

## PHP 核心技术

### 1. 写出一个能创建多级目录的 PHP 函数（新浪网技术部）

md

```
1 <?php
2 /**
3  * 创建多级目录
4  * @param $path string 要创建的目录
5  * @param $mode int 创建目录的模式，在windows下忽略
6  */
7 function create_dir($path,$mode = 0777){
8     if (is_dir($path)) {
9         # 如果目录已经存在，则不创建
10        echo '该目录已经存在';
11    } else {
12        # 不存在，创建
13        if (mkdir($path,$mode,true)) {
14            echo '创建目录成功';
15        } else {
16            echo '创建目录失败';
17        }
18    }
19 }
```

### 2. 写出 smarty 模板的特点（新浪网技术部）

速度快，编译型，缓存技术，插件机制，强大的表现逻辑，但 smarty3.0 速度较慢，但是从总体上面来说现在用的企业越来越少了，追求高效的企业，建议使用<?= 原生 PHP 的模版替换机制。

### 3. 打开 php.ini 中的 safe\_mode，会影响哪些函数？至少说出 6 个。（新浪）

safe\_mode, php 安全模式，它提供一个基本安全的共享环境，在一个有多个用户账户存在的 php 开发的 web 服务器上。当安全模式打开的时候，一些函数将被完全的禁止，而另一些函数的功能将会受到限制，如：chdir, move\_uploaded\_file, chgrp, parse\_ini\_file, chown, rmdir, copy, rename, fopen, require, mkdir, unlink 等。

注意，在 php5.3 以上版本，safe\_mode 被弃用，在 php5.4 以上版本，则将此特性完全去除了。

### 4. 抓取远程图片到本地,你会用什么函数? (51.com 笔试题)

file\_get\_contents 或者 curl

## 5. PHP 的垃圾收集机制是怎样的（腾讯）

PHP 可以自动进行内存管理，清除不再需要的对象。PHP 使用了引用计数(reference counting)这种单纯的垃圾回收(garbage collection)机制。每个对象都内含一个引用计数器，每个 reference 连接到对象，计数器加 1。当 reference 离开生存空间或被设为 NULL，计数器减 1。当某个对象的引用计数器为零时，PHP 知道你不再需要使用这个对象，释放其所占的内存空间。

## 6. 请写一段 PHP 代码，确保多个进程同时写入同一个文件成功（腾讯）核心思路：加锁

```
1 <?php
2 $fp = fopen("lock.txt", "w+");
3 if (flock($fp, LOCK_EX)) {
4     //获得写锁，写数据
5     fwrite($fp, "write something");
6     //解除锁定
7     flock($fp, LOCK_UN);
8 } else {
9     echo "file is locking...";
10 }
11 fclose($fp);
```

## 7. 写一个函数，尽可能高效的，从一个标准 url 里取出文件的扩展名，例如：

<http://www.sina.com.cn/abc/de/fg.php?id=1> 需要取出 php 或 .php（新浪）参考方案 1:

```
1 <?php
2 function getExt($url){
3     $arr = parse_url($url);
4     $file = basename($arr['path']);
5     $ext = explode('.', $file);
6     return $ext[count($ext)-1];
7 }
8 $path = 'http://www.sina.com.cn/abc/de/fg.php?id=1';
9 echo getExt($path);
```

参考方案 2:

```

1  <?php
2  function getExt($url){
3      $url = basename($url);
4      $pos1=strpos($url,'. ');
5      $pos2 = strpos($url,'? ');
6      if(strpos($url,'? ')){
7          return substr($url,$pos1+1,$pos2-$pos1-1);
8      }else{
9          return substr($url,$pos1);
10     }

```

相关题目：使用五种以上方式获取一个文件的扩展名，要求：**dir/upload.image.jpg**，找出 **.jpg** 或者 **jpg**，必须使用 **PHP** 自带的处理函数进行处理，方法不能明显重复，可以封装成函数，比如 `get_ext1($file_name)`, `get_ext2($file_name)`

## 8. 写一个函数，能够遍历一个文件夹下的所有文件和子文件夹。 (新浪)

```

1  <?php
2  function my_scandir($dir){
3      $files = array();
4      if (is_dir($dir)){
5          if ($handle = opendir($dir)){
6              while (($file = readdir($handle)) !== false){
7                  if ($file!="." && $file!=".."){
8                      if (is_dir($dir."/".$file)){
9                          $files[$file] = my_scandir($dir."/".$file);
10                     } else{
11                         $files[]=$dir."/".$file;
12                     }
13                 }
14             }
15             closedir($handle);
16             return $files;
17         }

```

## 9. 简述论坛中无限分类的实现原理。(新浪) 创建类别表如下:

```

CREATE TABLE category( cat_id smallint unsigned not null auto_increment primary key
    comment '类别 ID', cat_name VARCHAR(30) NOT NULL DEFAULT " COMMENT '
    类别名称', parent_id SMALLINT UNSIGNED NOT NULL DEFAULT 0 COMMENT
    '类别父 ID'

```

```
)engine=MyISAM charset=utf8;
```

编写一个函数，递归遍历，实现无限分类

```

1  <?php
2  function tree($arr,$pid=0,$level=0){
3      static $list = array();
4      foreach ($arr as $v) {
5          //如果是顶级分类, 则将其存到$list中,
6          //并以此节点作为根节点, 遍历找其子节点
7          if ($v['parent_id'] == $pid) {
8              $v['level'] = $level;
9              $list[] = $v;
10             tree($arr,$v['cat_id'],$level+1);
11         }
12     }
13     return $list;
14 }

```

10. 写一个函数, 算出两个文件的相对路径, 如 `$a = '/a/b/c/d/e.php'`;  
`$b = '/a/b/12/34/c.php'`;  
 计算出 `$b` 相对于 `$a` 的相对路径应该是 `../../c/d` (新浪)

```

1  <?php
2  function relative_path($path1,$path2){
3      $arr1 = explode('/', dirname($path1));
4      $arr2 = explode('/', dirname($path2));
5      for ($i = 0, $len = count($arr2); $i < $len; $i++) {
6          if ($arr1[$i] != $arr2[$i]) {
7              break;
8          }
9      }
10     #不在同一个根目录下
11     if ($i == 1){
12         $return_path = array();
13     }
14     #在同一个根目录下
15     if ($i != 1 && $i < $len) {
16         $return_path = array_fill(0, $len - $i, "..");
17     }
18     #在同一个目录下
19     if ($i == $len){
20         $return_path = array('./');
21     }
22     $return_path = array_merge($return_path, array_slice($arr1, $i));
23     return implode('/', $return_path);
24 }
25 $a = '/a/b/c/d/e.php';
26 $b = '/a/b/12/34/c.php';
27 $c = '/e/b/c/d/f.php';
28 $d = '/a/b/c/d/g.php';
29 echo relative_path($a,$b); // 结果 ../../c/d
30 echo relative_path($a,$c); // 结果 a/b/c/d
31 echo relative_path($a,$d); // 结果 ./

```

## 11. 有一个网页地址，比如 PHP 开发资源网主页： <http://www.phpres.com/index.html>，如何得到它的内容？

方法 1(对于 PHP5 及更高版本):

```
$readcontents = fopen("http://www.phpres.com/index.html", "rb");  
$contents = stream_get_contents($readcontents);  
fclose($readcontents); echo $contents;
```

方法 2:

```
echo file_get_contents("http://www.phpres.com/index.html");
```

## 12. 谈谈对 mvc 的认识

由模型(model),视图(view),控制器(controller)完成的应用程序由模型发出要实现的功能到控制器,控制器接收组织功能传递给视图;

MVC 是一个设计模式，它强制性的使应用程序的输入、处理和输出分开。使用 MVC 应用程序被分成三个核心部件：模型、视图、控制器。它们各自处理自己的任务。视图是用户看到并与之交互的界面。模型表示企业数据和业务规则。控制器接受用户的输入并调用模型和视图去完成用户的需求。

MVC 的优点：低耦合性、高重用性和可适用性、较低的生命周期成本、快速的部署、可维护性、可扩展性，有利于软件工程化管理

MVC 的缺点：没有明确的定义，完全理解 MVC 并不容易。不适合小型规模的应用程序。

## 13. What does the GD library do? (Yahoo)

GD 库提供了一系列用来处理图片的 API，使用 GD 库可以处理图片，或者生成图片。在网站上 GD 库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表。自从 PHP 4.3.0 版本后 GD 便内建在 PHP 系统中。

## 14. What function can you use to open a file for reading and writing? (Yahoo)

A. fget();

B. file\_open();

C. fopen();

D. open\_file(); 答案：C fget() 这不是一个 PHP 函数，会引致执行错误。

file\_open() 这不是一个 PHP 函数，会引致执行错误。

fopen()这是正确答案，fopen() 可以用来开启档案以便读写。

open\_file() 这不是一个 PHP 函数，会引致执行错误。

## 16. Smarty 的原理

smarty 是一个模板引擎，使用 smarty 主要是为了实现逻辑和外内容的分离，如果不使用模板的话，通常的做法就是 php 代码和 html 代码混编。使用了模板之后，则可以将业务逻辑都放到 php 文件中，而负责显示内容的模板则放到 html 文件中。Smarty 在执行 display 方法的时候，读取模板文件，并进行数据替换，生成编译文件，之后每次访问都会直接访问编译文件，读取编译文件省去了读取模板文件，和字符串替换的时间，所以可以更快，编译文件里时间戳记录模板文件修改时间，如果模板被修改过就可以检测到，然后重新编译（编译是把静态内容保存起来，动态内容根据传入的参数不同而不同）。如果启用了缓存，则会根据编译文件生成缓存文件，在访问的时候如果有缓存文件并且缓存文件没有过期，则直接访问缓存文件。

相关题目 1：能够使 HTML 和 PHP 分离使用的模板 smarty  
phplib 等

相关题目 2：您是否用过模板引擎？如果有您用的模板引擎的名字是？  
Smarty

## 17. PHP 如何实现页面跳转

方法一：php 函数跳转，缺点，header 头之前不能有输出，跳转后的程序继续执行，可用 exit 中断执行后面的程序。

```
header("Location: 网址"); //直接跳转
```

```
header("refresh:3,url=http://axgle.za.net"); //三秒后跳转
```

方法二：利用 meta

```
echo "<meta http-equiv=refresh content='0; url=网址'>";
```

## 18. PHP 可以和 sql server/oracle 等数据库连接吗？可以

## 19. 使用哪些工具进行版本控制？ SVN 或者 CVS 或者 git

相关题目：您是否用过版本控制软件？如果有您用的版本控制软件的名字是？  
TortoiseSVN-1.2.6

## 20. 写出一个正则表达式，过滤网页上的所有 JS/VBS 脚本（即把 script 标记及其内容都去掉）：

过滤 javascript 脚本参考：

```
1 <?php
2 $script="以下内容不显示: <script type='text/javascript'>alert('cc');</script>";
3 $pattern = '/<script[^\>]*?>. *?<\</script>/si';
4 echo preg_replace($pattern, "替换内容", $script);
5 //输出为 以下内容不显示: 替换内容
```

## 21. Given a line of text \$string, how would you write a regular expression to strip all the HTML tags from it? (Yahoo)

方案一，使用 PHP 内建函数 strip\_tags() 除去 HTML 标签方案二，自定义函数，如下：

```

1  <?php
2  header('Content-Type:text/html;charset=utf8');
3  function strip_html_tags($str){
4      $pattern = '/<("[^"]*"|'\'[^\']*\'|<[^>"]*)>/' ;
5      return preg_replace($pattern, '', $str);
6  }
7  //实例
8  $html = '<p id="">dddd<br />dd<br></p>';
9  echo strip_html_tags($html);
10 $html = '<p id="">aaaa<br />aa<br></p>';
11 echo strip_html_tags($html);
  
```

## 22. 请写一个函数验证电子邮件的格式是否正确（要求使用正则） （新浪） preg\_match('/^[w\-\.]++@[w\-\.]++(\.\w++)+\$/', \$email);

相关题目：请用正则表达式写一个函数，验证电子邮件的格式是否正确。（鑫众人云）

## 23. 请对 POSIX 风格和兼容 Perl 风格两种正则表达式的主要函数进行类比说明（腾讯）主要区别有以下三种：

- 1) preg\_replace()里面的正则可以写成型如："/.xxx/"而 ereg\_replace()里面的正则需写成型如  
".xxx"
- 2) preg\_replace()能操作数组，而 ereg\_replace()不可以
- 3) 在逆向引用用 preg\_replace()可使用 0-99 个，而 ereg\_replace()最多为 9 个使用 Perl 兼容正则表达式语法的 preg\_match() 函数通常是比 ereg() 更快的替代方案。

## 24. 请写出并说明如何在命令行下运行 PHP 脚本（写出两种方式） 同时向 PHP 脚本传递参

数？（腾讯）首先进入

php 安装目录

php -f d:/wamp/www/1.php

php -r phpinfo();

其中-f 参数指定要执行的 php 文件

其中-r 表示直接执行 php 代码，无需写开始结束标记

25. 使用正则表达式提取一段标识语言（html 或 xml）代码段中指定标签的指定属性值（需考虑属性值对不规则的情况，如大小写不敏感，属性名值与等号间有空格等）。此处假设需提取 test 标签的 attr 属性值，请自行构建包含该标签的串（腾讯）

