### 附录 A 编码规范

本书提供的编码规范是以 Zend 官方资料为依据，根据编程原则融合并 提炼了多家软件公司长时间积累下来的成熟模式。目的是让软件项目遵守公 共一致的标准，帮助那些刚刚入行的 PHP 新手和正在使 PHP 开发的项目组 形成良好一致的编程风格，使参与者可以很快的适应环境，减少编码出错的 机会，在团队协作开发和项目后期维护中有更高的效率，以达到事半功倍的 效果，使项目长远健康的发展。防止部分参与者出于节省时间的需要，自创 一套风格并养成终生的习惯，导致其它人在阅读时浪费过多的时间和精力。

|  |  |  |
| --- | --- | --- |
| 文件状态: | 文件标识 | 编码规范 |
| [ ]草稿 | 当前版本: | 1.0 |
| [ √]正式发布 | 作者: | 兄弟连 |
| [ ]正在修改 | 完成日期: | 2018-4-3 |

**A1空行和空格**

**A1.1 空行的使用时机**

每段较大的程序体，上、下应当加入一个空白行，下列情况应该总是使用一个空行，禁 止使用多行：

* 两个函数声明之间。
* 函数内的局部变量和函数的第一条语句之间。
* 块注释或单行注释之前。
* 一个函数内的两个逻辑代码段之间，用以提高可读性。

**A1.2空格的使用时机**

空格的应用规则是可以通过代码的缩进提高可读性。

* 空格一般应用于关键字与左括号“(”之间，不过需要注意的是，函数名称与左括 号之间不应该用空格分开。右括号“)”除后面是“)”或者“.”以外，其他一律用 空格隔开它们；
* 一般在函数的参数列表中的逗号后面插入空格。
* 数学算式的操作数与运算符之间应该添加空格（特例是二进制运算与一元运算除 外，字符连接运算符号两边不加空格）。
* for 语句中的表达式应该用逗号分开，后面添加空格。
* 强制类型转换语句中的强制类型的右括号与表达式之间应该用逗号隔开，添加空 格。
* 除字符串中特意需要，一般情况下，在程序以及 HTML 中不出现两个连续的空格； 说明或显示部分中，内容如含有中文、数字、英文单词混杂，应当在数字或者英文 单词的前后加入空格。

**A2注释**

**A2.1 php开始和结束标记**

注释是对于那些容易忘记作用的代码以及关键性作用的代码添加简短的介绍性内容，html中使用<!-- 内容 --!>进行注释,php文件中使用//进行注释.

在程序开发中难免留下一些临时代码和调试代码，以免日后遗忘，此类代码必须添加注 释。所有临时性、调试性、试验性的代码，都可以添加统一的注释标记.例如:var\_dump();echo;print\_r;

**A3命名规范**

**A3.1 命名原则**

就一般约定而言，文件、目录、类、函数、变量和常量的名字，应该让代码阅读者能够 容易的知道这些代码的作用。命名的原则就是以最少的字母达到最容易理解的意义。命名是 程序规划的核心，只有了解系统的程序员才能为系统取出最合适的名字。如果所有的命名都 与其自然相适合，则关系清晰，含义就可以推导得出。形式越简单、越有规则，就越容易让 人感知和理解，应该避免使用不标准的命名。

**A4数据库设计规范**

**A4.1 数据表的设计**

**A4.1.1 数据库表名**

* 表名应具有描述性，杜绝一切拼音、或拼音英文混杂的命名方式；
* 表名允许使用字母，数字和下划线，不允许使用其他字符。表名使用单词开头，不允许 使用数字和下划线开头；
* 表名长度不能超过 64 个字符；
* 所有数据表名称，只要其名称是可数名词，则建议以复数方式命名，

**A4.1.2 数据库字段名**

* 字段名应具有描述性，杜绝一切拼音、或拼音英文混杂的命名方式
* 字段名允许使用字母，数字和下划线，不允许使用其他字符。字段名鼓励使用与所在表 的内容相关单词开头，允许但不鼓励使用数字和其他字符开头
* 字段名长度不能超过 64 个字符。
* 当几个表间的字段有关连时，要注意表与表之间关联字段命名的统一
* 每个表建议都要有一个代表 id 自增量的字段，可以使用全称的形式，也可只将其命名 为 id。

