**Xenojoshua Coding Style**



**Version Control**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Change Description** |
| 1.0 | 01/24/2011 | Jonathan Dai | Document created |
| 1.1 | 01/25/2011 | Ila Wang | Add Javascript, HTML, CSS Style |
| 1.2 | 11/04/2011 | Jonathan Dai | Change the title |
| 1.3 | 01/01/2013 | Jonathan Dai | Enhance some parts |



1 PHP Part 2

1.1 basic基本点 2

1.2 class类 3

1.3 function函数 3

1.4 function comment函数注释 3

1.5 constant常量 3

1.6 binary operator二元运算符 3

1.7 ternary operator三元运算符 3

1.8 if判断语句 3

1.9 try catch语句 4

1.10 switch循环 4

1.11 for和foreach 4

1.12 string字符串处理 5

1.13 bo基本对象 5

2 MySql 5

2.1 database name库名 5

2.2 key word关键字 5

2.3 select查询 5

2.4 update更新 6

2.5 `的使用 6

3 JavaScript 6

3.1 用下划线\_，表示Javascript中的私有方法 6

4 HTML 6

4.1 页面使用<!doctype html>声明 6

4.2 完备的结构 6

4.3 Id, class, value等属性值均使用双引号包含 7

4.4 多个单词组成的id, class使用减号隔开，字母小写 7

4.5 大块的Div块的开始和结束加入注释 7

4.6 元素嵌套使用4个空格缩进 7

4.7 避免在html内嵌style属性 8

4.8 页面上的onclick事件绑定需要使用双引号 8

5 CSS 8

5.1 每项CSS均使用多行进行处理，属性名前使用4个空格缩进。 8

5.2 多个单词组成的id, class使用减号隔开，id和class名不允许有大写。 8

5.3 颜色代码使用十六进制，并使用大写 8

5.4 多种元素使用同一个CSS时，须归类并分行 8

5.5 禁止使用子集查询 9

# 

# PHP Part

## basic基本点

tab的宽度是4个空格

php标记只需要写在文件头，文件尾不需要该标记

<?php

## class类

以字母开头，首字母大写，词和词之间以大小写分隔

e.g.

class DBQueryBuilder {}

## function函数

以字母开头，首字母小写，词和词之间以大小写分隔，函数变量里的默认值等号左右都需要空格

e.g.

public function initFriends($uid, $range = NULL, $refresh = FALSE)

## function comment函数注释

必须包含三个部分，主要功能解释、参数、返回

e.g.

/\*\*

\* Convert join object | array to sql string

\* **@param** object | array $joins

\* **@return** string $query

\*/

解释部分首字母大写

参数部分以空格分隔首先需要写明白该参数的类型是什么，然后是参数名

返回值必须写，如果没有返回就写void

## constant常量

字母全部大写，词与词之间以下划线分隔，包括了static的常量和define定义出来的常量

e.g.

define('APPNAME', 'Pvzs');

public static $SECONDS\_IN\_ONE\_DAY = 86400;

## binary operator二元运算符

二元运算符左右都需要空格隔开，包含了[+ - \* / . == && || & | === > <]等一系列

此外，函数、变量里，一切用到逗号的地方，都需要在逗号后空格

e.g.

$user = $this->initUserData($uid);

$token -= $unlockRule->money;

e.g.

public function initFriends($uid, $range = NULL, $refresh = FALSE)

## ternary operator三元运算符

三元运算符的问号、冒号前后都需要空格

(is\_null($unlockNum)) ? $targetUnlockCount : $unlockNum;

## if判断语句

判断条件左右的括号都需要空格隔开，else与花括号不分行，如果有else if，分开写，不要写成elseif。此外，判断条件放在后面，这是为了防止双等号在某些时候会被误写成单等号赋值

e.g.

if (‘read’ == $mode) {

if (!$this->conRead->isConnected()) {

$this->conRead->connect();

}

} else if (‘write’ == $mode) {

if (!$this->conWrite->isConnected()) {

$this->conWrite->connect();

}

} else {

throw new DBException (1332, $mode);

}

## try catch语句

catch与之后的括号之间要空格

e.g.

try {

..

} catch (Exception $e) {

throw new DBException(1311, $query);

}

## switch循环

判断条件左右的括号都需要空格隔开，case以及下面的条件都需要隔开4个空格，

如果没有特殊的原因，每个case以及default都需要break

e.g.

switch ($action) {

case PvzsConst::$ACTION\_SET\_CONFIG:

$this->setConfigs();

break;

case PvzsConst::$ACTION\_IP\_LEVELS:

$processor = new PvzsExcelProcessor('IMPORTLEVELS');

$processor->setLevels();

break;

default:

throw new Exception('Invalid action provided!');

break;

}

## for和foreach

for遵循二元运算符的规则，foreach也是，判断条件左右的括号都需要空格隔开

e.g.

for ($i = 0; $i < $count; $i++) {

$query .= $columns[$i] . ' = ' . $values[$i];

if ($i < $count - 1) {

$query .= ', ';

}

}

e.g.

foreach ($table as $tableName => $tableAs) {

$query .= ‘ $tableName’ . $this->tableNum . ‘ $tableAs ‘;

}

## string字符串处理

能不用双引号的地方都不要用，尽量使用单引号

e.g.