编写如下函数：

```
function getAttrValue($str, $tagName, $attrName){
    $pattern1 = "</> . $tagName . "(\\s+\\w+\\s*=\\s*(\\\"|'\"")?([^\\"|'\""]*)
        (\\2))*\\s+" . $attrName . "\\s*=\\s*(\\\"|'\"")?([^\\"|'\""]*)
        (\\s+\\w+\\s*=\\s*(\\\"|'\"")?([^\\"|'\""]*)\\s*>/i";
    $arr = array();
    $re = preg_match($pattern1, $str,
    $arr); if($re){ echo "<br
    />\\$arr[6]={ $arr[6]}";
    }
    else{
        echo "<br />没找到。";
    }
}
```

测试：

```
$str1 = "<test attr='ddd'>";
getAttrValue($str1,"test","attr"); //找 test 标签中 attr 属性的值,结果为 ddd
$str2 = "<test2 attr='ddd' attr2='ddd2' t1='t1 value' t2 = 't2 value'>";
getAttrValue($str2,"test2","t1"); //找 test2 标签中 t1 属性的值, 结果为 t1 value
```

26. What does the following code do? Explain what's going on there.  
`$date='08/26/2003';`

```
print_ereg_replace("(\\d{2}/\\d{2}/\\d{4})", "\\d{2}/\\d{2}/\\d{4},$date); (Yahoo)
```

这是把一个日期从 MM/DD/YYYY 的格式转为 DD/MM/YYYY 格式。

输出 26/08/2003

27. What function would you use to redirect the browser to a new page?  
 (Yahoo)

- A. `redir()`
- B. `header()`
- C. `location()`    D. `redirect()`

答案：B `redir()` 这不是一个 PHP 函数，会引致执行错误。  
`header()` 这个是正确答案，`header()` 函数发送头信息，可以用来使浏览器转向到另一个页面，



例如: header("Location: <http://www.search-this.com/>"). location() 这不是一个 PHP 函数, 会引致执行错误。 redirect() 这不是一个 PHP 函数, 会引致执行错误。

28. when turned on \_\_\_\_\_ will \_\_\_\_\_ your script with different variables from HTML forms and cookies. (腾讯)

- A. show\_errors, enable
- B. show\_errors, show
- C. register\_globals, enhance
- D. register\_globals, inject

答案: C

29. 一个函数的参数不能是对变量的引用, 除非在 php.ini 中把\_\_\_\_\_ 设为 on。 allow\_call\_time\_pass\_reference

是否启用在函数调用时强制参数被按照引用传递

30. 在 HTML 语言中, 页面头部的 meta 标记可以用来输出文件的编码格式, 以下是一个标准的 meta 语句 <META http-equiv='Content-Type' content='text/html; charset=gbk'>, 请使用 PHP 语言写一个函数, 把一个标准 HTML 页面中的类似 meta 标记中的 charset 部分值改为 big5。(新浪) 请注意:

- (1) 需要处理完整的 html 页面, 即不光此 meta 语句
- (2) 忽略大小写
- (3) ' 和 " 在此处是可以互换的
- (4) 'Content-Type' 两侧的引号是可以忽略的, 但 'text/html; charset=gbk' 两侧的不行
- (5) 注意处理多余空格编写正则表达式如下:

```
$reg1 = "/(<meta\\s*http-equiv\\s*=\\s*([\"']?)Content-Type(\\2)\\s*content\\s*=\\s*([\"'])text/html; charset=(UTF-8)(\\4)(\\s*\\/?)>)/i";
```

31. PHP 中如何判断一个字符串是否是合法的日期模式: 2007-03-13 13:13:13。要求代码不超过 5 行。(酷讯)

```
1  <?php
2  function checkDateTime($data) {
3      if (date('Y-m-d H:i:s', strtotime($data)) == $data){
4          return true;
5      } else {
6          return false;
7      }
8  }
9  //实例
10 $date = '2013-06-01 20:12:56';
11 var_dump(checkDateTime($date)); //true
12 $date = '2013-03-32 20:16:10';
13 var_dump(checkDateTime($date)); //false
```

### 32. PHP 中，如何获得一个数组的键值？（酷讯）

使用 `key()` 可以获得数组中当前元素的键名，使用 `current()` 则可以返回当前元素的值。  
使用 `array_keys()` 则可以得到数组中所有的键名。使用 `foreach` 结构 `foreach($arr as $key=>$value)` 可以通过 `$key` 和 `$value` 分别获取键名和值。

### 34. 哪个选项会匹配下边的这个正则表达式? (/\*xyz\d/)

- A. \*\*\*\*\*xyz
  - B. \*\*\*\*\*\_xyz1
  - C. \*\*\*\*\*xyz2
  - D. \*\_xyz
- 答案: C

### 35. 以下哪个错误无法被标准的错误控制器获取?

- A. E\_WARNING
- B. E\_USER\_ERROR
- C. E\_PARSE
- D. E\_NOTICE

答案: B

### 36. 以下哪种错误类型无法被自定义的错误处理器捕捉到? (奇矩互动)

- A. E\_WARNING
- B. E\_USER\_ERROR
- C. E\_PARSE
- D. E\_NOTICE

答案: C

### 37. (^s\*)|(s\*\$)这个正则表达式作用是: \_\_\_\_\_;

匹配以 0 个或多个空白符开头或者 0 个或多个空白符结尾的字符串

### 38. 编写函数取得上一月的最后一天

```
1 <?php
2 date_default_timezone_set('PRC');
3 /**
4  * 获取给定月份的上一月最后一天
5  * @param $date string 给定日期
6  * @return string 上一月最后一天
7  */
8 function get_last_month_last_day($date = ''){
9     if ($date != '') {
10         $time = strtotime($date);
11     } else {
12         $time = time();
13     }
14     $day = date('j',$time);//获取该日期是当前月的第几天
15     return date('Y-m-d',strtotime("-{$day} days",$time));
16 }
17 //测试
18 echo get_last_month_last_day();
19 echo get_last_month_last_day('2013-3-21');
```

39. 在很多时候，我们可以通过 **apache** 的主配置文件来设置对 **test** 目录的访问权限控制，如 **http://IP/test** 请问如果需设置 **test** 下的一个子目录的访问控制权限，是否可以在主配置文件中修改，如果不可以应如何解决。可以，还可以在需要控制的子目录下创建 **.htaccess** 文件，写入访问控制。

40. 如果我的网站用的 **utf-8** 编码，为防止乱码出现，都需要注意哪些地方？从以下几个方面考虑：

- 1) 数据库中库和表都用 **utf8** 编码
- 2) **php** 连接 **mysql**，指定数据库编码为 **utf8** ；
- 3) **php** 文件指定头部编码为 **utf-8** header("content-type:text/html;charset=utf-8");
- 4) 网站下所有文件的编码为 **utf8**
- 5) **html** 文件指定编码为 **utf-8** <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

41. 在 **url** 中用 **get** 传值的时候，若中文出现乱码，应该用哪个函数对中文进行编码？ `urlencode()`

42. 写出两种对变量加密的函数? md5(\$str); sha1(\$str);

43. 如何把 2009-9-2 10:30:25 变成 unix 时间戳?

```
1 <?php
2 date_default_timezone_set("PRC");
3 //变成unix时间戳
4 $unix_time = strtotime("2009-9-2 10:30:25");
5 echo $unix_time;
6 //格式化为正常时间格式
7 echo date("Y-m-d H:i:s",$unix_time);
```

44. 如何把一个 GB2312 格式的字符串转换成 UTF-8 格式?

```
<?php
iconv('GB2312','utf-8','千锋教育PHP学院');
```

45. 如果需要原样输出用户输入的内容, 在数据入库前, 要用哪个函数处理? htmlspecialchars 或者 htmlentities

46. 写出五种以上你使用过的 PHP 的扩展的名称 (提示: 常用的 PHP 扩展) mb\_string、iconv、curl、GD、XML、socket、MySQL、PDO 等

47. 了解 MVC 模式吗? 请写出三种以上目前 PHP 流行的 MVC 框架名称 (不区分大小写)

FleaPHP、Zend Framework、CakePHP、Symfony、ThinkPHP、YII、CodeIgniter 等

48. php 中 WEB 上传文件的原理是什么, 如何限制上传文件的大小?

上传文件的表单使用 post 方式, 并且要在 form 中添加 enctype='multipart/form-data'。一般可以加上隐藏域: <input type=hidden name=MAX\_FILE\_SIZE value=dddddd>, 位置在 file 域前面。value 的值是上传文件的客户端字节限制。可以避免用户在花等待上传大文件之后才发现文件过大上传失败的麻烦。

使用 file 文件域来选择要上传的文件, 当点击提交按钮之后, 文件会被上传到服务器中的临时目录, 在脚本运行结束时会被销毁, 所以应该在脚本结束之前, 将其移动到服务器上

的某个目录下，可以通过函数 `move_uploaded_file()` 来移动临时文件，要获取临时文件的信息，使用 `$_FILES`。

限制上传文件大小的因素有：

- 1) 客户端的隐藏域 `MAX_FILE_SIZE` 的数值（可以被绕开）。
- 2) 服务器端的 `upload_max_filesize`，`post_max_size` 和 `memory_limit`。这几项不能够用脚本来设置。
- 3) 自定义文件大小限制逻辑。即使服务器的限制是能自己决定，也会有需要个别考虑的情况。所以这个限制方式经常是必要的。

#### 49. 简述 UBB code 的实现原理。(YG)

UBB 代码是 HTML 的一个变种，通过程序自定义我们的标签，比如“`[a]PHP 中 UBB 的使用[/a]`”这样的标签，其实质就是查找 `[a][/a]` 标签，将其替换成 `<a></a>` 的标准 html，说白了，就是将标准的 html 标记通过技术手段使其简化，其输出出来的结果还是标准的 html。

明白了 ubb 的原理，那么再制作一个简单的 ubb 编辑器就不难了，和 fck 之类的编辑器比较起来，ubb 代码最大的优点就是代码简单，功能很少，简单的 ubb 只需要一个文件，而且 ubb 标签可以自己来定义，更改起来很方便，在 php 中就是利用替换函数就可以将 html 进行标签化，输出时进行标签的转化。

#### 50. 怎么把文件保存到指定目录？怎么避免上传文件重名问题？

可以自己设置上传文件的保存目录，与文件名拼凑形成一个文件路径，使用 `move_uploaded_file()`，就可以完成将文件保存到指定目录。可以通过上传的文件名获取到文件后缀，然后使用时间戳+随机数+文件后缀的方式为文件重新命名，这样就避免了重名。

#### 49. \_\_\_\_\_ 函数能返回脚本里的任意行中调用的函数的名称。该函数同时还经常被用在调试中，用来判断错误是如何发生的。（奇矩互动） `debug_print_backtrace()`

#### 50. 在 Smarty 模板语法中如何获取当前时间，并且使用 `Y-m-d H:i:s` 的格式输出？（亿邮）使用 `{ $smarty.now }` 来获取当前时间，得到的是 unix 系统时间戳

使用变量调节器进行格式化，如下：

```
{ $smarty.now | date_format : "%Y-%m-%d %H:%M:%S" }
```

#### 55. 假如让你来写一个函数实现 `Utf-8 转 gb2312`，那么函数的名称应该怎么命名？（亿邮） `utf8_to_gb2312` 或者 `utf8togb2312`

## 56. 请描述如下 URL 重写规则的用意。(卓望)

```
<IfModule mod_rewrite.c>
    RewriteEngine on
    RewriteCond %{REQUEST_FILENAME} !-f
    RewriteCond %{REQUEST_FILENAME} !-d
    RewriteBase /
    RewriteRule ./index.php [L]
</IfModule>
```

如果 REQUEST\_FILENAME 文件存在，就直接访问文件，不进行下面的 rewrite 规则，如果 REQUEST\_FILENAME 目录存在，就直接访问目录，不进行下面的 rewrite 规则，

RewriteRule ./index.php [L] 的意思是把所有的请求都给 index.php 处理。

## 57. Warning: Cannot modify header information-headers already sent by (output started at

D:\src\init.php:7) in D:\src\init.php on line 10 通常什么情况下 php 会报该警告信息？(卓望) 一般是在 header、set\_cookie 以及 session\_start 函数前面有输出（包括空格）的情况下，会报该警告信息。

# 数据结构和算法

## 58. 使对象可以像数组一样进行循环，要求属性必须是私有。(Iterator 模式的 PHP5 实现，写一类实现 Iterator 接口) (腾讯)

```
class Test implements
    Iterator { private $item = array(
        'id' => 1,
        'name' => 'php'
    );

    public function rewind() { reset($this->item);
    }

    public function current() { return
        current($this->item);
    }
```

foreach  
模式的  
PHP5

```
public function key(){ return
    key($this->item);
}

public function next(){ return
    next($this->item);
}

public function valid(){ return ($this-
    >current() !== false);
}
}
//测试
$t = new Test; foreach
($t as $k => $v){
    echo $k,'--->',$v,'<br />';
}
```

### 59. 用 PHP 实现一个双向队列（腾讯）

```
class Deque{ private $queue = array();
    public function addFirst($item){
        return array_unshift($this->queue,
            $item);
    }

    public function addLast($item){
        return array_push($this->queue, $item);
    }

    public function removeFirst(){ return
        array_shift($this->queue);
    }

    public function removeLast(){ return
        array_pop($this->queue);
    }
}
```

