

《电商网站建设与优化》课程设计

授课班级		课次	1
授课方式	理论课。讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目(教学章、节或主题): PHP 环境配置、数据类型 常用PHP运算类型、PHP条件、循环语句的介绍与应用			
教学目的、要求(含思政育人目标) (分掌握、熟悉、了解三个层次): 了解PHP 环境的配置,wamp环境的设置。 掌握PHP的数据类型及其数据的运算规则。 掌握PHP的条件语句、循环语句的用法及语法规则。			
教学重点及难点: 动态网站的环境配置与PHP的条件语句、循环语句的用法及语法规则。			
教学基本内容及方法手段			
<p>PHP 环境搭配和代码调试</p> <p>1、PHP的优势和背景</p> <p>PHP 是能让你生成动态网页的工具之一.PHP代表: 超文本预处理器(PHP: Hypertext Preprocessor) .PHP是完全免费的,不用花钱,你可以从 PHP 官方网站点()自由下载.</p> <p>你不必把头埋进100多页的文档中努力学习才可以写出一个象样的程序。只要了解一些基本的语法和语言特色,你就可以开始你的PHP编码之旅了.如果更好的学习建议可以先看看些 HTML 和 SQL的基础。</p> <p>2、PHP 环境的搭配</p> <p>所需的工具: WampServer 5 集成环境</p> <p>Wamp5是Apache+PHP+Mysql 在Windows下的集成环境,拥有简单的图形和SQLiteMarager 1.2.0满了大部分PHPer的需求...</p> <p>介绍一下 wamp的虚拟目录、服务图标的显示、环境测试</p> <p>3、PHP常用工具和数据库常用工具</p> <p>PHP开发工具: Zend Stodio,PHPedit ,EditPlus 2,easyclipse, DW 等</p> <p>MYSQL工具: MYSQL Administrator</p> <p>4、书写和调试简单的PHP代码</p> <p>从语法上看,PHP语言近似于C语言.可以说,PHP是借鉴C语言的语法特征,由C语言改进而来的.我们可以混合编写PHP代码和HTML代码,不仅可以将PHP脚本嵌入到 HTML 文件中,我们甚至还可以把 HTML 标签也嵌入在 PHP 脚本里.以下是你可以采用的几种方法.你可以选用其中一种你最适合的并且就这样坚持这种方法!从HTML中分离以下是可以使用的方法:</p> <p><? . . . ? > 短 标 签 模 式</p> <p><?php . . . ? > 标 准 模 式</p>			

<script language="php"> . . . </script> 类似于 javascript 模式

<% . . . %> 修改模式，也可以支持

在 PHP 中用 “;” 来分隔语句。那些从 HTML 中分离出来的标志也表示语句的结束。

注释

PHP 支持 C, C 和 Unix 风格的注释方式:

/* C, C 风格多行注释 */

// C 风格单行注释

Unix 风格单行注释

例子:

```
<?php
```

```
echo "Hello World!"; 或 phpinfo();
```

```
?>
```

5、如何处理简单的 PHP 错误

- 1、你要把 PHP.ini 里的 display_errors ? = ? On 才可以显示错误位置
- 2、习惯使用 echo 或者 print 打印，每一步输出看看结果是否正确。
- 3、学会使用注释来屏蔽符号来调试

```
// .....
/*..... */
#.....
```

4、要懂得看错误信息

错误的显示信息的行数及什么错位置。

PHP 的数据类型 源码调试

1、PHP 基本语法和数据类型

(1)、PHP 基本语法:

- 1、htm 和 php 混编

```
<font color=red>我好</font>
```

- 2、一个语句以; (分号结束，英文半角)

