

php面试题汇总

天天



目 录

题库一
题库二
题库三
题库四
题库五
题库六
算法题

题库一

1.表单中 get与post提交方法的区别?

答: get是发送请求HTTP协议通过url参数传递进行接收, 而post是实体数据, 可以通过表单提交大量信息.

2.session与cookie的区别?

答: session: 储存用户访问的全局唯一变量, 存储在服务器上的php指定的目录中的 (session_dir) 的位置进行的存放
cookie: 用来存储连续访问一个页面时所使用, 是存储在客户端, 对于Cookie来说是存储在用户WIN的Temp目录中的。
两者都可通过时间来设置时间长短

3.数据库中的事务是什么?

答: 事务 (transaction) 是作为一个单元的一组有序的数据库操作。如果组中的所有操作都成功, 则认为事务成功, 即使只有一个操作失败, 事务也不成功。如果所有操作完成,

事务则提交, 其修改将作用于所有其他数据库进程。如果一个操作失败, 则事务将回滚, 该事务所有操作的影响都将取消。

简述题:

1、用PHP打印出前一天的时间格式是2006-5-10 22:21:21(2分)

答: `echo date('Y-m-d H:i:s', strtotime('-1 days'));`

2、echo(),print(),print_r()的区别(3分)

答: echo和print是PHP语言结构, print_r是函数, 语言结构没有返回值, 函数可以有返回值 (即便没有用)
print () 只能打印出简单类型变量的值 (如int, string)
print_r () 可以打印出复杂类型变量的值 (如数组, 对象)
echo 输出一个或者多个字符串

3、能够使HTML和PHP分离开使用的模板(1分)

答:Smarty,Dwoo,TinyButStrong,Template Lite,Savant,phemplate,XTemplate

4、使用哪些工具进行版本控制?(1分)

答:cvcs,svn,vss;

5、如何实现字符串翻转?(3分)

答:echo strrev(\$a);

6、优化MYSQL数据库的方法。(4分，多写多得)

答:

- 1、选取最适用的字段属性,尽可能减少定义字段长度,尽量把字段设置NOT NULL,例如'省份,性别',最好设置为ENUM
- 2、使用连接(JOIN)来代替子查询:
 - a.删除没有任何订单客户:DELETE FROM customerinfo WHERE customerid NOT in(SELECT customerid FROM orderinfo)
 - b.提取所有没有订单客户:SELECT FROM customerinfo WHERE customerid NOT in(SELECT customerid FROM orderinfo)
 - c.提高b的速度优化:SELECT FROM customerinfo LEFT JOIN orderid customerinfo.customerid=orderid.customerid WHERE orderinfo.customerid IS NULL
- 3、使用联合(UNION)来代替手动创建的临时表
 - a.创建临时表:SELECT name FROM `nametest` UNION SELECT username FROM `nametest2`
- 4、事务处理:
 - a.保证数据完整性,例如添加和修改同时,两者成立则都执行,一者失败都失败


```
mysql_query("BEGIN");
mysql_query('INSERT INTO customerinfo (name) VALUES ('. $name1. ')');
mysql_query("SELECT * FROM `orderinfo` where customerid=". $id);
mysql_query("COMMIT");
```
- 5、锁定表,优化事务处理:
 - a.我们用一个 SELECT 语句取出初始数据,通过一些计算,用 UPDATE 语句将新值更新到表中。

包含有 WRITE 关键字的 LOCK TABLE 语句可以保证在 UNLOCK TABLES 命令被执行之前,不会有其它的访问来对 inventory 进行插入、更新或者删除的操作

```
mysql_query("LOCK TABLE customerinfo READ, orderinfo WRITE");
mysql_query("SELECT customerid FROM `customerinfo` where id=". $id);
mysql_query("UPDATE `orderinfo` SET ordertitle='$title' where customerid=". $id);
mysql_query("UNLOCK TABLES");
```
- 6、使用外键,优化锁定表
 - a.把customerinfo里的customerid映射到orderinfo里的customerid,任何一条没有合法的customerid的记录不会写到orderinfo里

```
CREATE TABLE customerinfo
(
    customerid INT NOT NULL,
    PRIMARY KEY(customerid)
)TYPE = INNODB;
CREATE TABLE orderinfo
(
    orderid INT NOT NULL,
    customerid INT NOT NULL,
    PRIMARY KEY(customerid,orderid),
    FOREIGN KEY (customerid) REFERENCES customerinfo
    (customerid) ON DELETE CASCADE
)TYPE = INNODB;
```

注意:'ON DELETE CASCADE',该参数保证当customerinfo表中的一条记录删除的话同时也会删除order

表中的该用户的所有记录,注意使用外键要定义事务安全类型为INNODB;

7、建立索引:

a.格式:

(普通索引)->

创建:CREATE INDEX <索引名> ON tablename (索引字段)

修改:ALTER TABLE tablename ADD INDEX [索引名] (索引字段)

创表指定索引:CREATE TABLE tablename(...,INDEX[索引名](索引字段))

(唯一索引)->

创建:CREATE UNIQUE <索引名> ON tablename (索引字段)

修改:ALTER TABLE tablename ADD UNIQUE [索引名] (索引字段)

创表指定索引:CREATE TABLE tablename(...,UNIQUE[索引名](索引字段))

)

(主键)->

它是唯一索引,一般在创建表是建立,格式为:

CREATA TABLE tablename (... ,PRIMARY KEY[索引字段])

8、优化查询语句

a.最好在相同字段进行比较操作,在建立好的索引字段上尽量减少函数操作

例子1:

SELECT * FROM order WHERE YEAR(orderDate)<2008;(慢)

SELECT * FROM order WHERE orderDate<"2008-01-01";(快)

例子2:

SELECT * FROM order WHERE addtime/7<24;(慢)

SELECT * FROM order WHERE addtime<24*7;(快)

例子3:

SELECT * FROM order WHERE title like "%good%";

SELECT * FROM order WHERE title>="good" and name<"good";

8、PHP的意思(送1分)

答:PHP是一个基于服务端来创建动态网站的脚本语言,您可以用PHP和HTML生成网站主页

9、MYSQL取得当前时间的函数是?,格式化日期的函数是(2分)

答:now(),date()

10、实现中文字串截取无乱码的方法。(3分)

答:

```
function GBsubstr($string, $start, $length) {  
    if(strlen($string)>$length){  
        $str=null;  
        $len=$start+$length;  
        for($i=$start;$i<$len;$i++){  
            if(ord(substr($string,$i,1))>0xa0){  
                $str.=substr($string,$i,2);  
                $i++;  
            }else{  
                $str.=substr($string,$i,1);  
            }  
        }  
        return $str.'...';  
    }else{  
        return $string;  
    }  
}
```

12、您是否用过模板引擎? 如果有您用的模板引擎的名字是?(1分)

答:用过, smarty

13、请简单阐述您最得意的开发之作(4分)

答:信息分类

14、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?(4分)

答:确认服务器硬件是否足够支持当前的流量,数据库读写分离,优化数据表,程序功能规则,禁止外部的盗链,控制大文件的下载,使用不同主机分流主要流量

15、用PHP写出显示客户端IP与服务器IP的代码(1分)

答:打印客户端IP:echo \$_SERVER['REMOTE_ADDR']; 或者: getenv('REMOTE_ADDR');
打印服务器IP:echo gethostbyname("www.bolaiwu.com")

16、语句include和require的区别是什么?为避免多次包含同一文件,可用(?)语句代替它们?(2分)

答:require->require是无条件包含也就是如果一个流程里加入require,无论条件成立与否都会先执行require
include->include有返回值,而require没有(可能因为如此require的速度比include快)
注意:包含文件不存在或者语法错误的时候require是致命的,include不是

17、如何修改SESSION的生存时间(1分).

