1. 列举常见的http状态返回值及用途

200 OK  
指示客服端的请求已经成功收到，解析，接受。

202 Accepted  
请求已经被接受用来处理。但是处理并没有完成。请求可能或者根本没有遵照执行，因为处理实际执行过程中可能被拒绝。

400 Bad Request  
因为错误的[语法](http://wenwen.soso.com/z/Search.e?sp=S%E8%AF%AD%E6%B3%95&ch=w.search.yjjlink&cid=w.search.yjjlink" \t "http://bobo896.blog.163.com/blog/static/27315769201213115023150/_blank)导致服务器无法理解请求信息。

401 Unauthorized  
如果请求需要用户验证。回送应该包含一个WWW-Authenticate头字段用来指明请求资源的权限。

402 Payment Required  
保留状态码

403 Forbidden  
服务器接受请求，但是被拒绝处理。

404 Not Found  
服务器已经找到任何匹配Request-URI的资源。

500 Internal Server Error  
服务器遭遇异常阻止了当前请求的执行

501 Not Implemented  
服务器没有相应的执行动作来完成当前请求。

502 Bad Gateway

503 Service Unavailable  
因为临时文件超载导致服务器不能处理当前请求。

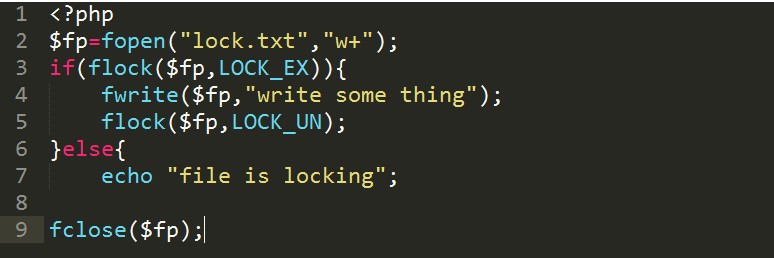
504 Gateway Timeout

505 Http Version Not Supporte

1. **如何在命令行下运行php脚本同时向php脚本传递参数**

参数直接跟到，执行php test.php 参数1,参数2,.脚本中这样接收 echo "接收到{$argc}个参数";  
print\_r($argv);

1. **请写一段php代码，确保多个进程同时写入同一个文件成功**

****

—fopen 打开文件或者 URL

--flock轻便的咨询文件锁定 --fwrite写入文件

# 5.[你用什么方法检查 PHP 脚本的执行效率（通常是脚本执行时间）和数据库 SQL 的效率（通常是数据库 Query 时间）， 并定位和分析脚本执行和数据库查询的瓶颈所在？](http://www.cnblogs.com/love1226/p/4646325.html)

 php：

一般是在你要检查的代码开头记录一个时间，结尾记录一个时间。取差值，

数据库SQL的效率  
    sql的explain(mysql),启用slow query log记录慢查询。  
   通常还要看数据库设计是否合理，需求是否合理等。

1. 一般都是不可控性。sql再优化也并不会提升多少速度（10%已经很多了）。除非出现慢查询，这个可以根据mysql日志分析。  
   2.很多时候程序和数据库是异地的，执行效率很难排查。  
   3.sql执行效率可以把方法拆分了。检查是否有循环请求sql操作。  
   4.增加缓存机制
2. 关于限制用户投票

## 在“vote”数据库中创建一个名为“user”的数据表用来保存用户进行投票的IP地址和所投票的信息。3.4、限制投票次数的功能实现

### 实现通过客户端IP限制投票的次数功能

1. **关于meta标签**

$html="<meta http-equiv='Content-Type' content='text/html;charset=gbk'>";

//匹配标准的meta标签

$pattern = "/<meta\s+http-equiv=(\'|\")?Content-Type(\'|\")?\s+content=(\'|\")text\/html;

\s+charset=(.\*)(\'|\")>/i";

$replacement="<meta http-equiv='Content-Type' content='text/html; charset=big5'>";

//执行一个正则表达式的搜索和替换

$result=preg\_replace($pattern, $replacement, $html);

**echo** htmlspecialchars($result);

1. 计算两个文件的相对路径

//写一个函数，计算两个文件的相对路径

$a='/a/b/c/d/e.php';

$b='/a/b/12/34/c.php';

**function** getRelativePath($a,$b){

//给出一个包含有指向一个文件的全路径的字符串，本函数返回去掉文件名后的目录名。

$returnPath=**array**(dirname($b));

//Array ( [0] => /a/b/12/34 )

// print\_r($returnPath);

$arrA=explode('/', $a);

// print\_r($arrA);Array ( [0] => [1] => a [2] => b [3] => c [4] => d [5] => e.php )

$arrB=explode('/', $returnPath[0]);

// print\_r($arrB);Array ( [0] => [1] => a [2] => b [3] => 12 [4] => 34 )

**for**($i=1,$len=count($arrB);$i<$len;$i++){

**if**($arrA[$i]!=$arrB[$i]){

**break**;

}

}

**echo** $i;

**if**($len-$i>0){

//array\_merge 合并一个或多个数组

//用 value参数的值将一个数组填充 num 个条目，键名由 start\_index 参数指定的开始。

$returnPath=array\_merge($returnPath,array\_fill(1, $len-$i, '..'));

print\_r($returnPath);

// Array ( [0] => /a/b/12/34 [1] => .. [2] => .. )

}

//— array\_slice从数组中取出一段

$returnPath=array\_merge($returnPath,array\_slice($arrA, $i));

// print\_r($returnPath);

// Array ( [0] => /a/b/12/34 [1] => .. [2] => .. [3] => c [4] => d [5] => e.php )

**return** implode('/', $returnPath);

}

**echo** getRelativePath($a, $b);

// /a/b/12/34/../../c/d/e.php