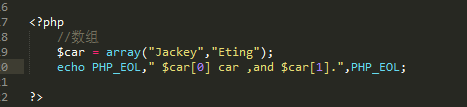
1. 在使用PHP的换行符(PHP\_EOL)时，注意要有逗号把其和要echo出的句子隔开，如：



1. 声明类的属性或方法为static时，可以不用实例化类而直接访问；

若属性为static声明，则不能通过实例化类访问，而方法为static声明时则可以；

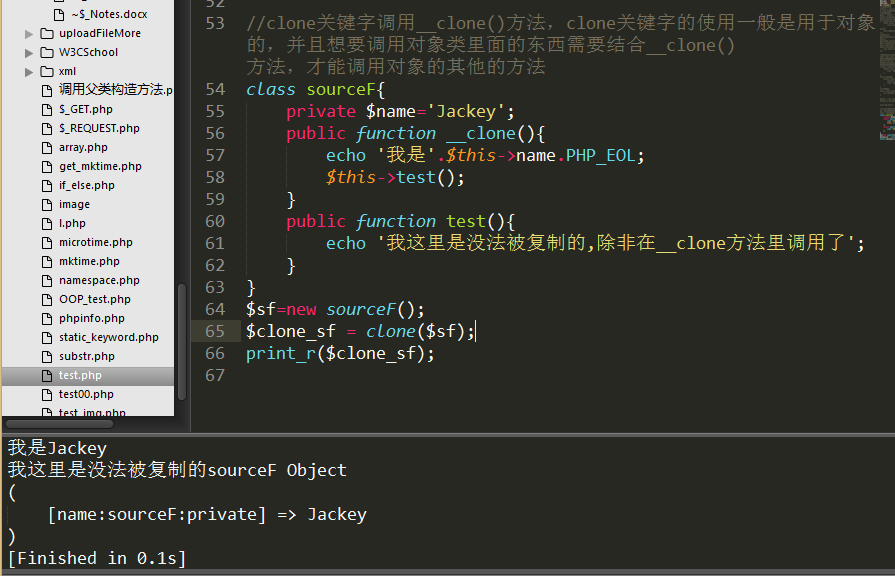
由于静态方法不需要通过对象即可调用，所以伪变量$this不能再static方法中使用；