答:方法1:将php.ini中的session.gc_maxlifetime设置为9999重启apache
方法2:\$savePath = "./session_save_dir/";
\$lifeTime = 小时 * 秒;
session_save_path(\$savePath);
session_set_cookie_params(\$lifeTime);
session_start();
方法3:setcookie() and session_set_cookie_params(\$lifeTime);

题库二

1.以下哪一句不会把 John 新增到 users 阵列？

```
$users[] = 'john';  
成功把 John 新增到阵列 users。  
array_add($users, 'john');  
函式 array_add() 无定义。  
array_push($users, 'john');  
成功把 John 新增到阵列 users。  
$users ||= 'john';  
语法错误。
```

2.sort()、assort()、和 ksort() 有什么分别？它们分别在什么情况下使用？

```
sort()  
根据阵列中元素的值，以英文字母顺序排序，索引键会由 0 到 n-1 重新编号。主要是当阵列索引键的值无关疼痒时用来把阵列排序。  
assort()  
PHP 没有 assort() 函式，所以可能是 asort() 的笔误。  
asort()  
与 sort() 一样把阵列的元素按英文字母顺序来排列，不同的是所有索引键都获得保留，特别适合替联想阵列排序。  
ksort()  
根据阵列中索引键的值，以英文字母顺序排序，特别适合用于希望把索引键排序的联想阵列。
```

3.以下的代码会产生什么？为什么？

```
$num =10;  
function multiply(){  
    $num =$num *10;  
}  
multiply();  
echo $num;  
由于函式 multiply() 没有指定 $num 为全域变量（例如 global $num 或者 $_GLOBALS['num']），所以 $num 的值是 10。
```

1. reference 跟一个正规的变量有什么分别？如何 pass by reference？在什么情况下我们需要这样做？

Reference 传送的是变量的地址而非它的值，所以在函式中改变一个变量的值时，整个应用都见到这个变量的新值。
一个正规变量传送给函式的是它的值，当函式改变这个变量的值时，只有这个函式才见到新值，应用的其他部分仍然见到旧值。


```
$myVariable = "its' value";
```

Myfunction(&\$myVariable); // 以 reference 传送参数以 reference 传送参数给函式，可以使函式改变了的变量，即使在函式结束后仍然保留新值。

5. 些函式可以用来在现正执行的脚本中插入函式库？

对这道题目不同的理解会有不同的答案，我的第一个想法是插入 PHP 函式库不外乎 include()、include_once()、require()、require_once()，

但细心再想，“函式库”也应该包括 com 物件和 .net 函式库，所以我们的答案也要分别包括 com_load 和 dotnet_load，下次有人提起“函式库”的时候，别忘记这两个函式。

6. foo() 与 @foo() 有什么分别？

foo() 会执行这个函式，任何解译错误、语法错误、执行错误都会在页面上显示出来。

@foo() 在执行这个函式时，会隐藏所有上述的错误讯息。

很多应用程式都使用 @mysql_connect() 和 @mysql_query 来隐藏 mysql 的错误讯息，我认为这是很严重的失误，因为错误不该被隐藏，你必须妥善处理它们，可能的话解决它们。

7. 你如何替 PHP 的应用程式侦错？

我并不常这样做，我曾经试过很多不同的侦错工具，在 Linux 系统中设定这些工具一点也不容易。不过以下我会介绍一个近来颇受瞩目的侦错工具。

PHP - Advanced PHP Debugger 或称 PHP - APD，第一步是执行以下的指令安装：

pear install apd 安装后在你的脚本的开头位置加入以下的语句开始进行侦错：

apd_set_pprof_trace(); 执行完毕，打开以下档案来查阅执行日志：

apd.dumpdir

你也可以使用 pprof 来格式化日志。

详细的资料可以参阅 <http://us.php.net/manual/en/ref.apd.php>。

8. “===”是什么？试举一个“==”是真但“===”是假的例子。

“===”是既可以送回布尔值“假”，也可以送回一个不是布尔值但却可以赋与“假”值的函式，strpos() 和 strrpos() 便是其中两个例子。

问题的第二部份有点困难，想一个 “==” 是假，但是 “===” 是真的例子却很容易，相反的例子却很少。但我终于找到以下的例子：

```
if (strpos("abc", "a") == true){
```

```
// 这部分永不会被执行，因为 "a" 的位置是 0，换算成布尔值“假”
```

```
}
```

```
if (strpos("abc", "a") === true){
```

```
// 这部份会被执行，因为“===”保证函式 strpos() 的送回值不会换算成布尔值。
```

```
}
```

9. 你会如何定义一个没有成员函式或特性的类别 myclass？

```
`class myclass{}`
```

10. 你如何产生一个 myclass 的物件？

```
`$obj = new myclass();`
```

11. 在一个类别内如何存取这个类别的特性及变改它的值？

使用语句：\$this->propertyName，例如：

```
class myclass{
```

```
    private $propertyName;
    public function __construct() {
        $this->propertyName = "value";
    }
}
```

12. include 和 include_once 有什么分别？require 又如何？

三者都是用来在脚本中插入其他档案，视乎 url_allow_fopen 是否核准，这个档案可以从系统内部或外部取得。但他们之间也有微细的分别：

include()：这个函式容许你在脚本中把同一个档案插入多次，若果档案不存在，它会发出系统警告并继续执行脚本。

include_once()：它跟 include() 的功能相似，正如它的名字所示，在脚本的执行期间，有关档案只会被插入一次。

require()：跟 include() 差不多，它也是用来在脚本中插入其他档案，但若果档案不存在，本文档使用 [看云](#) 构建

它会发出系统警告，这个警告会引致致命错误令脚本中止执行

13. 以下哪一个函式可以把浏览器转向到另一个页面？

`redir()`

这不是一个 PHP 函式，会引致执行错误。

`header()`

这个是正确答案，`header()` 用来插入卷头资料，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 函式，会引致执行错误。

`redirect()`

这不是一个 PHP 函式，会引致执行错误。

14. 以下哪一个函式可以用来开启档案以便读／写？

`fget()`

这不是一个 PHP 函式，会引致执行错误。

`file_open()`

这不是一个 PHP 函式，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，事实上这个函式还有很多选项，详细资料请参阅 php.net。

`open_file()`

这不是一个 PHP 函式，会引致执行错误。

15. `mysql_fetch_row()` 和 `mysql_fetch_array()` 有什么分别？

`mysql_fetch_row()` 把数据库的一列储存在一个以零为基数的阵列中，第一栏在阵列的索引 0，第二栏在索引 1，如此类推。

`mysql_fetch_assoc()` 把数据库的一列储存在一个关联阵列中，阵列的索引就是栏位名称，例如我的数据库查询送回 “first_name”、“last_name”、“email” 三个栏位，阵列的索

引便是 “first_name” 、 “last_name” 和 “email” 。

mysql_fetch_array() 可以同时送回 mysql_fetch_row() 和 mysql_fetch_assoc() 的值。

16. 下面的代码用来做什么？请解释。

```
$date='08/26/2003';print ereg_replace("([0-9]+)/([0-9]+)/([0-9]+)","  
[url=file://2///1///3%22,$date]\2\1\3",$date[url]);
```

答：

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

我的一个好朋友告诉我可以把这个正规表达式拆解为以下的语句，对于如此简单的表示是来说其实无须拆解，纯粹为了解说的方便：

// 对应一个或更多 0-9，后面紧随一个斜号 \$regExpression = "([0-9]+)/";

// 应一个或更多 0-9，后面紧随另一个斜号 \$regExpression .= "([0-9]+)/";