60. 请使用冒泡排序法对以下一组数据进行排序 10 2 36 14 10 25 23 85 99 45。

```

1  <?php
2  //冒泡排序
3  function bubble_sort(&$arr){
4      for ($len = count($arr),$i = 0;$i < $len; $i++) {
5          for ($j = 1; $j < $len - $i; $j++) {
6              if ($arr[$j - 1] > $arr[$j]) {
7                  $temp = $arr[$j - 1];
8                  $arr[$j - 1] = $arr[$j];
9                  $arr[$j] = $temp;
10             }
11         }
12     }
13 }
14 $arr = array(10, 2, 36, 14, 10, 25, 23, 85, 99, 45);
15 bubble_sort($arr);
16 print_r($arr);

```

61. 写出一种排序算法（要写出代码），并说出优化它的方法。（新浪）

//快速排序

```

function partition(&$arr,$low,$high){ $pivotkey =
    $arr[$low]; while($low < $high){ while($low < $high
    && $arr[$high] >= $pivotkey){
        $high--;
    }
    $temp = $arr[$low];
    $arr[$low] = $arr[$high];
    $arr[$high] = $temp;
    while($low < $high && $arr[$low] <=
    $pivotkey){ $low++;
    }
    $temp = $arr[$low];
    $arr[$low] = $arr[$high];
    $arr[$high] =
    $temp; }
    return $low;
}

```

```

function
quick_sort(&$arr,$low,$high){ if($low <
$high){

```



```

    $pivot = partition($arr,$low,$high); quick_sort($arr,$low,$pivot-
    1);
    quick_sort($arr,$pivot+1,$high);
  }
}

```

该算法是通过分治递归来实现的，其效率很大程度上取决于参考元素的选择，可以选择数组的中间元素，也可以随机得到三个元素，然后选择中间的那个元素（三数中值法）。

另外还有一点，就是当我们在分割时，如果分割出来的子序列的长度很小的话（小于 5 到 20），通常递归的排序的效率就没有诸如插入排序或希尔排序那么快了。因此可以会去判断数组的长度，如果小于 10 的话，直接用插入排序，而不再递归调用这个快速排序。

5. 一群猴子排成一圈，按 1, 2, ..., n 依次编号。然后从第 1 只开始数，数到第 m 只,把它踢出圈，从它后面再开始数，再数到第 m 只，在把它踢出去...，如此不停的进行下去，直到最后只剩下一只猴子为止，那只猴子就叫做大王。要求编程模拟此过程，输入 m、n, 输出最后那个大王的编号。（新浪）（小米）这是著名的约瑟夫环问题方案一，使用 php 来模拟这个过程

```

20 function king($n,$m){
21     $monkey = range(1, $n);
22     $i = 0;
23     while (count($monkey) > 1) {
24         $i += 1;
25         $head = array_shift($monkey); //一个个出列最前面的猴子
26         if ($i % $m != 0){
27             //如果不是m的倍数，则把该猴子返回尾部，否则就抛掉，也就是出列了
28             array_push($monkey, $head);
29         }
30     }
31     //剩下的最后一个就是大王了
32     return $monkey[0];
33 }
34 //测试
35 echo king(10,7);

```

方案二，使用数学方法解决

```

8 function josephus($n,$m) {
9     $r = 0;
10    for($i = 2; $i <= $n; $i++) {
11        $r = ($r + $m) % $i;
12    }
13    return $r+1;
14 }
15 print_r(josephus(10,7));

```

6. 写一个二维数组排序算法函数，能够具有通用性，可以调用 php 内置函数。

//二维数组排序， \$arr 是数据， \$keys 是排序的键值， \$order 是排序规则， 1 是降序， 0 是升序 function array\_sort(\$arr, \$keys, \$order=0) {

```
    if (!is_array($arr)) {
        return false;
    }
    $keysvalue = array();
    foreach($arr as $key => $val) {
        $keysvalue[$key] = $val[$keys];
    } if($order==0){

    asort($keysvalue);
    }else {
        arsort($keysvalue);
    }
    reset($keysvalue);
    foreach($keysvalue as $key => $vals) {
        $keysort[$key] = $key;
    }
    $new_array = array();
    foreach($keysort as $key => $val) {
        $new_array[$key] = $arr[$val];
    }
    return $new_array;
}
//测试
$person = array( array('id' => 2, 'name' => 'zhangsan',
    'age' => 23), array('id' => 5, 'name' => 'lisi', 'age'
=> 28), array('id' => 3, 'name' => 'apple', 'age' =>
    17)
);
$result = array_sort($person,'name',1);
print_r($result);
```

7. 用二分法查找一个长度为 10 的排好序的线性表，查找不成功时最多需要比较次数是（小米）

4

8. 从 0, 1, 2, 3, 4, 5, 6, 7, 8, 9，这十个数字中任意选出三个不同的数字，“三个数字中不含 0 和 5”的概率是（小米）

7/15

7. 一个三角形三个顶点有 3 只老鼠，一声枪响，3 只老鼠开始沿三角形的边匀速运动，请问他们相遇的概率是（小米）75%，每只老鼠都有顺时针、逆时针两种运动方向，3 只老

鼠共有 8 种运动情况，只有当 3 只老鼠都为顺时针或者逆时针，它们才不会相遇，剩余的 6 中情况都会相遇，故相遇的概率为  $6/8 = 75\%$ 。

### 9. 描述顺序查找和二分查找（也叫做折半查找）算法，顺序查找必须考虑效率，对象可以是一个有序数组（小米）

```
1 <?php
2 /**
3  * 顺序查找
4  * @param $arr array 数组
5  * @param $k 要查找的元素
6  * @return mixed 成功返回数组下标，失败则返回-1
7  */
8 function seq_sch($arr, $k){
9     for ($i = 0, $n = count($arr); $i < $n; $i++){
10         if ($arr[$i] == $k){
11             break;
12         }
13     }
14     if ($i < $n){
15         return $i;
16     } else{
17         return -1;
18     }
19 }
20 //测试
21 $arr1 = array(9,15,34,76,25,5,47,55);
22 echo seq_sch($arr1,47); //结果为6
```

```
1 <?php
2 /**
3  * 二分查找，要求数组已经排好序
4  * @param $arr array 数组
5  * @param $low int 数组起始元素下标
6  * @param $high int 数组末尾元素下标
7  * @param $k 要查找的元素
8  * @return mixed 成功返回数组下标，失败则返回-1
9  */
10 function bin_sch($array, $low, $high, $k){
11     if ($low <= $high){
12         $mid = intval(($low + $high) / 2);
13         if ($array[$mid] == $k){
14             return $mid;
15         } elseif ($k < $array[$mid]){
16             return bin_sch($array, $low, $mid - 1, $k);
17         } else {
18             return bin_sch($array, $mid + 1, $high, $k);
19         }
20     }
21     return -1;
22 }
23 //测试
24 $arr1 = array(5,9,15,25,34,47,55,76);
25 echo bin_sch($arr1,0,7,47); //结果为5
```

10. 我们希望开发一款扑克游戏，请给出一套洗牌算法，公平的洗牌并将洗好的牌存储在一个整形数组里。（鑫众人云）没有固定答案，简单洗牌算法参考

```
$card_num=54; //牌数 function
wash_card($card_num){
    $cards=$tmp=array();
    for($i=0;$i<$card_num;$i++){
        $tmp[$i]=$i;
    }
    for($i=0;$i<$card_num;$i++){
        $index=rand(0,$card_num-$i-1);
        $cards[$i]=$tmp[$index];
        unset($tmp[$index]);
        $tmp=array_values($tmp);
    }
    return $cards;
}
测试： print_r(wash_card($card_num));
```

12. 写出你所知道的排序方法（亿邮）冒泡排序，快速排序，插入排序，选择排序，

## PHP 网络编程

### 1. 禁用 COOKIE 后 SESSION 还能用吗? (51.com 笔试题)

可以, COOKIE 和 SESSION 都是用来实现会话机制的, 由于 http 协议是无状态的, 所以要想跟踪一个用户在同一个网站之间不同页面的 状态, 需要有这么一个机制----会话机制。

COOKIE: 将会话信息的保存到浏览器端。

SESSION: 将会话信息保存到服务器端。

SESSION 默认情况下是基于 COOKIE 的, 对于 SESSION 来说, 每生成一个 SESSIONID, 都会将其发送到浏览器端, 让后将其保存到 cookie 当中。

如果禁用了 COOKIE, 则基于 COOKIE 的 SESSION 不好使了, 我们可以使用 get, 传递 SID, 或者直接开启透明的 SID (此时需要关闭基于 cookie 的 SESSION 配置项)。

相关题目: **SESSION 与 COOKIE 的区别?** COOKIE 保存在客户端, 而 SESSION 则保存服务器端。从安全性的角度来讲, SESSION 的安全性要高。从保存内容的类型的角度来讲, COOKIE 只保存字符串 (及能够自动转换成字符串),

而 session 则可以保存所有的数据类型。

从保存内容的大小的角度来讲, COOKIE 保存的内容是有限制的, 比较小, 而 SESSION 基本上没有这个限制。

从性能的角度来讲, 用 SESSION 的话, 对服务器的压力会更大一些。

### 2. 请使用 socket 相关函数 (非 curl) 实现如下功能: 构造一个 post 请求, 发送到指定 http server 的指定端口的指定请求路径 (如 http://www.example.com:8080/test)。请求中包含以

下变量:

用户名 (username): 温柔一刀密码

(pwd): &123=321&321=123& 个人简介

(intro): Hello world!

且该 http server 需要以下 cookie 来进行简单的用户动作跟踪:

cur\_query: you&me

last\_tm: ... (上次请求的 unix 时间戳, 定为当前请求时间前 10 分钟) cur\_tm: ... (当前请求的 unix 时间戳)

设置超时为 10 秒, 发出请求后, 将 http server 的响应内容输出 (腾讯) 参考代码:

```
if($fp = fsockopen('localhost', '80')) {
```

```
//连接成功
```

```
    $request_data = "POST /2.php HTTP/1.1" . "\r\n"; //请求行
```

```
    $request_data .= "Host: localhost" . "\r\n"; //host 头信息
```

```
    $request_data .= "User-Agent:Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1;
```

```
Trident/5.0)" . "\r\n"; //host 头信息
$request_content = "username=".urlencode('温柔一刀
')." &pwd=".urlencode('&123=321&321=123&')." &intro=".urlencode("Hello world!");
$request_data .= "Content-Type: application/x-www-form-urlencoded" . "\r\n";
$request_data .= "Content-Length: " . strlen($request_content) . "\r\n";
$cur_query = urlencode("you&me");
$last_tm = time()-10*60;
$cur_tm = time();
$request_data .= "Cookie:cur_query=$cur_query;last_tm=$last_tm;cur_tm=$cur_tm" . "\r\n";
$request_data .= "\r\n"; //请求头信息结束时的空行
//请求主体数据部分
$request_data .= $request_content;
//利用建立好的通道，将数据发送过去
fwrite($fp, $request_data); //写入数据（发送数据）
//读取数据

while(!feof($fp)) { //判断是否没有到文件末尾
    $resp_data = fgets($fp);
    echo $resp_data . '<br>';
}
}
```

3. 在 HTTP 1.0 中，状态码 401 的含义是（未经授权：访问由于凭据无效被拒绝）；如果返回“找不到文件”的提示，则可用

**header** 函数，其语句为

```
(header("HTTP/1.0404 Not Found");
```

4. 如何修改 SESSION 的生存时间？要注意这里的生存时间指的是什么？

由于 session 默认是基于 cookie 的，也就是说使用 session 会话技术，首先是将 session 数据保存到服务器端，其次会将 sessionID 保存到浏览器端，保存在服务器端的 session 文件生命周期由 php.ini 中的 session.gc\_maxlifetime、gc\_probability 和 gc\_divisor 来决定，而保存在客户端的 sessionID 由客户端 cookie 来决定，默认其生存周期直到浏览器关闭，它也可以通过 php.ini 中的设置 session.cookie\_lifetime 来控制，二者共同来决定 session 的生存时间，二者中的其中任意一个失效了，就会造成 session 失效，要根据自己的需求来决定是要从哪方面来修改其生存时间。

参考 1:将 php.ini 中的 session.gc\_maxlifetime 设置为 9999 重启 apache

参考 2:

```
$savePath = "./session_save_dir/";
```

```
$lifeTime = 小时 * 秒;
```

```
session_save_path($savePath);
session_set_cookie_params($lifeTime);
session_start(); 参考 3:
session_start();
$lifeTime = 24 * 3600; // 保存一天
setcookie(session_name(), session_id(), time() + $lifeTime, "/");
```

## 5. How can you get round the stateless nature of HTTP using PHP? (Yahoo)

最主要的两个选择是 SESSION 和 COOKIE。使用 SESSION 的方法是在每一页的开始加 session\_start(), 然后利用\$\_SESSION 来存取 SESSION 变量。至于 COOKIE 你只需记着一个原则: 在输出任何文字之前调用 set\_cookie() 函数设置 COOKIE, 使用 \$\_COOKIE 获取 COOKIE 变量。

## 6. PHP 程序中如何设置当前页面的编码为 utf-8 ?

```
header("content-type:text/html;charset=utf-8");
```

## 7. 如何设置一个 cookie 的名字为 username, 值为 jack, 并且让此 cookie 一周后失效?

```
setcookie('username','jack',time()+7*24*3600);
```

## 8. 一个浏览器最多可以产生多少个 cookie, 每个 cookie 文件最大不能超过多少?

cookie 的总数量没有限制, 但是每个域名的 COOKIE 数量和每个 COOKIE 的大小是有限制的!

