一个普通意义上的编程规范。

Android首先是个平台，即大家常说的，移动领域内的操作系统。

而Android是Google，在基于嵌入式Linux基础上设计出的一套框架和平台，在此平台上可以开发Android的应用程序。

而针对于在Android平台上，用Java语言，开发Android的应用程序，也有自己特定的一些编程方面的规范，有些是和Java语言本身相关的，有些是和Android平台本身相关的。

第 8 章 C#语言编程规范

第 9 章 PHP语言编程规范

第 10 章 JavaScript语言编程规范

[Google JavaScript Style Guide](#) ¹

[JavaScript Style Guide | Contribute to jQuery](#) ²

[JavaScript编码规范 1 - SegmentFault](#) ³

[JavaScript编码规范 2 - SegmentFault](#) ⁴

¹ <https://google-styleguide.googlecode.com/svn/trunk/javascriptguide.xml>

² <http://contribute.jquery.org/style-guide/js/>

³ <http://segmentfault.com/blog/jslite/1190000002460897>

⁴ <http://segmentfault.com/blog/jslite/1190000002460944>

第 11 章 其他语言的编程规范

11.1. HTML编程规范

[编码规范 —— 编写灵活、稳定、高质量的 HTML 和 CSS 代码的规范 - SegmentFault](#)¹

11.2. CSS编程规范

[编码规范 —— 编写灵活、稳定、高质量的 HTML 和 CSS 代码的规范 - SegmentFault](#)²

¹ <http://segmentfault.com/blog/sunchengli/1190000000655954>

² <http://segmentfault.com/blog/sunchengli/1190000000655954>

参考书目

- [1] [非计算机专业C语言初学者编程规范\(学生用\)——命名规则_C语言中文网](#)¹
- [2] [Why Coding Style Matters - Smashing Magazine](#)²
- [3] [Coding conventions - Wikipedia, the free encyclopedia](#)³
- [4] [Programming style - Wikipedia, the free encyclopedia](#)⁴
- [5] [C Coding Standard](#)⁵
- [6] [C++ Coding Standard](#)⁶
- [7] [各种规范 - SegmentFault](#)⁷
- [8] [哈尔·阿伯尔森 - 维基百科，自由的百科全书](#)⁸

¹ <http://c.biancheng.net/cpp/html/752.html>

² <http://www.smashingmagazine.com/2012/10/25/why-coding-style-matters/>

³ http://en.wikipedia.org/wiki/Coding_conventions

⁴ http://en.wikipedia.org/wiki/Programming_style

⁵ <http://users.ece.cmu.edu/~eno/coding/CCodingStandard.html>

⁶ <http://users.ece.cmu.edu/~eno/coding/CppCodingStandard.html>

⁷ <http://segmentfault.com/bookmark/1230000002486653>

⁸ <http://zh.wikipedia.org/wiki/%E5%93%88%E5%B0%94%C2%B7%E9%98%BF%E4%BC%AF%E5%B0%94%E6%A3%AE>