// 再次对应一个或更多 0-9 \$regExpression .= "([0-9]+)";至于

[url=file://2///1///3]\2\1\3[url] 则是用来对应括号，第一个括号对的是月份，第二个括号对应的是日期，第三个括号对应的是年份。

17. 给你一行文字 \$string，你会如何编写一个正规表达式，把 \$string 内的 HTML 标签除去？

首先，PHP 有内建函式 strip_tags() 除去 HTML 标签，为何要自行编写正规表达式？好了，便当作是面试的一道考题吧，我会这样回答：

```
$stringOfText = "
```

```
This is a test
```

```
";
```

```
$expression = "/<(.*?)>(.*?)<\V(.*?)>/";
```

```
echo preg_replace($expression, "[url=file://2/]2[url]", $stringOfText);
```

```
// 有人说也可以使用 /(^[^>]+>)/
```

```
$expression = "/(<[^\>]*>)/";
```

```
echo preg_replace($expression, "", $stringOfText);
```

18. PHP 和 Perl 分辨阵列和散列表的方法有什么差异？

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 所有阵列变量都是以 @ 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。

19. 你如何利用 PHP 解决 HTTP 的无状态本质？

最主要的俩各选择是 session 和 cookie。使用 session 的方法是在每一页的开始加上 session_start()，然后利用 \$_SESSION 散列表来储存 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() 函式，此外只需使用 \$_COOKIE 散列表便可以存取所有 cookie 变量。

还有一个不那么可靠的方法，就是利用访客的 IP 地址，这个方法有特定的危险性。

20. GD 函式库用来做什么？

这个可能是我最喜欢的函式库，自从 PHP 4.3.0 版本后 GD 便内建在 PHP 系统中。

这个函式库让你处理和显示各式格式的图档，它的另一个常见用途是制作所图档。

GD 以外的另一个选择是 ImageMagick，但这个函式库并不内建于 PHP 之中，必须由系统管理员安装在伺服器上。

21. 试写出几个输出一段 HTML 代码的方法。

嗯，你可以使用 PHP 中任何一种输出语句，包括 echo、print、printf，大部分人都使用如下例的 echo：

echo "My string \$variable";你也可以使用这种方法：

echo << HTML tags in here as well. The END;

remarks must be on a line of its own, and can't contain any extra white space.END;

22. PHP 比 Perl 好吗？请讨论。

我们不要为一个简单的问题引发一场舌战，正如我经常说的：“为工作选择适合的语言，不要把工作迁就语言。”

我个人认为 Perl 十分适合用作命令行工具，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，但是个人认为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式，Perl 则似乎以命令行为设计之本。

个人来说两种语言我都喜欢，在大学期间我经常使用 Perl、PHP 和 Java，可惜工作上我使用 C#，

但在我家里我花不少时间操练 PHP、Perl、Ruby（现正学习）和 Java，保持我的技能知识在最新状态。

很多人问我 C 和 C++ 怎么样，它们是否仍有机会在我的应用中占一席位，我的答案基本上是“否”，

我近来的工作主要集中在网页开发，虽然 C 和 C++ 也可以用来写网页，但它们到底不是为这种工作而设计的，“为工作选择适合的语言”，

若果我需要编写一个控制台应用，用来展示 bubble sort、quick sort 和 merge sort 的效能比较，我一定会使用 C / C++。

若果我需要编写一个相片簿系统，我会使用 PHP 或者 C#（我认为制作用户介面方面 .NET 语言比网页更加）。

题库三

一、简答题

1.用PHP打印出今天的时间，格式是2009-9-2 22:21:21 2009-09-02 22:21:21

```
date("Y-m-d H:i:s",time());  date("Y-n-j H:i:s");
```

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

```
$str = file_get_contents('http://www.xxx.com/index.html');
```

3.写出发贴数最多的十个人名字的SQL，利用下表：

members(id,username,posts,pass,email)

```
$sql = "select username from members order by posts desc limit 0,10";
```

4.在PHP中error_reporting这个函数有什么作用？

```
打开或者关闭错误报告 //error_reporting(0); error_reporting(E_ALL ^ E_NOTICE)  
; error_reporting(E_ALL);
```

5.func()和@func()之间有什么区别？

第二个函数调用失败不会报错，第一个会报错

6.mysql_fetch_row() 和mysql_fetch_assoc()和mysql_fetch_array之间有什么区别？

第一个是返回结果集中的一行作为索引数组，第二个是返回关联数组，而第三个既可以返回索引数组也可以返回关联数组，取决于它的第二个参数 MYSQL_BOTH MYSQL_NUM MYSQL_ASSOC 默认为MYSQL_BOTH

```
$sql ="select * from table1";  
$result = mysql_query($sql);  
mysql_fetch_array($result, MYSQL_NUM);
```

7.写出以下程序的输出结果

```
<?
$b=201;
$c=40;
$a= $b>$c?4:5;
echo $a;
?>
```

答案：4

8.取得查询结果集总数的函数是？

```
mysql_num_rows($result)
```

9.取得增删改等操作所影响的行数的函数是什么？

```
mysql_affected_rows($result) 不写参数则返回最近一次操作的结果集 失败返回-1
```

10.取得最新一次添加记录（假设id为主键，并且是自增类型）所产生的id的函数是什么？

```
mysql_insert_id();如果上一查询没有产生 AUTO_INCREMENT 的值，则 mysql_insert_id() 返回 0
```

11.\$arr = array('james', 'tom', 'symfony'); 请将 'jack' 添加到\$arr数组的开头，并把jack打印出来

```
array_unshift($arr, 'jack'); echo $arr[0];
```

12.\$arr = array('james', 'tom', 'symfony');请将\$arr数组的值用 ' ' 分割并合并成字符串输出？

```
echo implode(' ', $arr);
```

13.\$str = 'jack,james,tom,symfony' ; 请将\$str用 ' ' 分割，并把分割后的值放到\$arr数组中？


```
$arr = explode(',', $str);
```

14.\$arr = array(3,7,2,1,' d' , ' abc'); 请将\$arr按照从大到小的顺序排序, 并保持其键值不变?

```
arsort($arr); print_r($arr);
```

15.\$mail = "gaofei@163.com"; 请将此邮箱的域(163.com)取出来并打印, 看最多能写出几种方法?

```
echo strstr($mail, '163');
echo substr($mail, 7);
$arr = explode("@", $mail); echo $arr[1];
```

16.php连接mysql之后, 如何设置mysql的字符集编码为utf8?

```
mysql_query("set names utf8");
```

17.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");
```

18.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?

```
1数据库库和表都用utf8_general_ci编码
2 php连接mysql, 指定数据库编码为utf8 mysql_query("set names 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" />
```

19.php中, 如何跳转到blog/view.php页面, 请用函数实现?