IE 每个域名限制为 50 个。

Firefox 每个域名 cookie 限制为 50 个。

Opera 每个域名 cookie 限制为 30 个。

Safari/webkit 貌似没有 cookie 限制。但是假如 cookie 很多, 则会使 header 大小超过服务器的处理的限制, 会导致错误发生。不同浏览器间每个 cookie 文件大小也不同

Firefox 和 safari 是 4097 个字节, 包括名 (name)、值 (value) 和等号。

Opera 是 4096 个字节, 包括: 名 (name)、值 (value) 和等号。

IE 是 4095 个字节, 包括: 名 (name)、值 (value) 和等号。

## 9. 设置或读取 session 之前, 需要做什么?

session\_start()前面不能有任何输出，包括空行。

相关题目：使用 **setcookie** 函数前，需要注意什么？

前面不能有任何输出，包括空行。

## 10. 请描述出七层网络模型的名称，由下到上（可以使用中文描述）

物理层、数据链路层、网络层、传输层、会话层、表示层、应用层

## 11. 说说下面这些这些协议的全称和中文解释（提示：都是工作在应用层）SMTP、POP3、HTTP、FTP、DNS（小米）

SMTP (Simple Mail Transfer Protocol) 简单邮件传输协议

POP3 (Post Office Protocol 3) 邮局协议第 3 版

HTTP (Hypertext Transfer Protocol) 超文本传输协议

FTP (File Transfer Protocol) 文件传输协议

DNS (Domain Name System and Domain Name Service protocol) 域名系统（服务）协议

## 12. COOKIE、SESSION 的联系和区别，多台 web 服务器如何共享 SESSION?

COOKIE 和 SESSION 都是用于会话机制，COOKIE 保存在客户端，而 SESSION 则保存在服务器端。在默认情况下，SESSION 机制是基于 COOKIE 的，每生成一个 SESSIONID，都会将其发送到浏览器端，然后将其保存到 COOKIE 当中，在下次请求的时候，由浏览器携带这个 COOKIE。要想多台 web 服务器共享 SESSION，可以利用 MySQL 数据库存储 SESSION 数据。

## 13. HTTP/1.0 中，状态码的含义（小米）

1\*\*：（临时响应）表示临时响应并需要请求者继续执行操作的状态代码

2\*\*：操作成功收到，分析、接受

3\*\*：(重定向) 表示要完成请求，需要进一步操作。通常，这些状态码用来重定向。

4\*\*：(请求错误) 这些状态代码表示请求可能出错，妨碍了服务器的处理。

5\*\*：(服务器错误) 这些状态代码表示服务器在尝试处理请求时发生内部错误。这些错误可能是服务器本身的错误，而不是请求出错。

以下几个仅仅是随意列出的，其他的也需要记忆。

200 OK 服务器成功处理了请求

301 Moved Permanently（重定向）请求的 URL 已移走

304 Not Modified（未修改）客户的缓存资源是最新的，要客户端使用缓存

403 Forbidden（禁止）请求被服务器拒绝了

404 Not Found 未找到资源

500 (内部服务器错误)服务器遇到一个错误，使其无法为请求提供服务



#### 14. 请描述 PHP(或其他语言)Session 的运行机制，大型网站中 Session 方面应注意什么？（小米）

session 机制是一种服务器端的机制，它将数据保存到服务器端。

当程序需要为某个客户端的请求创建一个 session 的时候，服务器首先检查这个客户端的请求里是否已包含了一个 session 标识，称为 sessionid，如果已包含一个 sessionid 则说明以前已经为此客户端创建过 session，服务器就按照 sessionid 把这个 session 检索出来使用（如果检索不到，可能会新建一个），如果客户端请求不包含 sessionid，则为此客户端创建一个 session 并且生成一个与此 session 相关联的 sessionid，sessionid 的值应该是一个既不会重复，又不容易被找到规律以仿造的字符串，这个 sessionid 将被在本次响应中返回给客户端以 cookie 的形式保存。

session 在大访问量网站上确实影响系统性能，影响性能的原因之一由文件系统设计造成，在同一个目录下超过 10000 个文件时，文件的定位将非常耗时，另外就是小文件的效率问题，一般我们的 session 数据都不会太大（1~2K），如果有大量这样 1~2K 的文件在磁盘上，IO 效率肯定会很差，我们可以采用改写 session 存储机制，比如存放到数据库中。

#### 15. 在子网掩码为 255.255.255.248 的局域网中能够同时使用的 IP 有多少个？（亿邮）

6 个，255.255.255.248 换算成二进制就是 11111111.11111111.11111111.11111000 后面的三位用来表示主机，前面的 29 位用来表示网络。

三位表示主机，就是可以用来表示  $2^3=8$  个，再减去网络地址和广播地址，就是 6 个了。

#### 16. 简述 Tcp 协议的三次握手过程。（亿邮）

TCP 是主机对主机层的传输控制协议，提供可靠的连接服务，采用三次握手确认建立一个连接：

第一次握手：建立连接时，客户端发送 syn 包(syn=j)到服务器，并进入 SYN\_SEND 状态，等待服务器确认；

第二次握手：服务器收到 syn 包，必须确认客户的 SYN (ack=j+1)，同时自己也发送一个 SYN 包 (syn=k)，即 SYN+ACK 包，此时服务器进入 SYN\_RECV 状态；

第三次握手：客户端收到服务器的 SYN + ACK 包，向服务器发送确认包 ACK(ack=k+1)，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

完成三次握手，客户端与服务器开始传送数据。

#### 17. 你会用哪个函数设置当前内容的 Content-Type？（卓望）

header 函数

2. 通过页面输入用户名 **abc** 和密码 **123** 登录到**www.10086.cn**，请写出该次请求的**HTTP协议报文(包括请求行、消息报头、请求正文)**。(卓望)

```
POST www.10086.cn HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Referer: http://localhost/a.html
Accept-Language: zh-CN
User-Agent: Mozilla/5.0 (compatible; MSIE 10.0; Windows NT 6.1; WOW64; Trident/6.0;
KB974487)
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
Host: localhost
Content-Length: 25

Connection: Keep-Alive

username=abc&password=123
```

## Linux 部分

### 3. 1. 请解释下列 10 个 shell 命令的用途（新浪网技术部）

top、ps、mv、find、df、cat、chmod、chgrp、grep、wc

top: 该命令提供了实时对系统处理器状态的监控, 它能够实时显示系统中各个进程的资源占用情况。该命令可以按照对 CPU、内存使用和执行时间对系统任务进程进行排序显示, 同时还可以通过交互式命令进行设定显示。

ps: 显示系统进程在瞬间的运行动态。mv: 文件/目录改名或变更存储位置。find: 在指定的路径下查找指定文件。

df: 检查磁盘空间占用情况。

cat: 将文件的内容打印到标准输出。

chmod: 改变文件的权限。chgrp: 改变文件所属组。

grep: 过滤文本, 根据指定的字符串, 对文件的每一行进行搜索, 如找到, 则输出该行内容。

wc: 统计指定文件中的字节数、字数、行数, 并将统计结果显示输出。

相关题目: 写出 15 个以上你所知道的常用的 Linux 命令和它的功能。

ls: 列出目录

cp: 复制 rm: 删除

cat: 将文件的内容打印到标准输出

mkdir: 建立目录

tar: 打包压缩 ps: 查看进程

top: 查看机器使用情况

df: 检查磁盘空间占用情况

find: 在指定路径下查找指定文件 grep: 过滤文本

cd: 改变当前工作目录 mount: 挂载/卸载指定的文件系统

ifconfig: 配置网络或显示当前网络接口状态 telnet:: 远程登录

### 4. vi编辑器中, 选中、复制、粘贴、删除的命令各是什么(酷讯)

选中: v (以字符为单位), V (以行为单位) 复制: y, 如 yy, nyy, y1G, yG, y0, y\$

粘贴: p (粘贴在光标之后), P (粘贴在光标之前) 删除: d, 如 dd, ndd, d1G, dG, d\$,d0

## 5. 获取文件行数 (酷讯)

```
wc -l filename
```

## 6. 输入文件的最后 5 行到另一个文件中 (酷讯)

```
tail -n 5 file1 >> file2
```

## 7. 查找文件中包含 hello 的行 (酷讯)

```
grep hello filename
```

## 8. 查找当前目录下所有目录名为CVS的子目录的命令 (酷讯)

```
find ./CVS -maxdepth 1 -type d -print
```

## 9. 如何让一个程序在后台运行并把输入定向到指定的文件 (酷讯)

可使用命令: `nohup 程序 >> 文件名 2>&1 &` (注意, `2>&1` 要放在输出文件名的后面)

## 10. 如何把一个文件的内容添加到另一个文件的末尾 (酷讯)

```
cat file1 >> file2
```

## 11. 如何实时的显示一个文件的输出 (酷讯)

使用 `more` 或者 `less`

## 12. 定时执行一个程序的方法有什么 (酷讯)

可以使用 `at` 或者 `crontab`, 其中 `at` 是处理仅执行一次就结束调度的命令, 适用于突发性工作, 而 `crontab` 将会循环一直进行下去, 适用于例行性工作。

## 13. vi 编辑器中, 如何替换指定的字符串 (酷讯)

使用 `s` 命令, 例如:

: `n1,n2s/word1/word2/g`, 将第 `n1` 行与 `n2` 行之间的 `word1` 替换为 `word2`

: 1,\$s/word1/word2/g, 从第一行到最后一行中的 word1 替换为 word2

#### 14. 当更新后，cvs中文件有冲突时。如何判断哪些你编辑的内容和更新下来的内容(酷讯)

使用 CVS 时，如果出现冲突，双击冲突的文件（标识为红色），显示文件比较窗口。其中蓝色为需要传入的更新，其中灰色为需要传出的修改，其中红色为需要手工解决的冲突。

可以首先将蓝色需要传入的更新，然后手工解决红色部分，自行选择保留左边还是右边，并选择 mark as merge 将文件标志为合并，此时将以左边窗口中的内容为依据向服务器提示传出，即灰色和红色部分都会以左边窗口的内容为依据向服务器提交。

如果冲突文件打开无任何的颜色提示，可选择 update and override，使用服务器上的文件更新本地文件。

#### 15. Linux下，如何查看所有用户\_\_\_\_\_

```
cat /etc/passwd
```

#### 16. Linux 上的 crond 服务可对服务器设置计划任务，请问如果我需设置每 2 分钟执行一次 /tmp/test.sh 的脚本应如何设置计划任务，请写出操作方法。

首先用 root 用户的权限登录。

切换到/etc/init.d/目录下，使用命令：crond start 启动 crond 服务切换到/etc/目录下，使用命令： crontab -e 编辑如下：\*/2 \* \* \* \* /tmp/test.sh

#### 17. 查看磁盘空间使用率的 Linux 命令是什么？查看有哪些系统进程正在运行命令是什么？检测 Linux 性能（cpu，磁盘 io，内存，网络等）都用到哪些命令？（卓望）

查看磁盘空间使用率的命令：df

查看系统进程命令：top

检测 Linux 性能的命令：ps, free, vmstat, netstat

#### 18. Linux 下终止一个进程用什么命令？打包压缩和解包用什么命令？软连接和硬链接有什么区别？建立软连接的命令是？（卓望）

终止进程的命令：kill 打包压缩和解包命令：tar 建立软连接的命令：ln -s 源文件 目标链接名

硬链接是通过文件系统的 `inode` 来产生新文件名，而不是新文件，它只是在某个目录新建一条文件名链接到某 `inode` 号码的管理记录而已，而软链接是创建一个独立的文件，该文件会让数据的读取指向它链接的那个文件的文件名，类似于 `windows` 中的快捷方式。

## 19. 说说 Linux 下的 `find` 命令和 `grep` 命令的区别

`find`: 在磁盘/分区中找到文件，可以配 `type` 可以配 `size time` 等，通过文件名或文件大小或访问时间找到指定文件。

`grep`: 查找文件里符合条件的字符串，并把匹配的行打印出来，可以使用正则表达式。

## 20. 有 `mail.log` 的一个文档，内容为若干邮件地址，其中用 `\n` 将邮件地址分隔。要求从中挑选出 `sina.com` 的邮件地址（包括从文件读取、过滤到列印出来）。

`mail.log` 内容如下：

`james@sina.com.cn`

`jack@163.com`

`zhansan@sohu.com`

`lisi@hotmail.com`

`wangwu@gmail.com` 参考答

案：

```
cat mail.log | grep sina.com
```

或者：

```
#!/bin/bash
```

```
while read row ;
```

```
do case $row in
```

```
    *sina.com*) echo
```

```
$row ;; esac done < mail.log
```

