**PHP编码规范和实用技巧**

1. 文件编码必须utf8（没有bom），换行符必须使用Linux/Unix的 \n
2. 对于只含有 php 代码的文件，我们将在文件结尾处忽略掉 "?>" 。例如：  
   *<?php*  
   *$foo = 'foo';*
3. 缩进尽量使用四个空格， 禁止使用制表符TAB（phpstrom等IDE可以设置空格代替TAB）  
   例如：  
   if (1 == $x) {    $indented\_code = 1;    if (1 == $new\_line) {        $more\_indented\_code = 1;    }}
4. 每行代码长度应控制在80个字符以内，最长不超过120个字符。因为 linux 读入文件一般以80列为单位，就是说如果一行代码超过80个字符，那么系统将为此付出额外操作指令。这个虽然看起来是小问题，但是对于追求完美的程序员来说也是值得注意并遵守的规范。
5. 可以使用PhpStorm 快捷键 ctrl + alt + L 格式化代码
6. 每行结尾不允许有多余的空格，可以在phpstorm中设置。
7. 常量CAPS\_SNAKE\_CASE

例如：

const BAR\_BAZ = 0;

define('BAR\_BAZ', 0);

1. 类成员方法函数名单词首字母大写的驼峰语法。 全局函数名以小写字母下划线分割

例如：

function camelCase(){

// ...

}

function get\_file\_foo(){

// ...

}

1. 使用 $\_SERVER['REQUEST\_TIME'] 替代 time()
2. str\_replace、explode等对字符串操作的函数，先用 strops 判断是否存在字符。
3. 当替换前后的字符长度相同时，应使用strtr 取代 str\_replace
4. 使用数组值之前必须isset($arr[$key])判断是否存在。
5. 尽量使用 require/include 代替 require\_once /include\_once，并且使用绝对路径包含文件。
6. 有对应的常量的时候，不要使用函数。例：使用 \_\_DIR\_\_ 代替dirname()
7. 不要使用 @ 抑止错误警告
8. 代码不应该出现 E\_NOTICE警告，不应该通过php.ini屏蔽E\_NOTICE
9. 用单引号替代双引号引用字符串；
10. foreach效率更高，尽量用foreach代替while和for循环；

###### 常用时间范围取数据代码

1. 基本上每个项目都会用到开始时间和结束时间用来查找数据，下面有一个兼容性比较好的代码。

private $\_begindate = '';

private $\_enddate = '';

private $\_begindate\_int = 0; //查询条件：开始时间

private $\_enddate\_int = 0; //查询条件：结束时间

/\*\*

\* @return string

\*/

public function getBegindate()

{

return $this->\_begindate;

}

/\*\*

\* @param string $begindate

\*/

public function setBegindate($begindate)

{

$this->\_begindate = $begindate;

if ($this->\_begindate != '') {

$this->\_begindate\_int = strtotime($this->\_begindate);

if ($this->\_enddate\_int > 0 && $this->\_begindate\_int > $this->\_enddate\_int) {

$this->swapBeginAndEndTime();

}

}

return $this;

}

/\*\*

\* 交换开始和结束时间

\*/

private function swapBeginAndEndTime()

{

list($this->\_enddate, $this->\_begindate) = array($this->\_begindate,

$this->\_enddate);

list($this->\_enddate\_int, $this->\_begindate\_int) = array($this->\_begindate\_int,

$this->\_enddate\_int);

}

/\*\*

\* @param string $enddate

\*/

public function setEnddate($enddate)

{

$this->\_enddate = $enddate;

if ($this->\_enddate != '') {

$this->\_enddate\_int = strtotime($this->\_enddate);

$this->\_enddate\_int += 86399;

if ($this->\_begindate\_int > 0 && $this->\_begindate\_int > $this->\_enddate\_int) {

$this->swapBeginAndEndTime();

}

}

return $this;

}

/\*\*

\* @return string

\*/

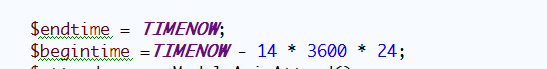
public function getEnddate()

{

return $this->\_enddate;

}

1. 取N天以内的数据



这种时间计算方式取14天以内的记录，只能取到14天前当前时间，而忽略了凌晨时间。

更合理的方法取14天前的凌晨时间：

$dt = new DateTime('00:00:00');

$begintime = $dt->sub(new DateInterval('P14D'))->getTimestamp();

**太多内容说不尽……**

**记住一个编程哲学思想：**

**DRY: Don't Repeat Yourself.（不要总是用相同代码解决相同问题）  
KISS: Keep It Simple, Stupid.（让代码简单直接）**