- 3、如何定义一个变量，和变量的使用

```
$abc=9;//以$开始，以 A-Z 或 a-z
```

```
Echo $abc;
```

(2)、PHP 数据运算类型

四种标量类型:

boolean (布尔型) 理解为真假型

```
$bo=TRUE; $bo=FALSE;//赋值为 0 时为假，大于 0 为真
```

integer (整型)

```
$bo=1; $bo=-12;
```

float (浮点型，也作“double”) 理解为小数型

```
$bo=1.001; $bo=3.1415926;
```

string (字符串)

```
$bo=“这段字符串 or EN Word”; //双引号支持转义字符，单引号不支持。
```

两种复合类型:

array (数组)

```
$bo=array(1, 2, 3, 4); $bo=array(“A” =>1 , “B” =>2);
```

object (对象)

源码调试

- 1、解压源码到网站目录
- 2、按照数据库
- 3、调试成功
- 2、学习一个 PHP 源码调试 (discuz 论坛)

常用 PHP 运算类型介绍与应用

1、算术运算 //理解为数学运算

例: $5 * 6 - 12$
 $2*(28+1)$

注: 算术运算遵循数学运算规则:从左到右, 先算乘除后算加减, 遇到括号先算括号内.

2、赋值运算 //把一个值写入一个变量

例: $\$a=1;$
 $\$a+=2;$
 $\$a*=3;$

注: 把右边的值赋予左的变量。

3、比较运算 /用来确定两个数之间的关系

例: $1==2$
 $3!=2$
 $5<3$
 $"ok"=="ok"$

$1=== '1'$ //值相等且类型相同, 用 echo 输出时真为 1, 假不输出。

注: 比较运算得到的值为布尔值

4、逻辑运算//先将比较的两边转换成布尔类型, 再执行他们的关系

例: $1 \&\& 1$
 $0 \text{ or } 1$
 $1 \text{ xor } 1$
 $!\$a$

注: 逻辑运算得到的值为布尔值

5、递增递减运算 //只操作变量的一种运算

例: $\$a++$ // $\$a=\$a+1$
 $++\$a$
 $\$a--$ // $\$a=\$a-1$
 $--\$a$

PHP 条件语句的介绍与应用

1、if 条件语句

程序中最常见的, 大部分语言中都存在的一种条件语句, 在开发中起着至关重要的作用, 格式如下:

if ...else...	if ...else...	if ...elseif...
<pre>if(expr) echo TRUE else echo FALSE</pre>	<pre>if(expr) { echo TRUE } else { echo FALSE</pre>	<pre>if(expr) { echo TRUE } elseif(expr) { echo CON } else{</pre>

	}	echo FALSE
--	---	------------

2、switch 条件语句

在多个条件当中使用 Switch 更加精巧，格式如下：

<pre>switch (\$i) { case 0: echo "输出 0"; break; case 1: echo "输出 1"; break; case 2: echo "输出 2"; break; default:echo "其他默认输出"; }</pre>	<pre>if (\$i == 0) { echo "输出 0"; } elseif (\$i == 1) { echo "输出 1"; } elseif (\$i == 2) { echo "输出 2"; }</pre>
--	---

PHP 循环语句的介绍与应用

1、break n 循环控制语句 //也可以理解为循环中断语句，跳出循环语句

break: 跳出一层循环
 break n ; 跳出 N 条

```
for($a=1;$a<10;$a++)
{
    if($a==5) break;
    echo"循环".$a."</br>";
}
```

2、do...while 循环语句

<pre>do{ } while(expr)</pre>	<pre>while(expr){ }</pre>
先执行一次循环再判断条件	先判断条件执再行一次循环

for 循环语句//for 循环是 PHP 中最复杂的循环结构。

```
for (expr1; expr2; expr3) {
    .....}
for ($i = 1; $i <= 10; $i++) {
    echo $ i;
// $i = 1: 无条件执行
// $i <= 10, $i++: 循环条件
}
```

作业、讨论题、思考题：PHP环境的配置，wamp环境的设置, 练习PHP的条件语句、循环语句的用法及语法规则的应用。

课后小结：本节课主要讲解了 PHP 环境的配置，wamp 环境的设置，介绍了 PHP 的数据类型及 PHP 各种运算及运算规则。重点介绍了 PHP 的条件语句、循环语句的用法及语法规则。