```
header("location:blog/view.php");
```

20.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()
```

21.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），php端接收数据的时候，应该如何处理？

```
htmlspecialchars($_POST['title']);
```

22.写出两种对变量加密的函数？

```
md5($str);          sha1($str);
```

23.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);
```

24.一个浏览器做多可以产生多少个cookie，每个cookie文件最大不能超过多少？

20个，每个做多不超过4K

25.设置或读取session之前，需要做什么？

可以直接在php.ini中开启session.auto_start = 1或者在页面头部用session_start();开启session，session_start()前面不能有任何输出，包括空行。

26.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。

27.\$str=" 你好baby好久不见了!";如何页面编码为utf-8格式，我要截取12个字符，不出现乱码，应该用什么函数？

```
echo $str2 = mb_substr($str,12);  
自定义函数吧
```

28.如何把2009-9-2 10:30:25变成unix时间戳？

```
$unix_time = strtotime("2009-9-2 10:30:25");//变成unix时间戳  
echo date("Y-m-d H:i:s",$unix_time);//格式化为正常时间格式
```

二、选择题

1.下面哪个函数可以打开一个文件，以对文件进行读和写操作？

(c)

(a) fgetc() (b) file_open() (c) fopen() (d) open_file()

2.下面哪个选项没有将 john 添加到users 数组中？

(b, d)

- (a) \$users[] = 'john' ;
- (b) array_add(\$users,' john');
- (c) array_push(\$users, 'john');
- (d) \$users ||= 'john' ;

3.下面的程序会输出什么?(1分)

c

```
<?php  
$num = 10;  
function multiply(){  
    $num = $num * 10;  
}  
multiply();  
echo $num;  
?>
```

(a) 100 (b) 没有任何输出 (c) 10 (d) 11

三、答题

1.使用php写一段简单查询，查出所有姓名为“张三”的内容放到一个数组中，并用foreach

把张三的信息打印到table表格中。

表名User

Name Tel Content Date

张三 13333663366 大专毕业 2006-10-11

张三 13612312331 本科毕业 2006-10-15

张四 021-55665566 中专毕业 2006-10-15

请根据上面的题目完成代码：

```
$link = mysql_connect('localhost','root','') or die('连接mysql失败');
mysql_select_db('database1');
mysql_query('set names utf8');
$sql = "select * from table1 where name='张三'";
$result = mysql_query($sql);
while( $row = mysql_fetch_assoc($result) ){
    $arr[] = $row;
}
$str .= '<table>';
foreach( $arr as $v ){
    $str .= '<tr><td>'.$v['name'].'</td><td>'.$v['age'].'</td><td>'.$v['class'].'</td></tr>';
}
$str .= '</table>';
echo $str;
```

2.写出 SQL语句的格式：插入，更新，删除 (4分)

表名User

Name Tel Content Date

张三 13333663366 大专毕业 2006-10-11

张三 13612312331 本科毕业 2006-10-15

张四 021-55665566 中专毕业 2006-10-15

(a) 有一新记录(小王 13254748547 高中毕业 2007-05-06)请用SQL语句新增至表中

```
$sql = "insert into User(Name,Tel,Content,Date) values('小王','13254748547', '高中毕业',' 2007-05-06')";
```

(b) 请用sql语句把张三的时间更新成为当前系统时间

```
$sql = "update User set Date='".time()." where Name='张三'";
```

(c) 请写出删除名为张四的全部记录

```
$sql = "delete from User where Name='张四'";
```

3.以下请用PHPMYADMIN完成

(一)创建新闻发布系统，表名为message有如下字段 (3分)

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

(二)同样上述新闻发布系统：表comment记录用户回复内容，字段如下 (4分)

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

```
SELECT t. *, (
SELECT count( * )
FROM COMMENT tt
WHERE tt.id = t.id
)num
FROM message t order by num desc
LIMIT 0 , 30 ;
```

(三)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 selectCategory($arr)
{
    If( empty($arr) ){
        Return "<select name='category'><option>暂无分类</option><
/selected>";
    }
    $str = "<select name='category'><option>文章分类</option>";
    Foreach($arr as $v){
        $str.="<option value='". $v['category_id']. "'>". $v['category_name'
]. "</option>";
    }
    $str .="</select>";
    Return $str;
}
```

题库四

1.类中如何定义常量、如何类中调用常量、如何在类外调用常量。

类中的常量也就是成员常量，常量就是不会改变的量，是一个恒值。
定义常量使用关键字const。
例如：`const PI = 3.1415326;`
无论是类内还是类外，常量的访问和变量是不一样的，常量不需要实例化对象，
访问常量的格式都是类名加作用域操作符号（双冒号）来调用。
即：类名 :: 类常量名；

2.接口和抽象类的区别是什么？

抽象类是一种不能被实例化的类，只能作为其他类的父类来使用。
抽象类是通过关键字 `abstract` 来声明的。
抽象类与普通类相似，都包含成员变量和成员方法，两者的区别在于，抽象类中至少要包含一个抽象方法，
抽象方法没有方法体，该方法天生就是要被子类重写的。
抽象方法的格式为：`abstract function abstractMethod();`
因为php中只支持单继承，如果想实现多重继承，就要使用接口。也就是说子类可以实现多个接口。
接口是通过`interface`关键字来声明的，接口中的成员常量和方法都是`public`的，方法可以不写关键字`public`，
接口中的方法也是没有方法体。接口中的方法也天生就是要被子类实现的。
抽象类和接口实现的功能十分相似，最大的不同是接口能实现多继承。在应用中选择抽象类还是接口要看具体实现。
子类继承抽象类使用`extends`，子类实现接口使用`implements`。

3.__call的参数有几个，类型是什么，意义是什么？

魔术方法`__call()`的作用是当程序调用一个不存在或不可见的成员方法时，php会先调用`__call()`方法，
将那个不存在的方法的方法名和参数都存储下来。
`__call()`包含两个参数，第一个参数是那个不存在的方法的方法名，是个字符串类型；
第二个参数是那个不存在的方法的所有参数，是个数组类型。
本人认为`__call()`方法的意义更多在于调试，可以定位到错误。同时可以捕捉异常，如果某个方法不存在，
则执行其它可替代方法。

4.构造方法、析构方法是如何定义的？

本文档使用 [看云](#) 构建

对象。

构造方法就是在实例化一个对象的同时自动执行的成员方法，作用就是初始化对象。

php5之前，一个跟类名完全相同的方法是构造方法，php5之后魔术方法__construct()就是构造方法。

如果类中没有定义构造方法，那么php会自动生成一个，这个自动生成的构造方法没有任何参数，没有任何操作。

构造方法的格式如下：

```
function __construct(){}
或者：function 类名(){}
构造方法可以没有参数，也可以有多个参数。
```

析构方法的作用和构造方法正好相反，是对象被销毁时被自动调用的，作用是释放内存。

析构方法的定义方法为：__destruct();

因为php具有垃圾回收机制，能自动清除不再使用的对象，释放内存，一般情况下可以不手动创建析构方法。

5.===是什么运算？

是php比较运算符中的一种，表示恒等，说明两个变量不只是数值上相等，而且两者的数据类型也相等。

如果\$a !== \$b, 那么说明两个变量要么数值不等，要么就是数据类型不相同。

6.__autoload()函数是如何运作的？

使用这个魔术函数的基本条件是类文件的文件名要和类的名字保持一致。

当程序执行到实例化某个类的时候，如果在实例化前没有引入这个类文件，那么就自动执行__autoload()函数。

这个函数会根据实例化的类的名称来查找这个类文件的路径，当判断这个类文件路径下确实存在这个类文件后

就执行include或者require来载入该类，然后程序继续执行，如果这个路径下不存在该文件时就提示错误。

使用自动载入的魔术函数可以不必写很多个include或者require函数。

7.网页静态化要用到哪类函数？

使用ob系列函数。这个问题需要到php高级才学。

8.PHP有几组函数来处理正则表达式？

preg系列函数可以处理。具体有以下几个：

```
string preg_quote ( string str [, string delimiter] )
```


转义正则表达式字符 正则表达式的特殊字符包括：. \ + * ?
 [^] \$ () { } = ! < > | :。
 preg_replace -- 执行正则表达式的搜索和替换
 mixed preg_replace (mixed pattern, mixed replacement, mixed subject [, int limit])
 preg_replace_callback -- 用回调函数执行正则表达式的搜索和替换
 mixed preg_replace_callback (mixed pattern, callback callback, mixed subject [, int limit])
 preg_split -- 用正则表达式分割字符串
 array preg_split (string pattern, string subject [, int limit [, int flags]])