Static属性不能用实例化 -> 来访问，要用 类名 :: static的属性 来访问，::（域运算符）

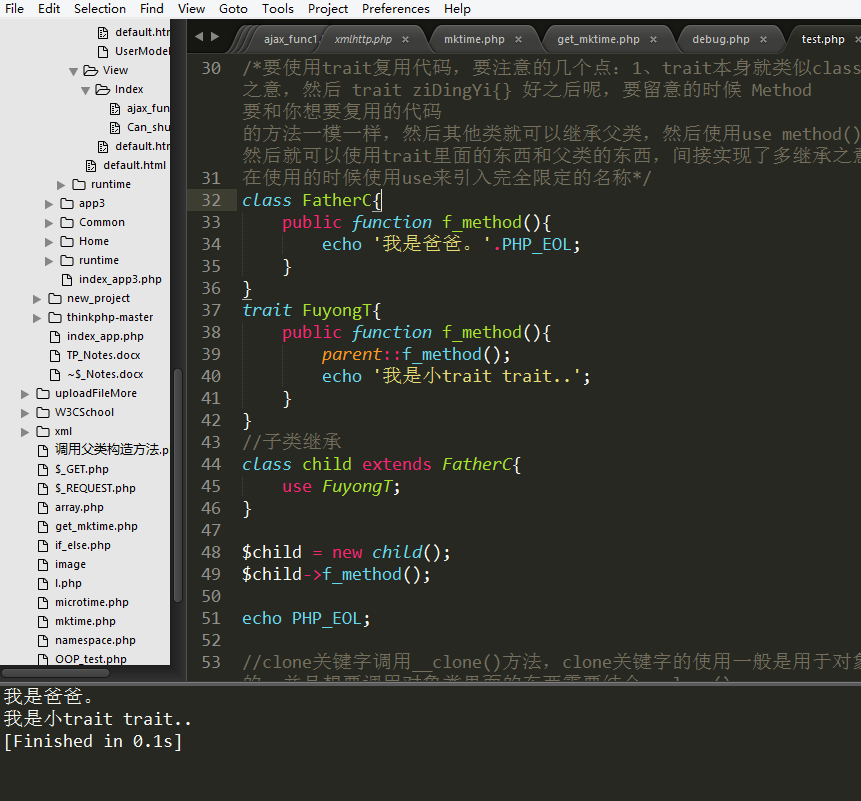
1. 数据库查询的一个雷坑。



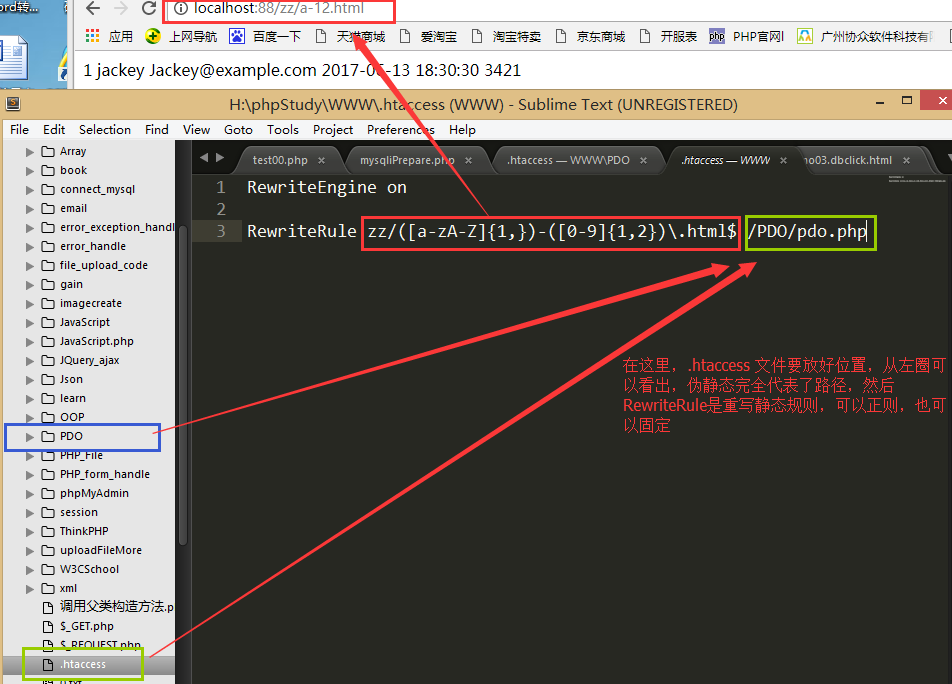
4、关键字clone() 来复制对象的变量和\_\_clone()方法里面的东西，不会去克隆其他意外的方法



5、trait是方法代码复用，trait 后面的名字随便定，然后{}花括号进去后，方法是要跟你要复用的类的方法要一模一样，然后如果一个子类想要使用trait的东西，那子类就得extends父类，然后再到子类里使用 use 来导入完全限定名称，调用trait了，下意识的使用了多个继承的样子，要注意的是:trait会覆盖了基类，子类会覆盖trait



6、apache 的伪静态

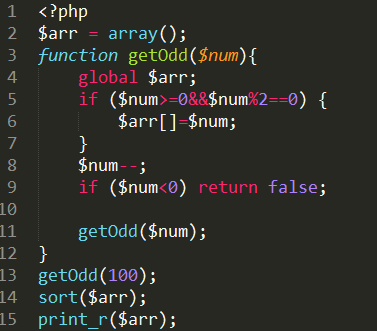


7、在使用 json\_decode 的时候要注意了，，如果你直接使用$\_POST,$\_REQUEST，$GET来直接获取一个前台传来的值的时候，不需要用json\_decode，因为你用了前面三个那样后，返回的是数组啦，兄弟。Json\_decode,是用于返回json格式的，如{name:”Jackey”}