《电商网站建设与优化》课程设计

授课班级		课次	2
授课方式	理论课讨论课□实验课□习题课□其他□	课时安排	3
授课题目(教学章、节或主题): PHP数组、函数及Mysql 简介和创建新的数据库与数据库中的常用SQL语句			
<p>教学目的、要求(含思政育人目标) (分掌握、熟悉、了解三个层次):</p> <p>熟练掌握 PHP数组的定义、数组的遍历、foreach 的应用。</p> <p>掌握创建数据库的方法及其基本语句。</p> <p>掌握数据库中的常用SQL语句。</p> <p>了解计算机软件从业人员应当具备的职业道德守则,为进军软件行业做准备;</p> <p>理解并敬重工匠精神,在学习中努力发扬工匠精神;</p> <p>了解PHP 强大的生态体系,与华为孟晚舟事件相联系。</p> <p>新时代与个人成才的关系;</p> <p>通过软件行业发展前景,引发学生对未来的职业愿景,激发学生对社会主义核心价值观的认同感。</p>			
教学重点及难点: creat、select、update、insert语句的应用。			
教学基本内容及方法手段			
<p>● 思政元素的融入(德育点)</p> <p>1、新时代与个人成才的关系;</p> <p>2、通过软件行业发展前景,引发学生对未来的职业愿景,激发学生对社会主义核心价值观的认同感。</p> <p>· 课程思政教学素材</p> <p>1、世界上十大黑客的经历——引出: 计算机软件从业人员应当具备的职业道德规范;</p> <p>2、PHP的发展历程——引出: 工匠精神;</p> <p>3、美国的芯片垄断引出: 民族认同感。</p> <p>4、工匠精神,敬业求精</p> <p>工匠精神是一种职业精神,它是职业道德、职业能力、职业品质的体现,是从业者的一种职业价值取向和行为表现。应告诉学生在学好程序设计技术之后,走上工作岗位会成为程序员、软件系统运维人员、软件测试员、售前售后服务人员等.在这些职位岗位上,要发挥工匠精神,精益求精地将程序开发、系统运维、程序测试、需求分析及技术问题处理等工作内容完成好,</p>			

保证软件系统运行时正确、稳定，保证客户的需求被精确采集和纳入软件开发计划，保证软件运行时遇到问题能被及时解决。引导学生在学习时，将知识夯实、精技强能，方能在今后工作中本领过硬，不出纰漏，工作成果令用户满意。引导学生认识到，作为职业人，其专注、敬业、责任担当对完成好本职工作，进而促进软件行业整体的高水平、优质化发展具有重要意义。

PHP 数组的创建修改使用

1、创建和修改数组 / 多维数组

PHP 中的数组是一个关键字或值的集合（值可以是任何一种类型，包括另一个数组）

PHP 中使用 array 来创建一个数组：

```
array( key=>value , key=>value …… )
```

例子：

```
$arr = array (3, 5, 7, 9, 6);
```

```
$arr = array ("id"=>2, "title"=>3);
```

用方括号的语法来修改数组： \$arr[] = value

```
$arr = array("a" => "理论", "b"=>"教程", "c" => "第二次课");
```

```
$arr['a'] = "PHP 学习";
```

```
$arr['c'] = "第二讲";
```

介绍一下 print_r() 函数

```
$arr=array(array(2,4),array(3,5));
```

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

```
Array ( [0] => Array ( [0] => 2 [1] => 4 ) [1] => Array ( [0] => 3 [1] => 5 ) )
```

2、数组与数组的函数

使用 count 函数统计数组条数 count (\$arr);

使用 is_array 函数判断数组 is_array (\$arr);

另一种创建数组的函数 explode explode ("key" , value); //分解函数

```
$a="1986-1983-2010";
```

```
$arr=explode("-", $a);
```

```
Print_r($arr);
```