9.试举正则表达式对字符串最常用的3种操作？

1. 利用正则表达式，实现对一个字符串的切割，切割后返回一个数组。如split()。
 2. 利用正则可以实现对一个字符串中符合的内容批量替换。如果替换为空则达到了过滤的作用。如preg_replace()。
 3. 利用正则判断一个字符串中是否含有符合的子字符串。如preg_match()。

10.字符串怎么转成整数，有几种方法？怎么实现？

强制类型转换：(整型)字符串变量名；
 直接转换：settype(字符串变量, 整型);
 intval(字符串变量);

11.php怎么定义函数？

使用关键字function加函数名，参数和函数体来定义函数。格式为：
 function 函数名(参数){函数体};

12.函数内怎么使用局部变量和全局变量？

局部变量是函数内部定义的变量，其作用域是所在的函数。如果函数外还有一个跟局部变量名字一样的变量，
 程序会认为它们两个是完全不同的两个变量。当退出函数的时候，其中的局部变量就同时被清除。

全局变量是定义在所有函数以外的变量，其作用域是整个php文件，但是在用户自定义的函数内部是无法使用的。
 如果一定要在用户自定义的函数内部使用全局变量，那么就需要使用global关键字声明。

也就是说如果在函数内的变量前加上global来修饰，那么函数内部就可以访问到这个全局变量，
 不仅可以利用这个全局变量进行运算而且可以对这个全局变量进行重新赋值。

13.字符串和数组怎么取长度？

字符串：strlen();
 count() -- 计算数组中的元素个数。
 可以使用count(数组名)或者count(数组名,1),如果有第二个参数，并且是数字1，则表示递归统计数组元素的个数。
 如果第二个参数是数字0，则等同于只有一个参数的count()函数。
 sizeof() -- count() 的别名

14.页面字符出现乱码，怎么解决？

1.首先考虑当前文件是不是设置了字符集。查看是不是meta标签中写了charset，如果是php页面还可以看看是不是在header()函数中指定了charset；
 2.如果设置了字符集（也就是charset），那么判断当前文件保存的编码格式是否跟页面设置的字符集保持一致，两者必须保持一致；
 3.如果涉及到从数据库提取数据，那么判断数据库查询时的字符集是否跟当前页面设置的字符集一致，两者必须统一。

15.php访问数据库有哪几步？

1.连接数据库服务器：
 mysql_connect('数据库服务器的主机名或ip','数据库服务器的用户名','数据库服务器的密码');
 2.选择数据库：
 mysql_select_db(数据库名);
 3.设置从数据库提取数据的字符集：
 mysql_query("set names utf8");
 4.执行sql语句：
 mysql_query(sql语句);
 5.关闭结果集，释放资源：
 mysql_free_result(\$result);
 6.关闭与数据库服务器的连接：
 mysql_close(\$link);

16.文件上传应该注意哪些细节？

1.首现要在php.ini中开启文件上传；
 2.在php.ini中有一个允许上传的最大值，默认是2MB。必要的时候可以更改；
 3.上传表单一定要记住在form标签中写上enctype="multipart/form-data"；
 4.要注意上传文件的大小、文件类型是否符合要求，上传后存放的路径是否存在。

17.怎么把文件保存到指定目录？怎么避免上传文件重名问题？

可以通过上传的文件名获取到文件后缀，然后使用时间戳+文件后缀的方式为文件重新命名，这样就避免了重名。
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。

18.session和cookie有什么区别？

`cookie`是保存在客户端机器的，对于未设置过期时间的`cookie`，`cookie`值会保存在机器的内存中，只要关闭浏览器则`cookie`自动消失。如果设置了`cookie`的过期时间，那么浏览器会把`cookie`以文本文件的形式保存到硬盘中，当再次打开浏览器时`cookie`值依然有效。

`session`是把用户需要存储的信息保存在服务器端。每个用户的`session`信息就像是键值对一样存储在服务器端，其中的键就是`sessionid`，而值就是用户需要存储信息。服务器就是通过`sessionid`来区分存储的`session`信息是哪个用户的。

两者最大的区别就是`session`存储在服务器端，而`cookie`是在客户端。`session`安全性更高，而`cookie`安全性弱。

19.session有什么用？怎么使用？

`session`在web开发中具有非常重要的份量。它可以将用户正确登录后的信息记录到服务器的内存中，当用户以此身份访问网站的管理后台时，无需再次登录即可得到身份确认。而没有正确登录的用户则不分配`session`空间，即便输入了管理后台的访问地址也不能看到页面内容。通过`session`确定了用户对页面的操作权限。

使用`session`的步骤：

1. 启动`session`：

使用`session_start()`函数来启动。

2. 注册会话：

直接给`$_SESSION`数组添加元素即可。

3. 使用会话：

判断`session`是否为空或者是否已经注册，如果已经存在则像普通

数组使用即可。

4. 删除会话：

1. 可以使用`unset`删除单个`session`；

2. 使用`$_SESSION=array()`的方式，一次注销所有的会话变量；

3. 使用`session_destroy()`函数来彻底销毁`session`。

20.cookie有什么用？怎么使用？

浏览速度。

1. 记录用户访问的部分信息
2. 在页面间传递变量
3. 将所查看的internet页存储在cookies临时文件夹中，可以提高以后的

创建cookie：

```
setcookie(string cookiename , string value , int expire);
```

读取cookie：

通过超级全局数组\$_COOKIE来读取浏览器端的cookie的值。

删除cookie：有两种方法

- 1.手工删除方法：

右击浏览器属性，可以看到删除cookies，执行操作即可将所有cookie文件删除。

- 2.setcookie()方法：

跟设置cookie的方法一样，不过此时讲cookie的值设置为空，有效时间为0或小于当前时间戳。

21.怎么定义常量？

```
define('变量名','值');
```

还可以使用constant()函数来定义一个常量。例如constant(常量名)。

22.break和continue的区别？

break可以终止循环。

continue没有break强大，只能终止本次循环而进入到下一次循环中。

23.循环语句有哪几种？

```
while(){};
do{}while();
for(){};
foreach(){};
```

24.数组怎么声明？怎么赋值？

数组的声明方式主要有两种。

- 1.通过array()函数声明数组；

可以通过key=>value的方式分别定义索引和值，也可以不定义数组的索引下标，只给出数组的元素值。

- 2.直接为数组元素赋值，不需要调用array()函数。例如：

```
$arr[0] = 1;
$arr[1] = 2;
```

25.说出数组涉及到的常用函数。

```

array -- 声明一个数组
count -- 计算数组中的单元数目或对象中的属性个数
foreach -- 遍历数组
list -- 遍历数组
explode -- 将字符串转成数组
implode -- 将数组转成一个新字符串
array_merge -- 合并一个或多个数组
is_array -- 检查是否是数组

print_r -- 输出数组
sort -- 数组排序

array_keys -- 返回数组中所有的键名
array_values -- 返回数组中所有的值
key -- 从关联数组中取得键名

```

26.数组有哪几种类型？

数字索引数组
 关联数组
 根据数组构造还可以分一维数组、二维数组、多维数组。

27.foreach 用法。

```

foreach($数组名 as $key=>$value){
    循环体
};

```

28.说出有哪几种分支语句。

```

if(){};
if(){}else{};
if(){}elseif(){}else{};
switch(){case1 ,case2,.....};

```

29.什么是全局变量，自定义函数中如何使用全局变量？

全局变量是定义在所有函数以外的变量，其作用域是整个php文件，但是在用户自定义的函数内部是无法使用的。

如果一定要在用户自定义的函数内部使用全局变量，那么就需要使用global关键字声明。

也就是说如果在函数内的变量前加上global来修饰，那么函数内部就可以访问到这个全局变量，

不仅可以利用这个全局变量进行运算而且可以对这个全局变量进行重新赋值。

30.字符串的常用函数？

```
trim();--去除空格
strlen();--字符串长度
substr();--截取字符串
str_replace();--替换字符串函数
substr_replace() -- 对指定字符串中的部分字符串进行替换
strstr();--检索字符串函数
explode();--分割字符串函数
strcasecmp();--比较字符串函数
```