8、递归函数心得：

1）如果你是用数字加到总和来递增的时候是可以一直加，然后最后返回一个数就行了；

2）如果是你递归到数组里面的，你可以采用全局数组命名，然后按顺序来改变参数值，如果值不符合则 return false 来停止自我的递归，如图：(主要抓住，数组不返回，用全局定义，然后用 return false 来结束递归)



9、一定要系统使用date\_default\_timezone\_set(‘PRC’)来设置后使用date()，在file\_put\_contents()前面的所有echo,print\_r等输出都会弄到ob 缓存里面，它之后才会在这页面显示(同时也是会在缓存页面出现。)



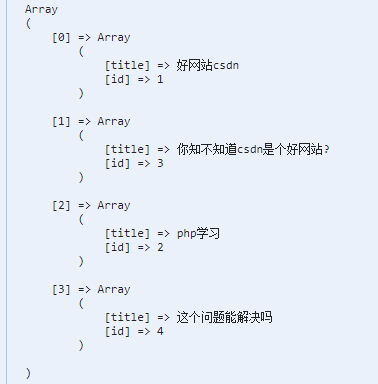
10、在做数组排序的时候，可以使用usort(array,method);用户自定义方法，失败则返回false,如： (usort主要是返回-1排前，返回1排后)

1、**$arr = array(2,3,1,10,5); usort($arr,” my\_sort”); function my\_sort($a,$b){ if($a==$b) return 0; return ($a<$b)? -1 : 1;}；**

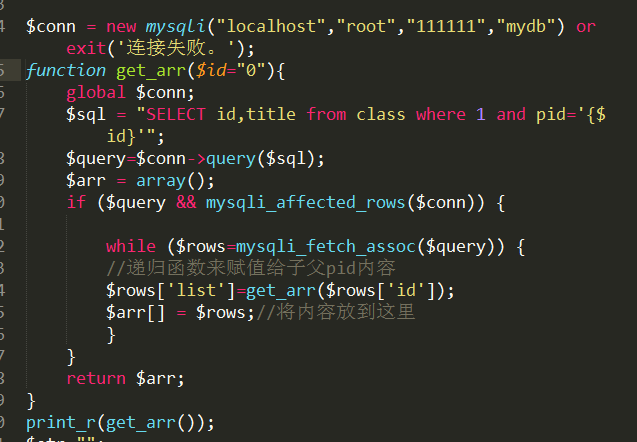
2、那假如我们需要对数组里面的内容相同的排在前面的做法：结合stripos(array,str);



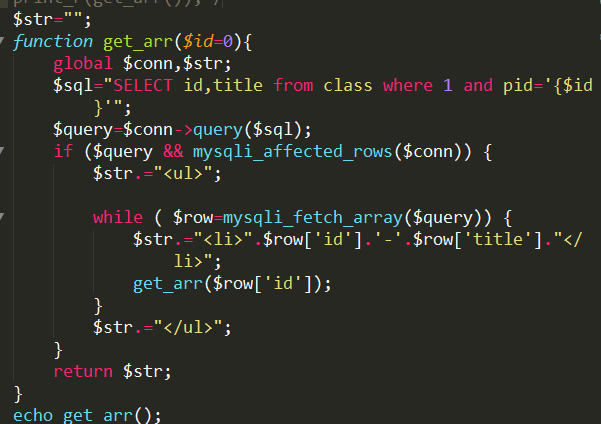
结果：

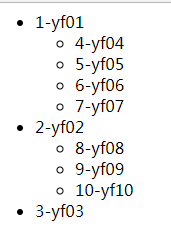


11、获取数据数组无限极递归：



获取数据字符串无限极递归：





12.1、http\_build\_query()是模仿http请求，经过了URL\_encode,然后生成一个字符串：





12.2、curl 的时候，遇到https 的请求，结果会提示你没有权利去请求，这个时候可以用：

curl\_setopt($ch, CURLOPT\_SSL\_VERIFYPEER, FALSE); curl\_setopt($ch, CURLOPT\_SSL\_VERIFYHOST, FALSE);,,然后设置了这个时候还会提示就要检查，**url 那里绝对不能有空格**，不然你会无厘头。