## 21、简述 DNS 进行域名解析的过程。

参考答案：

首先，客户端发出 DNS 请求翻译 Ip 地址或主机名。DNS 服务器在收到客户机的请求后：

(1)检查 DNS 服务器的缓存，若查到请求的地址或名字，即向客户机发出应答信息；

(2)若没有查到，则在数据库中查找，若查到请求的地址或名字，即向客户机发出应答信息；

(3)若没有查到，则将请求发给根域 DNS 服务器，并依序从根域查找顶级域，由顶级查找二级域，二级域查找三级，直至找到要解析的地址或名字，即向客户机所在网络的

DNS 服务器发出应答信息，DNS 服务器收到应答后现在缓存中存储，然后，将解析结果发给客户机。

(4)若没有找到，则返回错误信息。

## 22. 简述 Linux 文件系统通过 i 节点把文件的逻辑结构和物理结构转换的工作过程。

参考答案：

Linux 通过 i 节点表将文件的逻辑结构和物理结构进行转换。

i 节点是一个 64 字节长的表，表中包含了文件的相关信息，其中有文件的大小、文件所有者、文件的存取许可方式以及文件的类型等重要信息。在 i 节点表中最重要的内容是磁盘地址表。在磁盘地址表中有 13 个块号，文件将以块号在磁盘地址表中出现的顺序依次读取相应的块。Linux 文件系统通过把 i 节点和文件名进行连接，当需要读取该文件时，文件系统在当前目录表中查找该文件名对应的项，由此得到该文件相对应的 i 节点号，通过该 i 节点的磁盘地址表把分散存放的文件物理块连接成文件的逻辑结构。

## 23. 简述进程的启动、终止的方式以及如何进程的查看。

参考答案：

在 Linux 中启动一个进程有手工启动和调度启动两种方式：

### (1)手工启动

用户在输入端发出命令，直接启动一个进程的启动方式。可以分为：

①前台启动：直接在 SHELL 中输入命令进行启动。

②后台启动：启动一个目前并不紧急的进程，如打印进程。

### (2)调度启动

系统管理员根据系统资源和进程占用资源的情况，事先进行调度安排，指定任务运行的时间和场合，到时候系统会自动完成该任务。

经常使用的进程调度命令为：at、batch、crontab。

## 24. 什么是静态路由，其特点是什么？什么是动态路由，其特点是什么？

参考答案：

静态路由是由系统管理员设计与构建的路由表规定的路由。适用于网关数量有限的场合，且网络拓扑结构不经常变化的网络。其缺点是不能动态地适用网络状况的变化，当网络状况变化后必须由网络管理员修改路由表。

动态路由是由路由选择协议而动态构建的，路由协议之间通过交换各自所拥有的路由信息实时更新路由表的内容。动态路由可以自动学习网络的拓扑结构，并更新路由表。其缺点是路由广播更新信息将占据大量的网络带宽。

## 25. 什么是符号链接，什么是硬链接？符号链接与硬链接的区别是什么？

参考答案：

链接分硬链接和符号链接。

符号链接可以建立对于文件和目录的链接。符号链接可以跨文件系统，即可以跨磁盘分区。符号链接的文件类型位是 l，链接文件具有新的 i 节点。

硬链接不可以跨文件系统。它只能建立对文件的链接，硬链接的文件类型位是 -，且硬链接文件的 i 节点同被链接文件的 i 节点相同。

## 26. 简述网络文件系统 NFS，并说明其作用。

参考答案：

网络文件系统是应用层的一种应用服务，它主要应用于 Linux 和 Linux 系统、Linux 和 Unix 系统之间的文件或目录的共享。对于用户而言可以通过 NFS 方便的访问远地的文件系统，使之成为本地文件系统的一部分。采用 NFS 之后省去了登录的过程，方便了用户访问系统资源。

## 27. 简述使用 ftp 进行文件传输时的两种登录方式？它们的区别是什么？常用的 ftp 文件传输命令是什么？

参考答案：



(1) ftp 有两种登录方式：匿名登录和授权登录。使用匿名登录时，用户名为：anonymous，密码为：任何合法 email 地址；使用授权登录时，用户名为用户在远程系统中的用户帐号，密码为用户在远程系统中的用户密码。

区别：使用匿名登录只能访问 ftp 目录下的资源，默认配置下只能下载；而授权登录访问的权限大于匿名登录，且上载、下载均可。

(2) ftp 文件传输有两种文件传输模式：ASCII 模式和 binary 模式。ASCII 模式用来传输文本文件，其他文件的传输使用 binary 模式。

(3) 常用的 ftp 文件传输命令为：bin、asc、put、get、mput、mget、prompt、bye

## 28. 解释 i 节点在文件系统中的作用。

参考答案：

在 linux 文件系统中，是以块为单位存储信息的，为了找到某一个文件在存储空间中存放的位置，用 i 节点对一个文件进行索引。I 节点包含了描述一个文件所必须的全部信息。所以 i 节点是文件系统管理的一个数据结构。

## 29. 当文件系统受到破坏时，如何检查和修复系统？

参考答案：

成功修复文件系统的前提是要有两个以上的主文件系统，并保证在修复之前首先卸载将被修复的文件系统。

使用命令 fsck 对受到破坏的文件系统进行修复。fsck 检查文件系统分为 5 步，每一步检查系统不同部分的连接特性并对上一步进行验证和修改。在执行 fsck 命令时，检查首先从超级块开始，然后是分配的磁盘块、路径名、目录的连接性、链接数目以及空闲块链表、i-node。

## 30. Linux 设备中字符设备与块设备有什么主要的区别？请分别列举一些实际的设备说出它们是属于哪一类设备

**字符设备：**字符设备是个能够像字节流（类似文件）一样被访问的设备，由字符设备驱动程序来实现这种特性。字符设备驱动程序通常至少实现 `open`,`close`,`read` 和 `write` 系统调用。字符终端、串口、鼠标、键盘、摄像头、声卡和显卡等就是典型的字符设备。

**块设备：**和字符设备类似，块设备也是通过 `/dev` 目录下的文件系统节点来访问。块设备上能够容纳文件系统，如：`u 盘`，`SD 卡`，`磁盘`等。

字符设备和块设备的区别仅仅在于内核内部管理数据的方式，也就是内核及驱动程序之间的软件接口，而这些不同对用户来讲是透明的。在内核中，和字符驱动程序相比，块驱动程序具有完全不同的接口。

### 31. 什么是位置变量？Shell 的变量的类型有哪些种？

位置变量是指命令行上传递给程序的参数

Shell 变量可分为：用户自定义变量、环境变量、位置变量、特殊变量

### 32. 若下达 `rmdir` 命令来删除某个已存在的目录，但无法成功，请说明可能的原因

参考答案：

此目录可能不存在，或者目录不为空，或者目录中有隐含的文件存在，或者目录被设置了权限。

解决办法：

先修改权限，再使用 `rm -r` 命令删除

### 33. 简述解决忘记 root 密码的办法。

参考答案：

1) 用 RedHat 标准安装盘启动系统

当出现提示符时输入：`linux rescue`

就会以修复模式启动系统

2) 系统启动完毕后，会将硬盘 `mount` 到 `/mnt/sysimage`，届时会有提示

3) 进入提示符后，输入命令：

`Chroot/mnt/sysimage`

4) 输入命令：

Passwd 更改密码

5) 连续输入两次 exit, 即重启系统, 将光盘拿出, 按正常模式启动系统

除了用光盘引导 rescue 模式, 还有一种方法就是启动装载器来进入单用户模式

#### (1) Grub

进入启动画面之后, 输入“e”, 把光标移动到 kernel...那一行, 再敲入“e”, 在 kernel 一行的最后加上空格 single, 回车

输入“b”, 启动系统, 即进入单用户模式,

这个时候就可以使用 passwd 命令改密码了。

#### (2) Lilo

进入 lilo 命令行模式, 输入 linux single, 即进入单用户模式。

Passwd...

如果 grub.lilo 设了密码, 一定要记住才行。要是忘了 grub/lilo 的密码, 就只有用 rescue 模式了

### 34. 创建一个 shell 脚本, 他从用户那里接受 10 个数, 并显示已输入的最小的数。

参考程序:

```
#!/bin/bash

Max=0

For((count=1 ;count<=10;count = count+1))

Do

Echo -n "Enter number $count;"

Read value

If[$max -lt $value]

Then

Max=$value

Fi
```

done

## PHP 综合应用

### 1. 写出下列服务的用途和默认端口（新浪网技术部）

ftp、ssh、http、telnet、https

**ftp:** File Transfer Protocol, 文件传输协议, 是应用层的协议, 它基于传输层, 为用户服务, 它们负责进行文件的传输, 其默认端口是 21。

**ssh:** Secure Shell, 安全外壳协议, 建立在应用层和传输层基础上的安全协议。SSH 是目前较可靠, 专为远程登录会话和其他网络服务提供安全性的协议, 其默认端口是 22。

**http:** hypertext transport protocol, 超文本传送协议, 是一种详细规定了浏览器和万维网服务器之间互相通信的规则, 通过因特网传送万维网文档的数据传送协议, 其默认端口是 80。

**telnet:** Telnet 协议是 TCP/IP 协议族中的一员, 是 Internet 远程登陆服务的标准协议和主要方式。它为用户提供了在本地计算机上完成远程主机工作的能力, 其默认端口是 23。

**https:** Hypertext Transfer Protocol over Secure Socket Layer, 是以安全为目标的 HTTP 通道, 用于安全的 HTTP 数据传输, 它的主要作用可以分为两种: 一种是建立一个信息安全通道, 来保证数据传输的安全; 另一种就是确认网站的真实性, 其默认端口是 443。

### 2. 你用什么方法检查 PHP 脚本的执行效率和数据库 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？（腾讯）

脚本执行时间, 启用 xdebug, 使用 WinCacheGrind 分析。

数据库查询, mysql 使用 EXPLAIN 分析查询, 启用 slow query log 记录慢查询。

### 6. 对于大流量的网站，您采用什么样的方法来解决访问量问题？

1) 确认服务器硬件能否支持当前的流量

对于普通的 pc server 来说, 它能够独立支持每天 10 万个独立 ip 访问, 如果访问量过大, 最好更好性能更高的专用服务器。

2) 优化数据库的访问

服务器的负载过大, 一个重要的原因就是 CPU 和内存负载过高, 而读写数据在这块占据较多的资源。可以从页面静态化、memcache 缓存和 mysql 优化几个方面着手。

3) 禁止外部盗链

占用较大的流量, 防盗链, 使用 reference 来判断一下。如果是图片的话, 使用添加水印即可很好的防止。

4) 控制大文件的下载

最好把文件下载的容量控制为相对较小的一个值，如果有大文件下载，最好使用专用的服务器。

- 5) 使用多台主机实现分流，集群
- 6) 使用流量分析软件进行分析统计 谷歌 和百度

## 7. 谈谈 asp, php, jsp 的优缺点

ASP 全名 Active Server Pages，是一个 WEB 服务器端的开发环境，利用它可以产生和运行动态的、交互的、高性能的 WEB 服务应用程序。ASP 采用脚本语言 VB Script (JavaScript) 作为自己的开发语言。

PHP 是一种跨平台的服务器端的嵌入式脚本语言。它大量地借用 C、Java 和 Perl 语言的语法，并结合自己的特性，使 WEB 开发者能够快速写出动态生成页面。它支持目前绝大多数数据库。还有一点，PHP 是完全免费的，不用花钱，你可以从 PHP 官方站点自由下载。而且你可以不受限制地获得源码，甚至可以从中加进你自己需要的特色。

JSP 是 Sun 公司推出的新一代站点开发语言，他完全解决了目前 ASP 和 PHP 的一个通病-----脚本级执行（据说 PHP4 也已经在 Zend 的支持下，实现编译运行）。Sun 公司借助自己在上的不凡造诣，将 Java 从 Java 应用程序 和 Java Applet 之外，又有新的硕果，就是 Java Server Page。JSP 可以在 Servlet 和 JavaBean 的支持下，完成功能强大的站点。

三者都提供在 HTML 代码中混合某种程序代码、由语言引擎解释执行程序代码的能力。但 JSP 代码被编译成 Servlet 并由 Java 虚拟机解释执行，这种编译操作仅在对 JSP 页面的第一次请求时发生。

在 ASP、PHP、JSP 环境下，HTML 代码主要负责描述信息的显示样式，而程序代码则用来描述处理逻辑。普通的 HTML 页面只依赖于 Web 服务器，而 ASP、PHP、JSP 页面需要附加的语言引擎分析和执行程序代码。程序代码的执行结果被重新嵌入到 HTML 代码中，然后一起发送给浏览器。

ASP、PHP、JSP 三者都是面向 Web 服务器的技术，客户端浏览器不需要任何附加的软件支持。

## 8. 请举例说明在你的开发过程中用什么方法来加快页面的加载速度。

要用到服务器资源时才打开，及时关闭服务器资源，数据库添加索引，页面可生成静态，图片等大文件单独服务器，使用代码优化工具等。

## 9. Is PHP better than Perl? - Discuss. (Yahoo)

我们不要为一个简单的问题引发一场舌战，为工作选择适合的语言，不要为工作迁就语言。Perl 十分适合作为命令行工具，虽然它在网页应用上也有不错的表现，但是它的真正实力在命令行上才能充分发挥。同样地，PHP 虽然可以在控制台的环境中使用，但是它在网页