3、使用 foreach 遍历数组

遍历数组很多种方法，可以使用 For 循环，while 循环，最巧妙的一种遍历数组 foreach 给我们带来了跟多方便

```
foreach ( $arr as $key => $value ){
```

```
.....
```

```
}
```

```
$arr=array(1986, 1983, 2010);
```

```
foreach($arr as $key =>$value)
```

```
{
```

```
    echo "需要的关键字 $key 需要的值 $value <br>";
```

```
}
```

PHP 函数和自定义函数

1、PHP 函数介绍


```

{
    echo "真";
}
else
{
    echo "假";
}

```

(4) 引用返回值使用&符号

Mysql 简介和创建新的数据库

1、MYSQL 简介与概要

MySQL 是一个小型关系型数据库管理系统，开发者为瑞典 MySQL AB 公司，现在已经被 Sun 公司收购，支持 FreeBSD、Linux、MAC、Windows 等多种操作系统，与其他的大型数据库例如 Oracle、DB2、SQL Server 等相比功能稍弱一些。

- 1、可以处理拥有上千万条记录的大型数据
- 2、支持常见的 SQL 语句规范
- 3、可移植行高，安装简单小巧
- 4、良好的运行效率，有丰富信息的网络支持
- 5、调试、管理，优化简单（相对其他大型数据库）

2、访问 MYSQL 的几种途径

知识点：MYSQL 默认使用的是 3306 端口

- 1、使用工具 MYSQL 官方的工具访问：MySQL Administrator tool

安装：步骤 下载

Backup project//备份

Restore //导入

MySQL query//浏览表和数据库

- 2、使用第三方工具访问：mysqlxexec 、 mysql 连接器

- 3、使用 WEB 方式访问：phpMyAdmin //最常见

比较：

使用客户端工具操作效率高，安全性高，使用 WEB 方式简单方便，不受限制。

3、介绍数据库中常用字段类型

- 整数型：TINYINT, SMALLINT, INT, BIGINT
- 小数型：FLOAT, DOUBLE, DECIMAL(M, D)
- 字符型：CHAR, VARCHAR
- 日期型：DATETIME , DATE, TIMESTAMP
- 备注型：TINYTEXT , TEXT , LONGTEXT

4、创建一个简单的数据库

创建一个数据库命令：CREATE TABLE

删除一个数据库命令：DROP TABLE

修改一个字段名称：RENAME TABLE `原字段` TO `新字段` ；

删除一个字段：ALTER TABLE `表` DROP `原字段`

插入一个字段：ALTER TABLE `表` ADD `新字段` ... AFTER `原字段`

```

CREATE TABLE `test` (
    `id`      int(10) NOT NULL auto_increment,
    `uid`     varchar(10) NOT NULL default '0',
    `regdate` date NOT NULL,

```

```
`remark` text NULL,  
PRIMARY KEY (`id`)
```

)

实例讲解:

1. 将命令复制到 phpmyadmin 中建立表
2. 用客户端工具建立表//设计几个字段

浏览/结构/SQL/搜索、插入、导出/import/操作.

导入数据库文件

1、SELECT 查询语句和条件语句

SELECT 查询字段 FROM 表名 WHERE 条件

查询字段: 可以使用通配符*、字段名、字段别名

表名: 数据库.表名, 表名

常用条件: = 等于、<>不等于、in 包含、not in 不包含、like 匹配、
BETWEEN 在范围、not BETWEEN 不在范围 <、>

条件运算: and、or、()

2、排序, 分组, 指针查询, 计算

分组语句: group by 字段

排序语句: order by 字段, 字段 ASC / DESC

指针查询: limit 初始值, 结束值

计算:

COUNT(*) 统计函数

MAX(*) 最大值函数

MIN (*) 最小值函数

AVG(*) 平均值函数

SUM(*) 累计值函数 (Σ)

3、Insert 插入语句

insert into 表名 (字段..., ...) values(值..., ...)

insert into 表名 values(值..., ...)