13、curl抓取页面的时候，如果遇到抓取的内容中，一些关键内容没有了，可能是对方做了referer认证，你可以使用 curl\_setopt($ch,CURLOPT\_REFERER,$url)

14、$this 代表本例，当前对象; self 是表示本类； 在有static 的时候，只能用self 去引用属性，而不能 $this-> 去引用当前类的属性。

15、php和mysql的事务处理回滚步骤：

1)关闭自动提交：mysqli, $mysqli->autocommit(false) , pdo , $pdh->setAttribute(ATTR\_AUTOCOMMIT,0);

2) 开始事务处理;(注意pdo 是要开始事务$pdh->beginTransaction(), mysqli则不用)

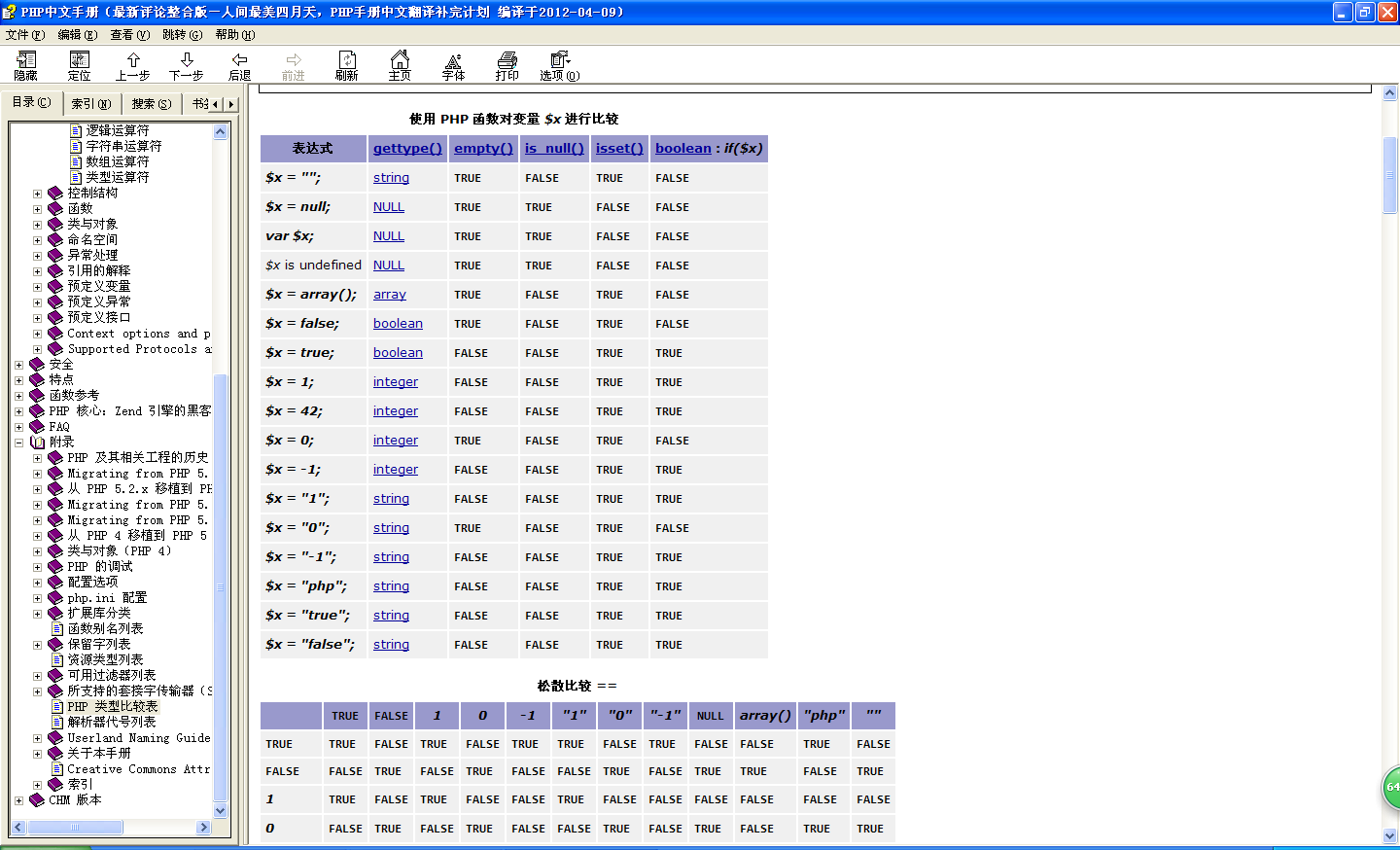
3) 抛出异常, 回滚(rollback() ,这里进行一些判断，错误则回滚);