应用上有更好的表现，PHP 有大量专门为网页应用而设计的函数，Perl 则似乎以命令行为设计之本。

## 10. What's the difference between the way PHP and Perl distinguish between arrays and hashes? (Yahoo)

这正是为何我老是告诉别人选择适当的编程语言，若果你只用一种语言的话你怎么能回答这道问题？这道问题很简单，Perl 所变量都是以 @ 开头，例如 @myArray，PHP 则沿用 \$ 作为所有变量的开头，例如 \$myArray。

至于 Perl 表示散列表则用 %，例如 %myHash，PHP 则没有分别，仍是使用 \$，例如 \$myHash。

## 11. How do you debug a PHP application? (Yahoo)

使用 Xdebug 或者 Advanced PHP Debugger

## 13. PEAR 中的数据库连接字符串格式是\_\_\_\_\_。

```
$dsn = 'mysql://username:password@localhost/test'  
$options = array(  
    'debug' => 2,  
    'portability' => DB_PORTABILITY_ALL,  
)
```

DB::connect(\$dsn, \$options)，其中 options 参数是可选的。

PEAR 是 PHP 扩展与应用库 (the PHP Extension and Application Repository) 的缩写。它是一个 PHP 扩展及应用的一个代码仓库，PEAR 处理数据库的模块是 PEAR DB。

## 14. 如何实现 PHP、JSP 交互？

题目有点含糊不清,SOAP,XML\_RPC,Socket function,CURL 都可以实现这些,如果是考虑 PHP 和 Java 的整合,PHP 内置了这种机制(如果考 PHP 和.NET 的整合,也可以这么回答)。

PHP 提供了支持 JAVA 的类库文件, 或者通过 HTTP 协议来交互数据。

## 15. apache+mysql+php 实现最大负载的方法

- 1) 问的太笼统,生成静态 html 页面, squid 反向代理, apache, mysql 的负载均衡。
- 2) 可以采取数据缓存的方法, 我们通常在统计数据的时候, 需要在原始数据的基础上经过计算等一系列操作, 才会得到最终的结果, 如果每做一个查询都需要这样一系列操作,

当数据量大时，势必会带来很多问题。可以建立一个结果表，写一个脚本，用 crontab 定时触发脚本去原始表取数据，计算，写入到结果表，前端查询从结果表取数据，这也是比较常用的一种做法。

- 3) 采用分布式，多个 apache，多个 mysql，其实就是 dns 负载均衡，dns 根据当前用户解析几个 ip 的 ping 值，将用户转移到某一台最快的服务器，或者平均分配。
- 4) money 不是问题的话，可以考虑 F5 硬件负载均衡！
- 5) 可以使用 Microsoft Windows Server 系统的负载均衡设置

## 16. 你觉得在 PV10W 的时候，同等配置下，LINUX 比 WIN 快多少？ (51.com)

不做优化的情况下一样。

## 17. Ajax, 数据库触发器，GUI，中断机制的共同思想。谈一谈该种思想（机制）（百度）

主要就是异步，主进程不会被一个异步任务阻塞，当进程发出命令之后，继续执行主任务，不用等待子任务执行完，这样效率更高。

数据库触发器和中断机制是数据库自动完成的，而 ajax 触发器是用户激发的。ajax 把 GUI 和数据库异步优化。

## 18. 把一篇英文文档中所有单词的首字母转为大写，文档存在 doc.txt 中。可以在多种编程语言中选择（C\C++, JAVA, PHP...）写出你的思路，尽量优化你的程序。（百度）

```
$str = file_get_contents('doc.txt');  
$str = ucwords($str);  
file_put_contents('doc.txt', $str);
```

## 19. 防止 SQL 注射漏洞一般用\_\_\_\_\_函数

addslashes

## 26. 综合运用，PHP+MySQL 编程，文件操作（C/BSI）以下请用 PHPMYADMIN 完成

(1). 创建新闻发布系统，表名为 **message** 有如下字段  
id 文章 id title 文章标题 content 文章内容  
category\_id 文章分类 id hits 点击量

创建表语句如下：

```
CREATE TABLE message( id iNT UNSIGNED NOT NULL  
AUTO_INCREMENT PRIMARY KEY, title VARCHAR(200) NOT  
NULL DEFAULT '', content TEXT,
```

```
category_id INT UNSIGNED NOT NULL DEFAULT 0,  
hits INT UNSIGNED NOT NULL DEFAULT 0  
) engine=InnoDB default charset=utf8
```

(2). 同样上述新闻发布系统: 表 **comment** 记录用户回复内容, 字段如下  
comment\_id 回复 id id 文章 id, 关联 message 表中的 id  
comment\_content 回复内容

现通过查询数据库需要得到以下格式的文章标题列表, 并按照回复数量排序, 回复最高的排在最前面  
文章 id 文章标题 点击量 回复数量

用一个 SQL 语句完成上述查询, 如果文章没有回复则回复数量显示为 0

查询语句如下:

```
SELECT message.id AS id, title, hits, IF(comment.id is NULL,0,count(*)) AS number  
FROM message  
LEFT JOIN comment  
ON message.id = comment.id  
GROUP BY message.id
```

(3). 上述内容管理系统, 表 **category** 保存分类信息, 字段如下

```
category_id int(4) not null auto_increment;  
category_name varchar(40) not null;
```

用户输入文章时, 通过选择下拉菜单选定文章分类写出如何实现这个下拉菜单

```
function categoryList(){  
    $result = mysql_query("select category_id,category_name from category") or die("Invalid  
query: " . mysql_error()); echo "<select name = 'category' value = ">"; while($row =  
mysql_fetch_array($result)){ echo "<option value =  
" . $row['category_id'] . ">" . $row['category_name'] . "</option>";  
    }  
    echo "</select>";  
}
```

文件操作部分: 上述内容管理系统, 用户提交内容后, 系统生成静态 **HTML** 页面, 写出实现的基本思路。

要生成静态 **HTML** 页面, 需要使用输出缓冲 `output buffering` 及文件操作, 首先使用 `ob_start()` 函数开启输出缓冲, 在页面内容执行完成时, 使用 `ob_get_contents()` 函数获取保存在输出缓冲区中的内容, 然后使用 `file_put_contents()` 函数, 生成静态 **HTML** 页面即可。

## 27. 请问 `cgi` 和 `fastcgi` 有什么不同, 你在什么情况下会选择哪个? (酷讯)

原理一样, 都是利用标准输入输出流处理 **HTTP** 之类的文本协议, 都是通过多进程模式处理多请求。不同之处在于 **FastCGI** 的一个进程处理完一个请求之后重置状态并挂起,



待下一个请求来时继续处理；而 CGI 的一个进程则处理完一个请求后退出，下一个请求来时再创建新进程。

## 28. zend optimizer 是什么？（酷讯）

Zend Optimizer 可以加速 PHP 脚本的执行，提高 PHP 应用程序的执行速度。实现的原理是对那些程序在被最终执行之前由运行编译器(Run-Time Compiler)产生的代码进行优化。

一般情况下，执行使用 Zend Optimizer 的 PHP 程序比不使用的要快 40%到 100%。这意味着网站的访问者可以更快的浏览网页，从而完成更多的事务，创造更好的客户满意度。

Zend Optimizer 还可以给用 Zend 加密的文件解密。

## 29. 列举 web 开发中的安全性问题 sql 注入攻击。

数据库操作安全，UPDATE、DELETE、INSERT 的操作没有限制用户操作权限，这将是一件很危险的事情。

没有验证用户 http 请求的方式 POST 或者 GET，GET 请求被合法通过。

没有验证表单来源的唯一性，不能识别是合法的表单提交还是黑客伪造的表单提交。

XSS 攻击。

## 30. 如何通过 php 程序防止外部页面提交表单？编写一段代码

```
<?php
session_start();
if(isset($_POST['name'])
    && !empty($_POST['name'])){ if($_POST['check']
    ==$_SESSION['check']){ echo '正常访问';
    } else {
        echo '外部访问';
    }
}
$token = md5(uniqid(rand(),true));
$_SESSION['check'] = $token;
?>
<form method="post" action="">
    <input type="text" name="name">
    <input type="hidden" name="check" value="<?php echo $token;?>">
    <input type="submit">
</form>
```

## 31. 如果某段与数据库交互的程序运行较慢你将如何处理？

一是首先提高数据库的查询速度，比如增加索引，优化表的结构。

二是优化程序代码，如果查询比较多，可以尽量用条件查询，减少查询语句，比如能用一条查询语句就不用两条。

三就是提高服务器的速度，优化服务器，把不必要的进程关掉。

### 32. 以下代码会产生什么问题，如何解决？

```
<?php
    $dir = $_POST['dir'];
    include("/usr/local/apache/htdocs/inc/$dir ");
?>
```

不安全，必须对用户的输入进行验证和过滤。

### 33. 请简述操作系统的线程与进程的区别。列举 LINUX 下面你使用过的软件？

进程是具有一定独立功能的程序关于某个数据集合上的一次运行活动，进程是系统进行资源分配和调度的一个独立单位。

线程是进程的一个实体是 CPU 调度和分派的基本单位，它是比进程更小的能独立运行的基本单位。

进程和线程的主要差别在于它们是不同的操作系统资源管理方式。进程有独立的地址空间，一个进程崩溃后，在保护模式下不会对其它进程产生影响，而线程只是一个进程中的不同执行路径。线程有自己的堆栈和局部变量，但线程之间没有单独的地址空间，一个线程死掉就等于整个进程死掉，所以多进程的程序要比多线程的程序健壮，但在进程切换时，耗费资源较大，效率要差一些。但对于一些要求同时进行并且又要共享某些变量的并发操作，只能用线程，不能用进程。

Linux 下常用软件，vim, emacs, tar, openoffice, putty, wget, links, ssh 等。

### 34. 用户在网站表单提交数据的时候，为了防止脚本攻击（比如用户输入<script>alert (111);

</script>），php 端接收数据的时候，应该如何处理？

可以对用户输入数据进行转义，如 htmlspecialchars(\$\_POST['title']);

### 35. 使用过 Memcache 缓存吗，如果使用过，能够简单的描述一下它的工作原理吗？

Memcache 是把所有的数据保存在内存当中，采用 hash 表的方式，每条数据由 key 和 value 组成，每个 key 是独一无二的，当要访问某个值的时候先按照找到值，然后返回结果。

Memcache 采用 LRU 算法来逐渐把过期数据清除掉。

### 36. 一个 Web 开发团队开发中，大致说说你所了解的所有成员的分工合作情况

每个公司的分工合作情况各不相同，一般会有策划，美工，前端开发，后台开发，维护，优化和推广等。

### 37. 假设给你 5 台服务器，请大致的描述一下，如何使用你所熟悉的开源软件，搭建一个日 PV 300 万左右的中型网站？参考结构：

3 台 Web 服务器，两台 MySQL 数据库服务器，采用 Master/Slave 同步的方式减轻数据库负载，Web 服务器可以结合 Memcache 缓存来减少负载，同时三台 Web 服务器内容一致，可以采用 DNS 轮询的方式来进行负载均衡。

### 38. 什么是 Ajax？Ajax 的原理是什么？Ajax 的核心技术是什么？Ajax 的优缺点是什么？

Ajax 是 Asynchronous JavaScript and XML 的缩写，是 JavaScript、XML、CSS、DOM 等多个技术的组合。

Ajax 的工作原理是一个页面的指定位置可以加载另一个页面所有的输出内容，这样就实现了一个静态页面也能获取到数据库中的返回数据信息了。所以 Ajax 技术实现了一个静态网页在不刷新整个页面的情况下与服务器通信，减少了用户等待时间，同时也从而降低了网络流量，增强了客户体验的友好程度。

Ajax 的核心技术是 XMLHttpRequest，它是 JavaScript 中的一个对象。

Ajax 的优点是：

(1).减轻了服务器端负担，将一部分以前由服务器负担的工作转移到客户端执行，利用客户端闲置的资源进行处理；

(2).在只局部刷新的情况下更新页面，增加了页面反应速度，使用户体验更友好。

Ajax 的缺点是不利于 SEO 推广优化，因为搜索引擎无法直接访问到 Ajax 请求的内容。

### 39. 请用 PHP 实现一个函数，将一个 2 进制数的无符号非负电位字符串非浮点字符串转成一个 10 进制数，返回该 10 进制数。不许使用 BIN 等系统内置函数（嘀嗒团）

题目意思有些模糊，题目本意可能是 将一个无符号的 2 进制字符串转成 10 进制数，如'10100010'，应该得到 10100010 的十进制表示 162。

```
1  <?php
2  function bin2dec($bin){
3      $temp = strrev($bin);
4      $result = 0;
5      for ($i=0,$len = strlen($temp); $i < $len ; $i++) {
6          $result += pow(2, $i) * $temp[$i];
7      }
8      return $result;
9  }
10
11  $a = '10100010';
12  echo bin2dec($a); //结果162
```

40. 请使用 PHP 设计一个函数，对学生英语考试得分从高到低排序，输入时所有学生的学号和考试得分，返回排好序的考试得分和对应学生的学号。考试满分为 100，得分可能会有小数，由于考试评分要求，小数位只会是 0 或 0.5 要求：