31.PHP数据类型有哪些？

php支持8种原始数据类型。
包括：
四种标量类型（布尔型boolean，整型integer，浮点型float/double，字符串string）
两种复合类型（数组array，对象object）
两种特殊类型（资源resource，NULL）

32.如何定义一个变量？

使用\$和变量名表示，变量名区分大小写。变量名不可以以数字或者@等特殊字符开头。

33.怎么定义一个整数变量？

```
$变量名=整数值；
```

34.print与echo有什么区别？

echo可以一次输出多个值，中间用逗号隔开。而print一次只能输出一个值。

35.你是怎么理解正则表达式的？

正则表达式是用于描述字符排列模式的一种语法规则。正则表达式也叫做模式表达式。网站开发中正则表达式最常用于表单提交信息前的客户端验证。比如验证用户名是否输入正确，密码输入是否符合要求，email、手机号码等信息的输入是否合法。
在php中正则表达式主要用于字符串的分割、匹配、查找和替换操作。

36.你是如何认识ajax的？

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。

ajax的核心技术是XMLHttpRequest, 它是javascript中的一个对象。

1. 你如何理解smarty, smarty的运用过程中注意哪些细节?

Smarty是基于MVC概念的一种模板引擎, 它将一个页面程序分成了两部分来实现: 即视图层和控制层,

也就是说smarty技术将用户UI与php代码分离开。

这样程序员和美工各司其职, 互不干扰。

smarty运用过程中要注意以下几个问题:

1. 正确配置smarty。主要要配置smarty模板文件的路径;

2. php页面中使用assign赋值 和 display显示页面;

3. smarty模板文件中不允许出现php代码段, 所有的注释, 变量, 函数都要包含在定界符内。

1. 如果理解oop?

OOP(object oriented programming), 即面向对象编程, 其中两个最重要的概念就是类和对象。

世间万物都具有自身的属性和方法, 通过这些属性和方法可以区分出不同的物质。

属性和方法的集合就形成了类, 类是面向对象编程的核心和基础,

通过类就将零散的用于实现某个功能的代码有效地管理起来了。

类只是具备了某些功能和属性的抽象模型, 而实际应用中需要一个一个实体, 也就是需要对类进行实例化,

类在实例化之后就是对象。

OOP具有三大特点:

1. 封装性:

也称为信息隐藏, 就是将一个类的使用和实现分开, 只保留部分接口和方法与外部联系, 或者说只公开了一些供开发人员使用的方法。

于是开发人员只需要关注这个类如何使用, 而不用去关心其具体的实现过程, 这样就能实现MVC分工合作, 也能有效避免程序间相互依赖,

实现代码模块间松藕合。

本文档使用 [看云](#) 构建

2. 继承性：

就是子类自动继承其父级类中的属性和方法，并可以添加新的属性和方法或者对部分属性和方法进行重写。继承增加了代码的可重用性。

php只支持单继承，也就是说一个子类只能有一个父类。

3. 多态性：

子类继承了来自父级类中的属性和方法，并对其中部分方法进行重写。

于是多个子类中虽然都具有同一个方法，但是这些子类实例化的对象调用这些相同的方法后却可以获得完全不同的结果，这种技术就是多态性。

多态性增强了软件的灵活性。

题库五

1.在PHP中，当前脚本的名称（不包括路径和查询字符串）记录在预定义变量（1）中；而链接到当前页面的前一页面URL记录在预定义变量（2）中

```
<?php
//本页地址，SCRIPT_NAME也可以:php/test.php
echo $_SERVER['PHP_SELF']."<br />";
//链接到当前页面的前一页面的 URL 地址:
echo $_SERVER['HTTP_REFERER']."<br />";

//其它的见参考手册：语言参考》变量》预定义变量
//前执行脚本的绝对路径名:D:\inetpub\wwwroot\php\test.php
echo $_SERVER["SCRIPT_FILENAME"]."<br />";
//正在浏览当前页面用户的 IP 地址:127.0.0.1
echo $_SERVER["REMOTE_ADDR"]."<br />";
//查询（query）的字符串（URL 中第一个问号 ? 之后的内容）:id=1&bi=2
echo $_SERVER["QUERY_STRING"]."<br />";
//当前运行脚本所在的文档根目录:d:\inetpub\wwwroot
echo $_SERVER["DOCUMENT_ROOT"]."<br />";
?>
```

2.执行程序段<?php echo 8%(-2) ?>将输出__。

```
<?php
//参考手册》语言参考》运算符》算术运算符》%为取模运算,输出0
echo 8%(-2)."<br />";
//取模 $a % $b 在 $a 为负值时的结果也是负值。输出 -2
echo ((-8)%3)."<br />";
//输出2
echo (8%(-3))."<br />";
?>
```

3.在HTTP 1.0中，状态码 401 的含义是；如果返回“找不到文件”的提示，则可用 header 函数，其语句为。

答：401表示未授权;header("HTTP/1.0 404 Not Found");[见参考手册》函数参考》HTTP 函数》header]

4.数组函数 arsort 的作用是__；语句 error_reporting(2047)的作用是__。

答：arsort:对数组进行逆向排序并保持索引关系 error_reporting(2047)的作用是:report All errors and warnings

5. 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）：

```
<?php
    $script="以下内容不显示：<script language='javascript'>alert('cc');</script>";
    echo preg_replace("/<script[^>].*?>.??</script>/si", "替换内容", $script);
?>
```

6. 以Apache模块的方式安装PHP，在文件http.conf中首先要用语句____动态装载PHP模块，

然后再用语句____使得Apache把所有扩展名为php的文件都作为PHP脚本处理。
 答：LoadModule php5_module "c:/php/php5apache2.dll";AddType application/x-httpd-php .php
 见参考手册》目录》II. 安装与配置》6. Windows 系统下的安装》Microsoft Windows 下的Apache 2.0.x

7. 语句 include 和 require 都能把另外一个文件包含到当前文件中，它们的区别是；为了避免多次包含同一文件，可以用语句来代替它们。

答：在如何处理失败时，include() 产生一个警告而 require() 则导致一个致命错误；require_once()/include_once()

8. 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.

答：allow_call_time_pass_reference boolean :是否启用在函数调用时强制参数被按照引用传递，见参考手册》附录G

9. SQL 中LEFT JOIN的含义是,如果 tbl_user记录了学生的姓名(name)和学号(ID), tbl_score记录了学生（有的学生考试以后被开除了，没有其记录）的学号(ID)和考试成绩(score)以及考试科目(subject)，要想打印出各个学生姓名及对应的各科总成绩，则可以用SQL语句__.