4) 开启自动提交;

16、依赖注入(设计一个接口或类)是为了提高代码的利用率和方便维护。



17、



18、读取excel 然后再网页上显示，主要是用PHPExcel\_Writer\_HTML 这个来保存到.html,

//这里引入PHPExcel类

$filePath = 'Excel文件.xls';

$fileType = PHPExcel\_IOFactory::identify($filePath); //文件名自动判断文件类型

$objReader = PHPExcel\_IOFactory::createReader($fileType);

$objPHPExcel = $objReader->load($filePath);

$savePath = '文件保存路径/文件名.html'; //这里记得将文件名包含进去

$objWriter = new PHPExcel\_Writer\_HTML($objPHPExcel);

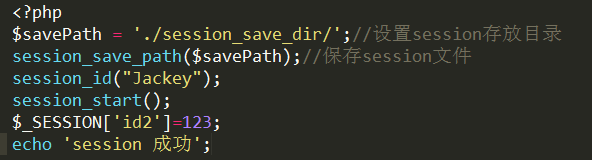
$objWriter->setSheetIndex(0); //可以将括号中的0换成需要操作的sheet索引

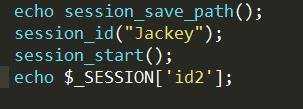
$objWriter->save($savePath); //保存为html文件

19、首先我们要规定一个父类或者接口，作为后续继承类的规范，然后工厂类主要是用来实现 实例化对象，这样易于维护和更改，但相对依赖注入缺少灵活性和拓展性，比如产品随时要更换短信发送验证，又突然改成邮箱验证，依赖注入会更灵活。



20、session是跟cookie没啥关系的，但session可以利用cookie来保存数据，比如设置session\_id($name) 这里来设置sessionID，然后可以在另外个页面读取这个session文件，但要留意你保存了路径的时候，同时要更改php.ini的session.save\_path 路径：



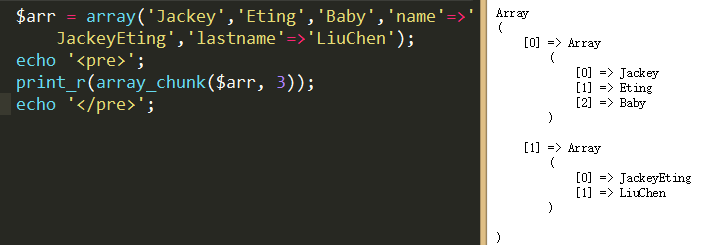


20.1 一开始浏览器默认是没有cookie的，但是一开始session\_start的时候，浏览器的cookie就会产生，就算你没setcookie()值，也会随机产生一个