请不要使用 `qsort` 等系统内置排序函数

请使用你认为最快最优的方法实现该函数并使排序的性能最高。（嘀嗒团）

```

1 <?php
2 /* 快速排序实现 */
3 function array_sort(&$arr,$left,$right){
4     if ($left < $right){
5         $pivot = $arr[$left];
6         $low = $left;
7         $high = $right;
8         while ($low < $high) {
9             while ($low < $high && $arr[$high]['score'] >= $pivot['score']){
10                 $high--;
11             }
12             $arr[$low] = $arr[$high];
13             while ($low < $high && $arr[$low]['score'] <= $pivot['score']){
14                 $low++;
15             }
16             $arr[$high] = $arr[$low];
17         }
18         $arr[$low] = $pivot;
19         array_sort($arr,$left,$low - 1);
20         array_sort($arr,$low + 1,$right);
21     }
22 }
23 $english = array(
24     array('sid'=>1, 'score'=>76),
25     array('sid'=>2, 'score'=>93),
26     array('sid'=>3, 'score'=>68.5),
27     array('sid'=>4, 'score'=>82.5)
28 );
29 $left = 0;
30 $right = count($english) - 1 ;
31 array_sort($english,$left,$right);
32 print_r($english);

```

41. 需要设置一个有效期为 31 天，的 memcache 值，请补充下面的代码（奇矩互动）

```

<?php
    $memcache_obj=new memcache
    $memcache_obj->connect('memcache_host,11211');
    $memcache_obj->set('varKey','varValue',0,____);
?> time() + 3600 * 24 * 31

```

42. 现在请你设计一个留言板系统，请简要的写出你设计的其中分页算法的思路。（奇矩互动）

主要是数据库的设计系统的架构思想

分页算法的原理是 limit offset, pagesize

其中，pagesize 是设定好的，而 offset 则要通过计算得到，不同的页数对应的 offset 也不同，设当前页为 currentpage，则  $offset = (currentpage - 1) * pagesize$ 。

43. 假设有“123<em>abc</em>456<em>def</em>789”这么一个字符串，写一个函数，可以传入一个字符串，和一个要截取的长度。返回截取后的结果。（小米）要求：

(1). <em>和</em>标记不得计算在长度之内。

(2). 截取后的字符串，要保留原有<em>标签，不过如果最后有一个标签没有闭合，则去掉其开始标签。示例：题中的字符串，要截取长度 5，则返回的字符串应该为 123ab，要截取长度 8，应返回 123<em>abc</em>45。参考代码：

```
function cut($str, $len=null) {
    $last = 0;
    $str_len = strlen($str);
    $result = "";
    $result_len = 0;
    do {
        $pattern = '/<em>(.*?)</em>/i';
        $num = preg_match($pattern, $str, $m, PREG_OFFSET_CAPTURE, $last);
        if($num) {
            $result .= substr($str, $last,
                $add_len=($m[0][1]-$last<$len-$result_len)?$m[0][1]-$last:$len-$result_len);
            $result_len += $add_len; $last = $m[0][1] +
                strlen($m[0][0]); if($result_len < $len)
                { if($len-$result_len >= strlen($m[1][0])) {
                    $result .= $m[0][0];
                    $result_len += strlen($m[1][0]);
                } else {
                    $result .= substr($m[1][0], 0, $len-$result_len);
                    break;
                }
            }
        } else {
            $result .= substr($str, $last, $len-$result_len);
            break;
        }
    } while($last<$str_len && $result_len<$len);
    return $result;
}
```

44. 请仅使用一次正则替换，将下面内容

```
private long contract_id;
private string
```

```
contract_number;      private
string customer_name;
```

替换为

```
private long contractId;
private string contractNumber;
private string customerName; (鑫众人云)
```

```
1 <?php
2 $str = "private long contract_id;
3   private string contract_number;
4   private string customer_name; ";
5
6 $pattern = '/_(\w)/em';
7 $result = preg_replace($pattern,"strtoupper('\1')",$str);
8 echo $result;
```

#### 45. 列举流行的 Ajax 框架？说明 Ajax 实现原理是什么及 json 在 Ajax 中起什么作用？（鑫众人云）

流行的 Ajax 框架有 jQuery, Prototype, Dojo, MooTools。

Ajax 的工作原理是一个页面的指定位置可以加载另一个页面所有的输出内容，这样就实现了一个静态页面也能获取到数据库中的返回数据信息了。所以 Ajax 技术实现了一个静态网页在不刷新整个页面的情况下与服务器通信，减少了用户等待时间，同时也从而降低了网络流量，增强了客户体验的友好程度。

在使用 Ajax 时，涉及到数据传输，即将数据从服务器返回到客户端，服务器端和客户端分别使用不同的脚步语言来处理数据，这就需要一种通用的数据格式，XML 和 json 就是最常用的两种，而 json 比 XML 更简单。

#### 46. 在 UNIX 或 windows 系统内以（）为单位分配资源以（）单位分配时间调度（亿邮） 进程，时间片

#### 47. 正则表达式中\*? +的作用分别是什么（亿邮）

\* ? + 都有用来匹配数量的，\* 表示 0 或多个，? 表示 0 个或 1 个，+ 表示 1 个或多个。

#### 48. 写出你所知道的 XML 解析器（亿邮）

DOM, SAX, SimpleXML, 其中前两种是通用的解析器，和具体语言无关，而 SimpleXML 则是 PHP 提供的解析器。

#### 49. 在程序中表示时间可以使用哪几种变量类型（亿邮）

在 PHP 中可以使用 int 或字符串来表示（php 中没有日期时间类型），在 MySQL 中，可以使用 int, date, datetime, timestamp。

## 50. 使用 Utf-8 编码存储中文姓名，一般会分配多少个字节的存储空间（亿邮）

UTF-8 编码是可变长编码，对于中文而言，一个字符使用 3 个字节来存储。

## 51. 用正则表达式判断\$a 是否是一个以半角逗号分隔的多个手机号码组成的字符串，是输出 yes（卓望）

```
1 <?php
2 $pattern = '/^1[358]\d{9}(,1[358]\d{9})*$ /';
3 $subject = '13507224985,13833103237';
4 if (preg_match($pattern, $subject)){
5     echo 'yes';
6 }
```

## 52. 如果要求每隔 5 分钟执行一次脚本 five.php，如何实现？（卓望）

用到的函数 ignore\_user\_abort(),set\_time\_limit(),sleep(\$interval)，此代码只要运行一次后关闭浏览器即可。

```
<?php
ignore_user_abort(); //关掉浏览器，PHP 脚本也可以继续执行.
set_time_limit(0); // 通过 set_time_limit(0)可以让程序无限制的执行下去
$interval=60*5; // 每隔 5 分钟运行 do{
//这里是你要执行的代码

    sleep($interval); // 等待 5 分钟
}while(true);
```

## 53.谈一谈你对 PSR-4 规范的理解？

### 1. 概述

本 PSR 是关于由文件路径 自动载入 对应类的相关规范，本规范是可互操作的，可以作为任一自动载入规范的补充，其中包括 PSR-0，此外，本 PSR 还包括自动载入的类对应的文件存放路径规范。



## 关于「能愿动词」的使用

为了避免歧义，文档大量使用了「能愿动词」，对应的解释如下：

必须 (MUST)：绝对，严格遵循，请照做，无条件遵守；

一定不可 (MUST NOT)：禁令，严令禁止；

应该 (SHOULD)：强烈建议这样做，但是不强求；

不该 (SHOULD NOT)：强烈不建议这样做，但是不强求；

可以 (MAY) 和 可选 (OPTIONAL)：选择性高一点，在这个文档内，此词语使用较少；

参见：RFC 2119

## 2. 详细说明

此处的「类」泛指所有的「Class 类」、「接口」、「traits 可复用代码块」以及其它类似结构。

一个完整的类名需具有以下结构：

`\<命名空间>(\<子命名空间>)*\<类名>`

完整的类名 必须 要有一个顶级命名空间，被称为 “vendor namespace”；

完整的类名 可以 有一个或多个子命名空间；

完整的类名 必须 有一个最终的类名；

完整的类名中任意一部分中的下滑线都是没有特殊含义的；

完整的类名 可以 由任意大小写字母组成；

所有类名都 必须 是大小写敏感的。

当根据完整的类名载入相应的文件

完整的类名中，去掉最前面的命名空间分隔符，前面连续的一个或多个命名空间和子命名空间，作为「命名空间前缀」，其必须与至少一个「文件基目录」相对应；

紧接命名空间前缀后的子命名空间 必须 与相应的「文件基目录」相匹配，其中的命名空间分隔符将作为目录分隔符。

末尾的类名 必须 与对应的以 .php 为后缀的文件同名。

自动加载器 (autoloader) 的实现 一定不可 抛出异常、一定不可 触发任一级别的错误信息以及 不应该 有返回值。

## 54、谈谈你对依赖注入设计模式的理解？

什么是依赖注入。

简单的说：在 A 类中使用了 B 类的实例时，B 对象的构造不是在 A 类某个方法中初始化的，而是在 A 类外部初始化之后以 B 类的对象传入进来。这个过程就是依赖注入。

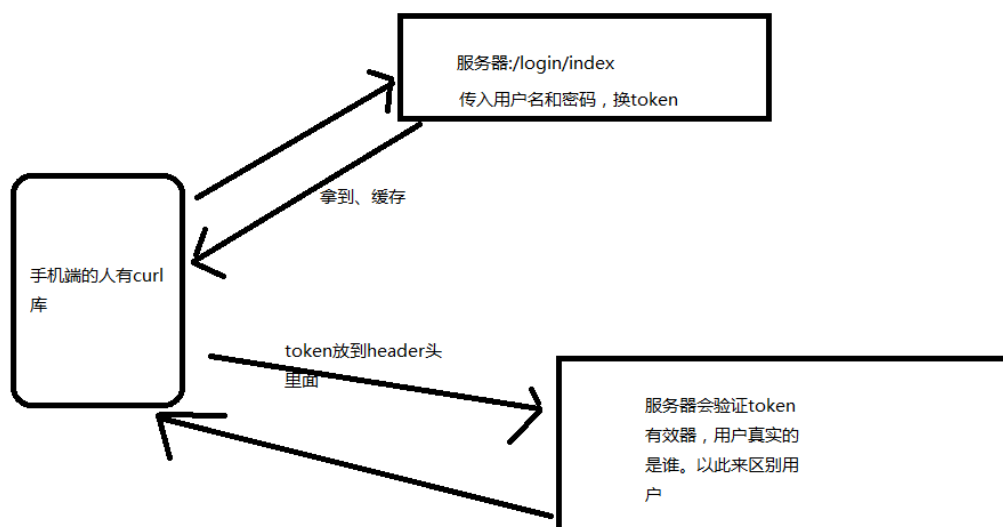
## 55、手机端不支持 cookie 和 session,但是你要实现用户登陆怎么办？

1. 最好在协议处使用 RestFul。
2. 换 token。

大概实现的原理如下：

手机端的人，如果拿到数据了，会将 token 缓存至它的手机缓存里面去。下次如果它希望能够取得你的数据。它会在它的 header 中，将 token 给带过来。

你就知道是哪个用户请求的了。你可以通过 token 换算出来是哪个用户。



## 56、什么是持续集成？什么是持续交付？什么是持续部署？

持续集成指的是频繁的（一天多次）将代码集成到主干。它的好处主要有两个：(1)、快速发现错误，每完成一点更新，就集成到主干，可以快速发现错误，定位错误也比较容易。

(2)、防止分支大幅偏离主干。如果不是经常集成，主干又在不断更新，会导致以后集成的难度变大，甚至难以集成。

持续集成的目的，就是让产品可以快速迭代，同时还能保持高质量。它的核心措施是，代码集成到主干之前，必须通过自动化测试。只要有一个测试用例失败，就不能集成。

持续交付（Continuous delivery）指的是，频繁地将软件的新版本，交付给质量团队或者用户，以供评审。如果评审通过，代码就进入生产阶段。

持续交付可以看作持续集成的下一步。它强调的是，不管怎么更新，软件是随时随地可以交付的。

持续部署（continuous deployment）是持续交付的下一步，指的是代码通过评审以后，自动部署到生产环境。

持续部署的目标是，代码在任何时刻都是可部署的，可以进入生产阶段。  
持续部署的前提是能自动化完成测试、构建、部署等步骤。

## 57、如何理解单元测试？

单元测试是测试的一个子类，并非写了测试就叫单元测试，甚至你用了单元测试框架也有可能写出越过单元测试边界的代码。正确的单元测试就是确保测试代码准确隔离（isolate）了待测代码，如果你测试一个类，那么测试代码中就应该避免出现对于其他类的依赖（语言的标准库或者框架提供的工具方法 / 助手方法例外），甚至你测试该类的某个方法都要尽量避免对类内部其他成员的依赖。

## 58、如何理解 php 中的各种设计模式？

1、单例模式：PHP 中的对象生存期间是从该脚本开始一直到该脚本结束为止，因此 PHP 的单例模式只是在一个页面中（这里可能包含很多其他页面，不是狭义的单页面）多次用到该对象时才会起作用，多次用到时不去重复的 new 对象(多个人做一个项目时，难免会碰到一次请求中多次实例一个对象的情况)，将不会耗费不必要的资源(数据控连接操做效果很明显)，还有一点就是可以保证整个脚本中都是同一个对象，这种模式是怎么实现的呢，他的实现有几个要注意的点：