$query .= 'GROUP BY ' . $groups . '';

e.g.

$query = "UPDATE `{$this->dbName}`.`$table{$this->tableNum}` SET ";

## bo基本对象

基本对象里的get和set方法，可能的话尽量写在一行里，并在花括号的内侧留下空格

e.g.

/\*\*

\* uid

\*/

public function getUid() { return $this->uid; }

public function setUid($uid) { $this->uid = $uid; }



# MySql

## database name库名

能添加库名的尽量添加，某些时候能减少非常莫名的错误

e.g.

UPDATE `pvzs`.`users001` SET `token` = 18494 WHERE `user\_id` = 245953145

## key word关键字

所有的关键字都需要大写

e.g.

UPDATE `pvzs`.`users001` SET `token` = 18494 WHERE `user\_id` = 245953145

## select查询

查询的时候不能使用\*，所有的字段都需要手动写出来，在以后添加字段之后可以在一定程度上减少错误出现

e.g.

SELECT `user\_id`, `level`, `experience`, `token`, `avatar`, `guide\_step` FROM `pvzs`.`users001` WHERE `user\_id` = 245953145

## update更新

update的时候尽量不要整行更新，只更新你需要更新的字段，减少写的压力

e.g.

UPDATE `pvzs`.`users001` SET `token` = 18494 WHERE `user\_id` = 245953145

## `的使用

库名、表名、列名，列名重命名AS後都需要用`包含

e.g.

UPDATE `pvzs`.`users001` SET `token` = 18494 WHERE `user\_id` = 245953145

SELECT `user\_id` AS `ID`, `level` AS `LV`, `experience`, `token`, `avatar`, `guide\_step` FROM `pvzs`.`users001` WHERE `user\_id` = 245953145



# JavaScript

基本同PHP的coding style

## 用下划线\_，表示Javascript中的私有方法

由于Javascript没有public private protected的概念，因此在function中使用this.\_name的方式标识私有方法和属性

e.g.

**var** Jonathan = **function**(){

**this**.\_age = 3;

**this**.\_getAge = **function**(){

**return** **this**.\_age;

}

}



# HTML

## 页面使用<!doctype html>声明

HTML5的标准，并且兼容市面上所有流行的浏览器。

## 完备的结构

页面须由以下顺序的嵌套格式进行构造

<!doctype html>

<html>

<head>

<meta http-equiv=*”Content-Type”* content=*”text/html;* charset=*utf-8”*>

<title>Title</title>

<style></style><!-- If Needed -->

<link rel=*”stylesheet”* type=*”text/css”* href=*”css/jquery-ui-1.8.6.custom.css”*>

…

<link rel=*”stylesheet”* type=*”text/css”* href=*”css/main.css”*>

<script type=*”text/javascript”* src=*”js/jquery-1.4.4.min.js”*></script>

…

<script type=*”text/javascript”* src=*”js/jquery-ui-1.8.6.custom.min.js”*></script>

</head>

<body>

<div>

…

</div>

</body>

</html>

## Id, class, value等属性值均使用双引号包含

e.g.

<div id=*”header”* class=*”blocks”*></div>

## 多个单词组成的id, class使用减号隔开，字母小写

使用减号而不是下划线是css通则

e.g.

<div id=*”this-id-is-very-long”*></div>

## 大块的Div块的开始和结束加入注释

Div嵌套过多会导致模板可读性变差，须加入注释缓解此问题。

e.g.

<!-- Body Begin -->

<div>

…

…

…

</div>

<!-- Body End -->

## 元素嵌套使用4个空格缩进

e.g.

<div>

<div></div>

<div>

<span></span>

</div>

<div></div>

</div>

## 避免在html内嵌style属性

style统一抽出放入CSS文件

## 页面上的onclick事件绑定需要使用双引号

e.g.

<div onclick=”popoutWindow();”>Popout</div>



# CSS

## 每项CSS均使用多行进行处理，属性名前使用4个空格缩进。

每种元素最后一行CSS需要加入分号结尾。

e.g.

#footer {

margin-left: *auto*;

margin-right: *auto*;

}

## 多个单词组成的id, class使用减号隔开，id和class名不允许有大写。

e.g.

#div-input-footer {

margin-left: *auto*;

margin-right: *auto*;

}

## 颜色代码使用十六进制，并使用大写

e.g.

#background {

background-color: *#CCCFFF*;

}

## 多种元素使用同一个CSS时，须归类并分行

e.g.

#head, #footer,

.title-first, .title-second, .title-third,

.list-first, .list-second, .list-third {

font-size: *150%*;

}

## 禁止使用子集查询

尽量不要使用全子集继承：

e.g.

#divA .list {

font-size: *150%*;

}

<div id=*”divA”*>

<div>

…

<div class=*”list”*></div>

…

</div>

</div>

子集继承会比较好，使用>将会只渲染当前选择器的子集：

e.g.

#divA > .list {

font-size: *150%*;

}