插入时须考虑清楚字段类型避免插入后出现缺值, 乱码现象

2、Update 更新语句

UPDATE 表名 SET 字段 = 值 WHERE 条件 limit

常用条件: = 等于、<>不等于、in 包含、not in 不包含、like 匹配、
BETWEEN 在范围、not BETWEEN 不在范围

条件预算: and、or、()

5、Delete 删除语句

DELETE FROM 表名 WHERE 条件 limit

常用条件: = 等于、<>不等于、in 包含、not in 不包含、like 匹配//%、
BETWEEN 在范围、not BETWEEN 不在范围

条件预算: and、or、()

MYSQL 在 PHP5 中的应用

1、PHP 与 mysql 建立链接

php.ini 加载 mysql 组件:

extension=php_mysql.dll 前的; 去掉

extension_dir = " " 路径是否正确

PHP 链接 mysql 函数

mysql_connect: 开启 MySQL 链接

```
$conn=@mysql_connect("localhost","root","") or die("链接错误");
```

mysql_select_db: 打开一个数据库

```
mysql_select_db("ww",$conn);
```

注意: @ 和 or die 隐藏错误 和 条件显示

mysql_connect("主机", "用户名", "密码")

mysql_select_db("打开数据库",连接标识符);

如果不特别声明连接标识符, 则默认为是上一次打开的连接。

2、如何去执行一个 SQL 语句

mysql_query (SQL 语句 ,连接标识符);

说明:mysql_query 用来根据连接标识符向该数据库服务器的当前数据库发送查询, 如果连接标识符默认, 则默认为是上一次打开的连接。

返回值: 成功后返回一个结果标识符, 失败时返回 false。

```
$sql="INSERT INTO test(id,name,regdate) values('','huafan',now())";
```

```
mysql_query($sql,$conn);
```

3、两种查询函数 array / row 区别

格式: mysql_fetch_row(result);

说明: mysql_fetch_row 用来查询结果的一行保存至数组, 该数组下标从 0 开始, 每一个数组元素对应一个域。通过循环, 可以将查询结果全部获得。

```
$sql="SELECT * FROM `test`";
```

```
$query=mysql_query($sql,$conn);
```

```
$row=mysql_fetch_row($query);
```

```
print_r($row);
```

格式: mysql_fetch_array(result);

说明: mysql_fetch_array 和 mysql_fetch_row 功能基本相同, 只不过它除了可以用从 0 开始的偏移量作索引, 还可以用域名作索引。

值返回下一行的所有域值, 并将其保存至一个数组中, 没有行时返回 false。

```
$sql="SELECT * FROM `test`";
```

```
$query=mysql_query($sql,$conn);
```

```
$row=mysql_fetch_array($query);
```

```
echo $row['name'];
```

输出所有内容:

```
$sql="SELECT * FROM `test`";
```

```
$query=mysql_query($sql,$conn);
```

```
while($row=mysql_fetch_array($query))
```

```
{
```

```
echo $row['name']."    ".$row['regdate']."<br>.<hr>";
```

```
}
```

mysql_query("set names 'GBK'"); 解决中文乱码

```
$conn=@mysql_connect("localhost","root","") or die("链接错误");
```

```
mysql_select_db("ww",$conn);
```

```
mysql_query("set names 'GBK'");
```

```
$sql="SELECT * FROM `test`";
```

```
$query=mysql_query($sql,$conn);
```

```
while($row=mysql_fetch_array($query))
```

```
{
    echo $row['name']. "    ".$row['regdate']. "<br>". "<hr>";
}
```

4、其他常用 Mysql 函数介绍

mysql_num_rows 用于计算查询结果中所得行的数目

```
echo mysql_num_rows($query);
```

mysql_insert_id 传回最后一次使用 INSERT 指令的 ID。

mysql_tablename 取得数据库名称

mysql_error 返回错误信息

mysql_close 关闭 MySQL 链接

学习制作 PHP+MYSQL 留言板

1、设计一个完整程序的流程

2、HTML 部分表单标签的使用

```
<form action=""
method="GET" >
<input type="text"
size=10 name="user"/>
<input type="text"
size=20
name="title"/>
<textarea
name="content"></textare
rea>
<input type="submit"
name="submit"/>
</form>
```