**A4.2 索引设计原则**

MySQL 索引，常用的有 PRIMARY KEY、INDEX、UNIQUE 几种。通常，在单表数据 值不重复的情况下，PRIMARY KEY 和 UNIQUE 索引比 INDEX 更快，要酌情使用。

索引能加快查询速度，而索引优化和查询优化是相辅相成的，既可以依据查询对索引进 行优化，也可以依据现有索引对查询进行优化，这取决于修改查询或索引，哪个对现有产品 架构和效率的影响最小。根据产品的实际运行和被访问情况，找出哪些 SQL 语句是最常被 执行的。最常被执行和最常出现在程序中是完全不同的概念。最常被执行的 SQL 语句，又 可被划分为对大表(数据条目多的)和对小表(数据条目少的)的操作。无论大表或小表，有可 分为读多、写多或读写都多的操作。

事实上，索引是将条件查询、排序的读操作资源消耗，分布到了写操作中，索引越多， 耗费磁盘空间越大，写操作越慢。因此，索引决不能盲目添加。对字段索引与否，最根本的 出发点，依次仍然是 SQL 语句执行的概率、表的大小和写操作的频繁程度。

**A5开发环境**

**A5.1 适用范围**

本文档提供的代码规则和文档指南不仅适用于《细说 PHP》中的每个实例，也适用于 所有 PHP 项目，意在帮助新手在编程风格上养成良好的习惯，也可以作为部分软件公司中 项目团队的参考文档，根据自己公司团队的特点进行部分修改。

**A5.2 目标**

能够遵守公共一致的编码标准对任何开发项目都很重要，特别是在多人的开发团队中， 编码标准能帮助确保代码的质量、减少 Bug 和容易维护。目标如下:

* 新人可以很快的适应环境，方便的融入到项目团队中
* 在一致的环境下，团队协作中有更高的效率，团队的成员可以减少犯错的机会
* 程序员可以方便的了解其他人的代码，弄清程序的状况，就和看自已的代码一样
* 防止接触 PHP 的新人自创一套风格并养成终生的习惯，一次次的犯同样的错误

**A5.3 开发工具**

PHP 的开发工具很多，常用的代码编辑工具有 Zend Studio、UltraEdit、EditPlus、PHPEdit、 Eclipse、 Dreamweaver 和 vim 等，每种开发工具都有其各自的优势。在编写程序时，一款 好的编辑工具会使程序员的编写过程更加轻松、有效和快捷，达到事半功倍的效果。对于一 款好的代码编辑工具，除了具备最基本的代码编辑功能外，一个必备的功能就是语法的高亮 显示、代码提示和代码补全。另外，一款好的代码编辑工具应具备格式排版功能，格式排版 功能可以使程序代码的组织结构清晰易懂，并且易于程序员进行程序调试，排除程序中的错 误异常。每程序员都可以根据自己的需求有选择性地使用开发工具，但开发工具种类之多， 也给程序员在选择上代来困惑。本规范对开发工具的使用有如下建议：

* 项目团队中的每个成员要在所使用的工具中设置统一的字符编码，例如 UTF-8，以 避免因为编码不统一，在项目整合时部分页面出现乱码
* 项目开发中常见的是使用缩进，缩进由制表符 tab 组成，目的是让代码组织结构和 层次清晰易懂。需每个参与项目的开发人员在编辑器中进行强制设定，每个缩进的 单位约定是一个 TAB(8 个空白字符宽度)， 以防在编写代码时遗忘而造成格式上 的不规范。本缩进规范适用于 PHP、JavaScript 中的函数、类、逻辑结构、循环等。
* 如果有必要每行代码的字符数也不宜过多，具体控制每行字符数量也需要在工具中 设定，80 字符以内是比较合适，但最多一行也不要超过 120 个字符
* 一个项目团队尽量要使用统一的开发工具，并且要统一版本