答：自然左外连接

```
create database phpinterview;
```

```
use phpinterview
```

```
create table tbl_user
```

```
(
```

```

        ID                int                not null,
        name              varchar(50)         not null,
        primary key (ID)
    );
create table tbl_score
(
    ID                int                not null,
    score             dec(6,2)           not null,
    subject           varchar(20)        not null
);

```

```

insert into tbl_user (ID, name) values (1, 'beimu');
insert into tbl_user (ID, name) values (2, 'aihui');
insert into tbl_score (ID, score, subject) values (1, 90, '语文');
insert into tbl_score (ID, score, subject) values (1, 80, '数学');
insert into tbl_score (ID, score, subject) values (2, 86, '数学');
insert into tbl_score (ID, score, subject) values (2, 96, '语文');

```

```

select A.id,sum(B.score) as sumscore
from tbl_user A left join tbl_score B
on A.ID=B.ID
group by A.id

```

1. 在PHP中，heredoc是一种特殊的字符串，它的结束标志必须____

答：结束标识符所在的行不能包含任何其它字符除";"

11.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。

```

<?php
function my_scandir($dir)
{
    $files=array();
    if(is_dir($dir))
    {
        if($handle=opendir($dir))
        {
            while(($file=readdir($handle))!==false)
            {
                if($file!="." && $file!="..")
                {

```

```
        if(is_dir($dir."/".$file))
        {
            $files[$file]=my_scandir($dir."/".$file);
        }
        else
        {
            $files[]=$dir."/".$file;
        }
    }
    closedir($handle);
    return $files;
}
}
}
print_r(my_scandir("D:Program FilesInternet ExplorerMUI"));
?>
```

题库六

1 请说明 PHP 中传值与传引用的区别。什么时候传值什么时候传引用？

答：传值只是把某一个变量的值传给了另一个变量，而引用则说明两者指向了同一个地方。

2 在PHP中error_reporting这个函数有什么作用？

答：The error_reporting() function sets the error_reporting directive at runtime. PHP has many levels of errors, using this function sets that level for the duration (runtime) of your script.

3 请用正则表达式 (Regular Expression) 写一个函数验证电子邮件的格式是否正确。

答：

```
<?php
if(isset($_POST['action']) && $_POST['action']=='submitted')
{
    $email=$_POST['email'];
    if(!preg_match("/^(?:w+.)?*w+@(?:w+.)?*w+$/", $email))
    {
        echo "电子邮件检测失败";
    }
    else
    {
        echo "电子邮件检测成功";
    }
}
else
{
    ?>
<html>
<head><title>EMAIL检测</title>
<script type="text/javascript">
    function checkEmail(sText)
    {
        var reg=/^(?:w+.)?*w+@(?:w+.)?*w+$/;
        var email=document.getElementById(sText).value;
        if(!reg.test(email))
        {
            alert("电子邮件检测失败");
        }
        else
        {
            alert("电子邮件格式正确");
        }
    }
}
```

```

    }
</script>
</head>
<body>
<form action="<?php echo $_SERVER['PHP_SELF'] ?>" method="POST">
电子邮件:<input type="text" id="email" name="email" /><br />
<input type="hidden" name="action" value="submitted" />
<input type="button" name="button" value="客户端检测" onclick="checkEmail('
email')" />
<input type="submit" name="submit" value="服务器端检测" />
</form>
</body>
</html>
<?php
}
?>

```

4 简述如何得到当前执行脚本路径，包括所得到参数。

```

<?php
echo "http://".$_SERVER['SERVER_NAME'].$_SERVER['PHP_SELF']."?".$_SERVER[
'QUERY_STRING'];
//echo "http://".$_SERVER['HTTP_HOST'].$_SERVER['REQUEST_URI'];
?>

```

5 有一个一维数组，里面存储整形数据，请写一个函数，将他们按从大到小的顺序排列。要求执行效率高。并说明如何改善执行效率。（该函数必须自己实现，不能使用php函数）

```

<?php
function BubbleSort(&$arr)
{
    $cnt=count($arr);
    $flag=1;
    for($i=0;$i<$cnt;$i++)
    {
        if($flag==0)
        {
            return;
        }
        $flag=0;
        for($j=0;$j<$cnt-$i-1;$j++)
        {
            if($arr[$j]>$arr[$j+1])
            {
                $tmp=$arr[$j];
                $arr[$j]=$arr[$j+1];
                $arr[$j+1]=$tmp;
                $flag=1;
            }
        }
    }
}

```

```
        }  
    }  
}  
$test=array(1,3,6,8,2,7);  
BubbleSort($test);  
var_dump($test);  
?>
```

6 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

答：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。使用代码优化工具啦

算法题

面试题——算法题:

1、插入排序（一维数组）基本思想：每次将一个待排序的数据元素，插入到前面已经排好序的数列中的适当位置，使数列依然有序；直到待排序数据元素全部插入完为止。 示例：

[初始关键字] [49] 38 65 97 76 13 27 49

J=2(38) [38 49] 65 97 76 13 27 49

J=3(65) [38 49 65] 97 76 13 27 49

J=4(97) [38 49 65 97] 76 13 27 49

J=5(76) [38 49 65 76 97] 13 27 49

J=6(13) [13 38 49 65 76 97] 27 49

J=7(27) [13 27 38 49 65 76 97] 49

J=8(49) [13 27 38 49 49 65 76 97]

```
function insert_sort($arr){
    $count = count($arr);
    for($i=1; $i<$count; $i++){
        $tmp = $arr[$i];
        $j = $i - 1;
        while($arr[$j] > $tmp){
            $arr[$j+1] = $arr[$j];
            $arr[$j] = $tmp;
            $j--;
        }
    }
    return $arr;
}
```

2、选择排序（一维数组）基本思想：每一趟从待排序的数据元素中选出最小（或最大）的一个元素，顺序放在已排好序的数列的最后，直到全部待排序的数据元素排完。 示例：

[初始关键字] [49 38 65 97 76 13 27 49]

第一趟排序后 13 [38 65 97 76 49 27 49]

第二趟排序后 13 27 [65 97 76 49 38 49]

第三趟排序后 13 27 38 [97 76 49 65 49]

第四趟排序后 13 27 38 49 [49 97 65 76]

第五趟排序后 13 27 38 49 49 [97 97 76]

第六趟排序后 13 27 38 49 49 76 [76 97]

第七趟排序后 13 27 38 49 49 76 76 [97]

最后排序结果 13 27 38 49 49 76 76 97

```
function select_sort($arr){
    $count = count($arr);
    for($i=0; $i<$count; $i++){
        $k = $i;
        for($j=$i+1; $j<$count; $j++){
            if ($arr[$k] > $arr[$j]) $k = $j;
        }
        if($k != $i){
            $tmp = $arr[$i];
            $arr[$i] = $arr[$k];
            $arr[$k] = $tmp;
        }
    }
    return $arr;
}
```

3、冒泡排序（一维数组）基本思想：两两比较待排序数据元素的大小，发现两个数据元素的次序相反时即进行交换，直到没有反序的数据元素为止。排序过程：设想被排序的数组 $R[1..N]$ 垂直竖立，将每个数据元素看作有重量的气泡，根据轻气泡不能在重气泡之下的原则，从下往上扫描数组 R ，凡扫描到违反本原则的轻气泡，就使其向上“漂浮”，如此反复进行，直至最后任何两个气泡都是轻者在上，重者在下为止。示例：

49 13 13 13 13 13 13 13

38 49 27 27 27 27 27 27

65 38 49 38 38 38 38 38

97 65 38 49 49 49 49 49

76 97 65 49 49 49 49 49

13 76 97 65 65 65 65 65

27 27 76 97 76 76 76 76

49 49 49 76 97 97 97 97

```
function bubble_sort($array){
    $count = count($array);
    if ($count <= 0) return false;
    for($i=0; $i<$count; $i++){
        for($j=$count-1; $j>$i; $j--){
            if ($array[$j]<$array[$j-1]){
```

```

        $tmp = $array[$j];
        $array[$j] = $array[$j-1];
        $array[$j-1] = $tmp;
    }
}
}
return $array;
}

```

4、快速排序（一维数组）基本思想：在当前无序区R[1..H]中任取一个数据元素作为比较的“基准”（不妨记为X），用此基准将当前无序区划分为左右两个较小的无序区：R[1..I-1]和R[I..H]，且左边的无序子区中数据元素均小于等于基准元素，右边的无序子区中数据元素均大于等于基准元素，而基准X则位于最终排序的位置上，即 $R[1..I-1] \leq X \leq R[I..H]$ ，当R[1..I-1]和R[I..H]均非空时，分别对它们进行上述的划分过程，直至所有无序子区中的数据元素均已排序为止。示例：