3、PHP 里的页面相互调用

require() 与 **require_once()**

通常放在 PHP 程序的最前面，PHP 程序在执行前，就会先读入 require 所指定引入的文件，如果出现错误是致命的。

include() 与 **include_once()**

可以放在 PHP 程序的任何一个位置，PHP 程序在执行到时，才会先读入 include 所指定引入的文件，如果出现错误将会提示。

```
require('con.php');
```

```
include('con.php');
```

3、PHP 接收参数的几种方式

PHP5 在默认的情况下接收参数是需要使用

```
$_GET['value'];
```

```
$_POST['value'];
```

还可以在 PHP.ini 文件中的

将 register_globals = Off

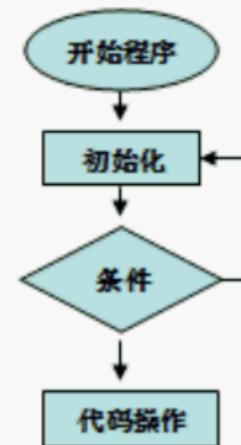
改 register_globals = on

可以直接使用，\$value 的值

设计流程:



程序设计:



作业、讨论题、思考题：制作 PHP+MYSQL 留言板

课后小结： 本节课讲解了 PHP 数组的创建修改，用 foreach 遍历数组，PHP 函数的应用和如何自定义函数, Mysql 的简介和创建新的数据库。创建一个数据库命令：CREATE TABLE, 删除一个数据库命令：DROP TABLE, 修改一个字段名称：RENAME TABLE `原字段` TO `新字段` ;删除一个字段： ALTER TABLE `表` DROP `原字段`, 插入一个字段： ALTER TABLE `表` ADD `新字段` ... AFTER `原字段`。

介绍了 SELECT 查询语句和条件语句，排序，分组，指针查询，计算；Insert 插入语句，Update 更新语句，Delete 删除语句，PHP 与 mysql 建立链接：
\$conn=@mysql_connect("localhost","root","") or die("链接错误");，如何去执行一个 SQL 语句：mysql_query (SQL 语句 ,连接标识符);，两种查询函数 array / row 区别，其他常用 Mysql 函数介绍如 mysql_num_rows 用于计算查询结果中所得行的数目 echo mysql_num_rows(\$query);mysql_insert_id 传回最后一次使用 INSERT 指令的 ID。mysql_tablename 取得数据库名称，mysql_error 返回错误信息，mysql_close 关闭 MySQL 链接。

《ASP 动态网站开发》课程教案

授课班级		课次	3
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目（教学章、节或主题）： 制作 PHP+MYSQL 留言板			
教学目的、要求（含思政育人目标）（分掌握、熟悉、了解三个层次）： 了解设计一个完整程序的流程。 掌握 HTML 部分表单标签的使用与 PHP 里的页面相互调用 掌握 PHP 接收参数的几种方式及提交表单 Javascript 验证 树立正确的技能观，努力提高自己的职业技能，为社会和人民造福，绝不能利用自己的技能去做违法犯罪之事； 培养学生的软件工匠精神，在潜移默化中培育社会主义核心价值观，提高综合职业素养，树立社会主义职业精神； 学习机房上课守则，要求同学们遵守学校的各项规章制度，强化制度约束，学会责任担当； 理解事物的联系是普遍存在的，引导学生用类比的方法进行知识的迁移。			
教学重点及难点： PHP 接收参数的几种方式及提交表单 Javascript 验证。			
教学基本内容及方法手段			
● 思政元素的融入（德育点） 1、树立正确的技能观，努力提高自己的职业技能，为社会和人民造福，绝不能利用自己的技能去做违法犯罪之事； 2、培养学生的软件工匠精神，在潜移默化中培育社会主义核心价值观，提高综合职业素养，树立社会主义职业精神。			
● 课程思政教学素材 1、通过对学院往届学生因不遵守校规，严重违反学校制度而被开除的案例一引出：制度约束，成			