- a. 首先就是要将\_\_construct()方法定义为私有方法，这样就不能通过 new 来得到一个新的实例了，单例模式不能在外进行实例化，这能字自身内部进行实例化；
- b. 同样要屏蔽\_\_clone()方法，防止从类外部进行克隆
- c. 然后就是定义一个用来保存实例的私有变量和获取私有变量的公有函数 getInstance()。

2、简单工厂设计模式：是由一个工厂对象决定创建出哪一种产品类的实例。简单工厂模式是工厂模式家族中最简单实用的模式，可以理解为是不同工厂模式的一个特殊实现。

简单工厂模式在项目中有何好处呢？他有两个好处：

- a. 首先就是使用简单工厂模式可以根据不同的参数去实例化不同的类，而不是 new 方法去实例化每一个不同的类，这样给使用者这样更好管理。
- b. 其次就是如果要实例化的类在多个文件中用到，当我们修改类名称的时候只需要去修改工厂类就行，而不必修改每一个实例化该类的文件。

3、观察者模式：实例化时先把新对象 add 成员属性的数组，然后执行一个触发器方法，用来遍历刚刚那个数组，当一个对象发生改变时，依赖他的对象全部会收到通知，并自动进行相应操作。

缺点：一个事件发生后，要执行一连串更新操作。传统的编程方式：时间代码后边直接加入处理逻辑，当更新的逻辑增多后，代码变得难以维护。耦合度高，增加新逻辑需要修改事件主题代码。

4、代理模式：代理模式给某一个对象提供一个代理对象，并由代理对象控制对源对象的引用。代理就是一个人或一个机构代表另一个人或者一个机构采取行动。某些情况下，客户不想或者不能够直接引用一个对象，代理对象可以在客户和目标对象直接起到中介的作用。客户端分辨不出代理主题对象与真实主题对象。代理模式可以并不知道真正的被代理对象，而仅仅持有有一个被代理对象的接口，这时候代理对象不能够创建被代理对象，被代理对象必须有系统的其他角色代为创建并传入。

5、策略模式：策略模式针对一组算法，将每一个算法封装到具有共同接口的独立的类中，从而使它们可以相互替换。策略模式使得算法可以在不影响到客户端的情况下发生变化，策略模式把行为和环境分开。环境类负责维持和查询行为类，各种算法在具体的策略类中提供。由于算法和环境独立开来，算法的增减，修改都不会影响到环境和客户端。

## 6、反射

它是指在 PHP 运行状态中，扩展分析 PHP 程序，导出或提取出关于类、方法、属性、参数等的详细信息，包括注释。这种动态获取的信息以及动态调用对象的方法的功能称为反射 API。反射是操纵面向对象范型中元模型的 API，其功能十分强大，可帮助我们构建复杂，可扩展的应用。

其用途如：自动加载插件，自动生成文档，甚至可用来扩充 PHP 语言。

php 反射 api 由若干类组成，可帮助我们用来访问程序的元数据或者同相关的注释交互。借助反射我们可以获取诸如类实现了那些方法，创建一个类的实例（不同于用 new 创建），调用一个方法（也不同于常规调用），传递参数，动态调用类的静态方法。

## 7、

Ioc 控制反转

DI 依赖注入

依赖注入和控制反转说的实际上是同一个东西，它们是一种设计模式，这种设计模式用来减少程序间的耦合。使用依赖注入，最重要的一点好处就是有效的分离了对象和它所需要的外部资源，使得它们松散耦合，有利于功能复用，更重要的是使得程序的整个体系结构变得非常灵活。

依赖注入和控制反转是对同一件事情的不同描述，从某个方面讲，就是它们描述的角度不同。

- 依赖注入是从应用程序的角度在描述，可以把依赖注入，即：应用程序依赖容器创建并注入它所需要的外部资源；
- 
- 而控制反转是从容器的角度在描述，即：容器控制应用程序，由容器反向的向应用程序注入应用程序所需要的外部资源。

**门面设计模式：**（有时候也称外观模式）是指提供一个**统一的接口**去访问多个子系统的多个不同的接口，它为子系统的一组接口提供一个统一的高层接口。使用子系统更容易使用。

案例：炒股票，新股民不了解证券知识做股票，是很容易亏钱的，需要学习的知识太多了，这样新手最好把炒股的事情委托给基金公司，基金公司了解证券知识，那么新股民把自己的股票托管给基金公司去运营，这样新股民不必了解哪只股票的走势就可以完成股票的买卖。基金公司在这里就是一个门面，针对于新股民的门面。

角色分析：

**门面（FacadeCompany）角色：**此角色封装一个高层接口，将客户端的请求代理给适当的子系统对象，是门面模式的核心接口。

**子系统（ICBC）角色：**实现子系统的的功能，处理 FacadeCompany 对象指派的任务。子系统没有 FacadeCompany 的任何信息，没有对 FacadeCompany 对象的引用。

优势:

1) 它可以**对客户屏蔽子系统组件**，因而减少了客户处理的对象的数目并使得子系统使用起来更加方便。

2) 它实现了子系统与客户之间的松耦合关系，而子系统内部的功能组件往往是紧耦合的。松耦合关系**使得子系统的组件变化不会影响到它的客户**。 Facade 模式有助于建立层次结构系统，也有助于对对象之间的依赖关系分层。

3) Facade 模式可以**消除复杂的循环依赖关系**。这一点在客户程序与子系统是分别实现的时候尤为重要。在大型软件系统中降低编译依赖性至关重要。在子系统类改变时，希望尽量减少重编译工作以节省时间。用 Facade 可以降低编译依赖性，限制重要系统中较小的变化所需的重编译工作。

4) Facade 模式同样也有利于简化系统在不同平台之间的移植过程，因为编译一个子系统一般不需要编译所有其他的子系统。

适配器设计模式：将一个类的换成客户端需要的另一个接口，使原本不兼容的接口能一起工作。一个现实中的例子，假如屋里墙上只有两项插口，而风扇有三个插头，这时需要再拿一个移动插线板，插线板接入墙壁插口的一段是两项插头，插线板另外一段能够提供三项插口给风扇，这时候插线板起到了一个适配器的作用。

在应用中，适配器模式分为类适配器和对象适配器。

## 59、git 版本控制器常用的命令？

Git clone 地址	就是将远程的分支给下载下来,初使化
Git add	添加文件
Git commit -m '内容'	添加说明
Git init	git 仓库初使化
Git push	提交
Git pull	拉取
Git diff	检查对比文件的一致性
Git status	显示相关的当前的 git 信息
Git status -s	哪些文件发生了变化
Git branch 分支	创建分支
Git checkout 分支名	切换分支
Git branch -d 分支名	删除分支
Git merge 源分支名	从源分支名合并内容至当前分支
Git log	查看 git 历史
Git rm	删除 git 中的文件

Git reset --hard 写上 log 中的编号	回滚至某个版本
Git checkout -b A B 分支	建立 A 分支，从 B 分支取内容
git branch	显示所有分支

## 60、请问你使用过 xdebug 进行过 php 调试吗，请简单介绍 xdebug

什么是 Xdebug?

Xdebug 是一个开放源代码的 PHP 程序调试器(即一个 Debug 工具)，可以用来跟踪，调试和分析 PHP 程序的运行状况。Xdebug 现在的最新版本是 [xdebug 2.0.0beta6](http://xdebug.org)，支持 PHP4/PHP5。

官方站点：[www.Xdebug.org](http://www.Xdebug.org)

安装教程：

1. 首先下载相应的 php\_xdebug.dll, 下载地址：

<http://www.xdebug.org/download.php>

2. 将下载文件复制到你的 php 下的 ext 目录里。

3. 编辑 php.ini 文件，加入下面信息：

```
[Xdebug]
```

```
zend_extension=D:\xampp\php\ext\php_xdebug.dll (Win)
```

```
xdebug.profiler_enable=on
```

```
xdebug.trace_output_dir="../../Projects/xdebug"
```

```
xdebug.profiler_output_dir="../../Projects/xdebug"
```

后面的目录“../../Projects/xdebug”为你想要放置 Xdebug 输出的数据文件的目录，可自由设置。

参数参考设置：

xdebug.default\_enable=on; 显示默认的错误信息

xdebug.auto\_trace=on; 自动打开“监测函数调用过程”的功模。该功能可以在你指定的目录中将函数调用的监测信息以文件的形式输出。此配置项的默认值为 off。

xdebug.collect\_params=on; 打开收集“函数参数”的功能。将函数调用的参数值列入函数过程调用的监测信息中。此配置项的默认值为 off。

xdebug.collect\_return=on; 打开收集“函数返回值”的功能。将函数的返回值列入函数过程调用的监测信息中。此配置项的默认值为 off。

xdebug.max\_nesting\_level=100

xdebug.profiler\_enable=on; 打开效能监测器。

xdebug.remote\_enable=on; 是否调试

xdebug.remote\_host=localhost

xdebug.remote\_port=9000; 调试端口

xdebug.remote\_handler=dbgp; 选择协议

xdebug.trace\_output\_dir="d:\Temp"; 设定函数调用监测信息的输出文件的路径。

xdebug.profiler\_output\_dir="d:\Temp"; 设定效能监测信息输出文件的路径。

至此，安装完成，可以通过<?php phpinfo();?>查看：

## xdebug

xdebug support	enabled
Version	2.1.1

Supported protocols	Revision
DBGp - Common DeBuGger Protocol	\$Revision: 1.145 \$

Directive	Local Value	Master Value
xdebug.auto_trace	Off	Off
xdebug.collect_assignments	Off	Off

可以看看调试效果了，随便打一段错误的 php 代码：

通过访问，显示错误信息如下：

```
Warning: require_once(test.php) [function.require-once]: failed to open stream: No such file or directory in I:\Projects\xdebug\index.php on line 3
```

#### Call Stack

#	Function	Location
1	{main} ()	I:\Projects\xdebug\index.php:0

```
Fatal error: require_once() [function.require]: Failed opening required 'test.php' (include_path=.;c:\php\includes;C:\php5\ext;C:\php5\PEAR;) in I:\Projects\xdebug\index.php on line 3
```

#### Call Stack

#	Function	Location
1	{main} ()	I:\Projects\xdebug\index.php:0

## 61、请简述一种 mysql 压力测试工具 ?

mysqlslap 是一个 mysql 官方提供的压力测试工具

mysqlslapMySQL 从 5.1.4 版开始带有一个压力测试工具 mysqlslap，通过模拟多个并发客户端访问 mysql 来执行测试，使用起来非常的简单。通过 mysqlslap -help 可以获得可用的选项，

--auto-generate-sql, -a

自动生成测试表和数据

--auto-generate-sql-load-type=type

测试语句的类型。取值包括：read, key, write, update 和 mixed(默认)。

--number-char-cols=N, -x N

自动生成的测试表中包含多少个字符类型的列，默认 1

--number-int-cols=N, -y N

自动生成的测试表中包含多少个数字类型的列，默认 1

--number-of-queries=N

总的测试查询次数(并发客户数×每客户查询次数)

--query=name,-q

使用自定义脚本执行测试，例如可以调用自定义的一个存储过程或者 sql 语句来执行测试。

--create-schema

测试的 schema，MySQL 中 schema 也就是 database

--commint=N

多少条 DML 后提交一次

--compress, -C

如果服务器和客户端支持都压缩，则压缩信息传递

--concurrency=N, -c N

并发量，也就是模拟多少个客户端同时执行 select。可指定多个值，以逗号或者 delimiter 参数指定的

值做为分隔符

--engine=engine\_name, -e engine\_name

创建测试表所使用的存储引擎，可指定多个



--iterations=N, -i N  
测试执行的迭代次数  
--detach=N  
执行 N 条语句后断开重连  
--debug-info, -T  
打印内存和 CPU 的信息  
--only-print  
只打印测试语句而不实际执行

## 62、Set\_exception\_handler 和 set\_error\_handler 以及 trigger\_error 的用法区别

Set\_exception\_handler 和 set\_error\_handler 函数都是用于 PHP 中的异常处理，但两者所应用的范围不一样，简单点说，set\_exception\_handler 是通用的异常处理程序，应用面比较广，而 set\_error\_handler 针对性更强一些。如果你可以预见某个地方会出现什么类型的错误，可以使用 set\_error\_handler，如果不能预见会出现什么异常，那就可以使用 set\_exception\_handler。

1. set\_exception\_handler :

set\_exception\_handler() **函数**设置用户自定义的异常处理**函数**。

该**函数**用于创建运行时期的用户自己的异常处理方法。

该**函数**会返回旧的异常处理程序，若失败，则返回 null。

2.set\_error\_handler

set\_error\_handler() **函数**设置用户自定义的错误处理**函数**。

该**函数**用于创建运行时期的用户自己的错误处理方法。

该**函数**会返回旧的错误处理程序，若失败，则返回 null。

3.trigger\_error

trigger\_error() **函数**创建用户定义的错误消息。

trigger\_error() 用于在用户指定的条件下触发一个错误消息。它与内建的错误处理器一同使用，也可以与由 set\_error\_handler() **函数**创建的用户自定义**函数**使用。

如果指定了一个不合法的错误类型，该**函数**返回 false，否则返回 true。