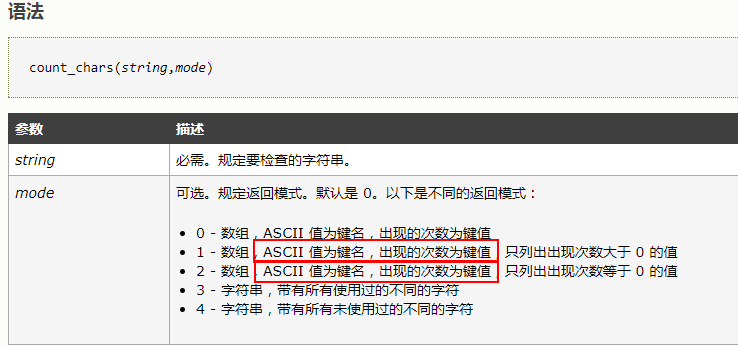
21、fileatime 来获取上次访问的时间，但前提是用打开了此文件去访问，比如fopen,file等打开文件指针，然后在window下有个小坑：就是怎么打开访问文件，fileatime的时间永远不变，那就要去更改下注册表的信息：

[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem]"NtfsDisableLastAccessUpdate"=dword:00000001改成[HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem]  
"NtfsDisableLastAccessUpdate"=dword:00000000。

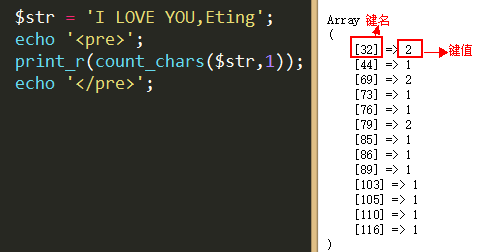
22、array\_chunk(array,size)的使用是将数组里的元素根据size划分，但是键名都被顺序化：



23、count\_chars(str,mode), mode 默认的是0 ,范围是0-4



可能不理解的应该是3,4, ：3：是返回这个字符串用过的所有字符；4：是返回所有没有用过的字符。



24、chr(ASCII码值)，值返回ASCII值后的字符。

25、一个有趣的执行外部东西的函数 ，shell\_exec(命令) 和 exec(文件名)(这里有个奇葩的就是当前目录下，读取文件，路径不能有’./’)； shell\_exec(命令)这里可以用windows下的cmd命令或者linux情况下的命令，比如 shell\_exec(“mkdir new\_file”);

26、empty(), isset(), isnull()的区别：

1) empty()::null ,undefined ,0 ,false ,空,都为true;

2)isset()::null,undefined,返回 true,,但是 空,0,false,返回false;

3)isnull()::就单纯的判断是否为null,是null就返回true;

27、今天下午看了下 urlencode，感觉有点懵逼，小试之后才觉得，咦，它主要是用于将 数字字母\_.-之外的转化成%号，但是空格的话是转为+号，它的编译是和application/x-form-urlencoded的编译模式是一样的，for instance:

$str='test-bla-bla-4>2-y-3<6';

$foo\_encoded=htmlspecialchar(urlencoded($str));

echo “<a href=’ mycgi? ’.$foo\_encoded>”;

**这么用也是为了在发送参数的时候可以不会因为其他的字符而影响到参数的完整性，在接收方可以用urldecoded来解析即可。**

28、在多并发的时候读写文件的时候，容易造成一些数据的丢失，这个时候就要考虑到 “锁”，有文件锁(flock($handle,$mode)); SQL悲观锁(for update); 也还有redis的锁，但redis本身就没有锁的机智; 还有队列::

1) flock($handle,LOCK\_EX|LOCK\_NB)是排他锁不阻塞，LOCK\_NB是非阻塞；

2)SQL的采用innodb的行锁来进行悲观锁(for update),记得要在begin里用，而且在where id=

3)redis本身就没有锁机制，只是利用了setnx来进行判断有责不操作，没有则操作，暂时不深入理解吧;

4)队列是可以慢慢的按顺序一排一排的进行；