初始关键字 [49 38 65 97 76 13 27 49]

第一次交换后 [27 38 65 97 76 13 49 49]

第二次交换后 [27 38 49 97 76 13 65 49]

J向左扫描，位置不变，第三次交换后 [27 38 13 97 76 49 65 49]

I向右扫描，位置不变，第四次交换后 [27 38 13 49 76 97 65 49]

J向左扫描 [27 38 13 49 76 97 65 49]

（一次划分过程）

初始关键字 [49 38 65 97 76 13 27 49]

一趟排序之后 [27 38 13] 49 [76 97 65 49]

二趟排序之后 [13] 27 [38] 49 [49 65] 76 [97]

三趟排序之后 13 27 38 49 49 [65] 76 97

最后的排序结果 13 27 38 49 49 65 76 97

各趟排序之后的状态

```

function quickSort(&$arr){
    if(count($arr)>1){
        $k=$arr[0];
        $x=array();
        $y=array();
        $_size=count($arr);
        for($i=1;$i<$_size;$i++){
            if($arr[$i]<=$k){
                $x[]=$arr[$i];
            }elseif($arr[$i]>$k){
                $y[]=$arr[$i];
            }
        }
    }
}

```

```

    }
    $x=quickSort($x);
    $y=quickSort($y);
    return array_merge($x,array($k),$y);
  }else{
    return$arr;
  }
}

```

5、希尔排序 (shell sort) — $O(n \log n)$

```

functionshell_sort(&$arr){
    if(!is_array($arr))return;$n=count($arr);
    for($gap=floor($n/2);$gap>0;$gap=floor($gap/=2)){
        for($i=$gap;$i<$n;++$i){
            for($j=$i-$gap;$j>=0&&$arr[$j+$gap]<$arr[$j];$j-=$gap){
                $temp=$arr[$j];
                $arr[$j]=$arr[$j+$gap];
                $arr[$j+$gap]=$temp;
            }
        }
    }
}

```

6、二分查找

```

/**
 * 二分算法查找
 * @param array $array 要查找的数组
 * @param int $min_key 数组的最小下标
 * @param int $max_key 数组的最大下标
 * @param mixed $value 要查找的值
 * @return boolean
 */
function bin_search($array,$min_key,$max_key,$value){
    if($min_key <= $max_key){
        $key = intval(($min_key+$max_key)/2);
        if($array[$key] == $value){
            return true;
        }elseif($value < $array[$key]){
            return bin_search($array,$min_key,$key-1,$value);
        }else{
            return bin_search($array,$key+1,$max_key,$value);
        }
    }else{
        return false;
    }
}

```

7、线性表的删除（数组中实现）

```
function delete_array_element($array, $i)    {
    $len = count($array);
    for ($j=$i; $j<$len; $j++){
        $array[$j] = $array[$j+1]
    }
    array_pop($array);
    return $array;
}
```

8、字符串长度

```
function strlen($str)    {
    if ($str == '') return 0;
    $count = 0;
    while (1){
        if ($str[$count] != NULL){
            $count++;
            continue;
        }else{
            break;
        }
    }
    return $count;
}
```

9、字符串翻转

```
function strrev($str)    {
    if ($str == '') return 0;
    for ($i=(strlen($str)-1); $i>=0; $i--){
        $rev_str .= $str[$i];
    }
    return $rev_str;
}
```

10、字符串比较

```
function strcmp($s1, $s2)    {
    if (strlen($s1) < strlen($s2)) return -1;
    if (strlen($s1) > strlen($s2)) return 1;
    for ($i=0; $i<strlen($s1); $i++){
        if ($s1[$i] == $s2[$i]){
            continue;
        }else{

```

```

        return false;
    }
}
return 0;
}

```

11、查找字符串

```

function strstr($str, $substr) {
    $m = strlen($str);
    $n = strlen($substr);
    if ($m < $n) return false;
    for ($i=0; $i<=($m-$n+1); $i++){
        $sub = substr($str, $i, $n);
        if (strcmp($sub, $substr) == 0) return $i;
    }
    return false;
}

```

12、字符串替换

```

function str_replace($substr, $newsubstr, $str) {
    $m = strlen($str);
    $n = strlen($substr);
    $x = strlen($newsubstr);
    if (strpos($str, $substr) == false) return false;
    for ($i=0; $i<=($m-$n+1); $i++){
        $i = strpos($str, $substr);
        $str = str_delete($str, $i, $n);
        $str = str_insert($str, $i, $newstr);
    }
    return $str;
}

```

13、插入一段字符串

```

function str_insert($str, $i, $substr) {
    for($j=0; $j<$i; $j++){
        $startstr .= $str[$j];
    }
    for ($j=$i; $j<strlen($str); $j++){
        $laststr .= $str[$j];
    }
    $str = ($startstr . $substr . $laststr);
    return $str;
}

```

14、删除一段字符串

```
function str_delete($str, $i, $j){
    for ($c=0; $c<$i; $c++){
        $startstr .= $str[$c];
    }
    for ($c=($i+$j); $c<strlen($str); $c++){
        $laststr .= $str[$c];
    }
    $str = ($startstr . $laststr);
    return $str;
}
```

15、复制字符串

```
function strcpy($s1, $s2){
    if (strlen($s1)==NULL || !isset($s2)) return;
    for ($i=0; $i<strlen($s1); $i++){
        $s2[] = $s1[$i];
    }
    return $s2;
}
```

16、连接字符串

```
function strcat($s1, $s2){
    if (!isset($s1) || !isset($s2)) return;
    $newstr = $s1;
    for($i=0; $i<count($s); $i++){
        $newstr .= $st[$i];
    }
    return $newsstr;
}
```

17、简单编码函数（与php_decode函数对应）

```
function php_encode($str) { if ($str==" " && strlen($str)>128) return false; for($i=0; $i<strlen($str); $i++){ $c = ord($str[$i]); if ($c>31 && $c<107) $c += 20; if ($c>106 && $c<127) $c -= 75; $word = chr($c); $s .= $word; } return $s; }
```

18、简单解码函数（与php_encode函数对应）

```
function php_decode($str) {
```

```

    if ($str==' ' && strlen($str)>128) return false;
    for($i=0; $i<strlen($str); $i++){
        $c = ord($word);
        if ($c>106 && $c<127) $c = $c-20;
        if ($c>31 && $c<107) $c = $c+75;
        $word = chr($c);
        $s .= $word;
    }
    return $s;
}

```

19、简单加密函数 (与php_decrypt函数对应)

```

function php_encrypt($str)    {
    $encrypt_key = 'abcdefghijklmnopqrstuvwxyz1234567890';
    $decrypt_key = 'ngzqtcobmuhelkpdawxfyivrsj2468021359';
    if (strlen($str) == 0) return false;
    for ($i=0; $i<strlen($str); $i++){
        for ($j=0; $j<strlen($encrypt_key); $j++){
            if ($str[$i] == $encrypt_key[$j]){
                $enstr .= $decrypt_key[$j];
                break;
            }
        }
    }
    return $enstr;
}

```

21、简单解密函数 (与php_encrypt函数对应)

```

function php_decrypt($str)    {
    $encrypt_key = 'abcdefghijklmnopqrstuvwxyz1234567890';
    $decrypt_key = 'ngzqtcobmuhelkpdawxfyivrsj2468021359';
    if (strlen($str) == 0) return false;
    for ($i=0; $i<strlen($str); $i++){
        for ($j=0; $j<strlen($decrypt_key); $j++){
            if ($str[$i] == $decrypt_key[$j]){
                $enstr .= $encrypt_key[$j];
                break;
            }
        }
    }
    return $enstr;
}

```