年人要学会为自己的行为负责；

2、通过与前续课程的对比学习—引出：知识迁移的学习方法。

3、职业素养，行为合规

职业素养是人类在社会活动中需要遵守的行为规范。职业道德、职业思想、职业行为习惯是职业素养中最根基的部分。引导学生在学习时注意程序的书写格式、变量与方法的命名方式、合理添加注释、合理规划程序工程文件，这些都是合格的软件开发从业人员的基本素质。引导学生注重职业道德，尊重他人的知识产权。平时未经同学允许，不随意拷贝他人的程序成果；工作后，不偷窥和复制同事的开发成果，不盗取公司的研发成果。使学生明确：职业素养是一个职业人的立身之本。在学生时代，就应该不断提升个人修养和思想道德水平，着力锻造良好的职业素养。

实例操作我们的留言板

创建表：

```
CREATE TABLE `message` (
  `id` tinyint(1) NOT NULL auto_increment,
  `user` varchar(25) NOT NULL,
  `title` varchar(50) NOT NULL,
  `content` tinytext NOT NULL,
  `lastdate` date NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=gbk AUTO_INCREMENT=1
```

con. php //数据库配置

```
$conn=@mysql_connect("localhost","root","") or die("连接失败");
mysql_select_db("liuyanban",$conn);
mysql_query("set names 'GBK'");
```

add. php //操作文件

```
include("conn.php");
if($_POST['submit'])
{
  $sql="insert into message(id,user,title,content,lastdate)".
  "values('','$_POST[user]','$_POST[title]','$_POST[content]',n
  ow())";
  mysql_query( $sql);
  echo"发表成功";
}
?>
<form action="add.php" method="post">
用户: <input type="text" size="10" name="user" /><br>
标题: <input type="text" name="title" /><br>
内容: <textarea name="content" > </textarea><br>
  <input type="submit" name="submit" value="发布留言"/>
</form>
```

list. php //列表文件

```
<?php
include("conn.php");
?>
```


作业、讨论题、思考题： 设计一个留言板

课后小结： 本节介绍了设计一个完整程序的流程，HTML 部分表单标签的使用，PHP 里的页面相互调用，接收参数的几种方式，提交表单 Javascript 验证，str_replace 替换函数。

讲解了 htmlspecialchars 格式化 html，md5 加密函数与 COOKIE 的使用和登陆。

《ASP 动态网站开发》课程教案

授课班级		课次	4
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目（教学章、节或主题）： PHP+MYSQL 分页原理与 PHP 上传原理及应用及文件操作			
教学目的、要求（含思政育人目标）（分掌握、熟悉、了解三个层次）： 了解 SQL 语句中的 limit 用法与学习分页的一种公式。 掌握 FORM 标签 enctype 属性与\$_FILES 系统函数 掌握 move_uploaded_file 函数与 is_uploaded_file 函数 了解 PHP 部分文件操作函数与 unlink() rmdir() 删除函数 通过介绍软件公司 PHP 工程师的招聘条件，让学生了解 PHP 程序开发规范的重要性，培养学生的职业素质和道德规范； 软件行业规划解析，培养学生的软件工匠精神。			
教学重点及难点： 分页公式与文件上传原理			
教学基本内容及方法手段			
● 思政元素的融入（德育点） 1、通过介绍软件公司 PHP 工程师的招聘条件，让学生了解 PHP 程序开发规范的重要性，培养学生的职业素质和道德规范； 2、软件行业规划解析，培养学生的软件工匠精神。 3、团结协作，合作共赢 有人说，在学生时代一个人与同学的关系，就是今后职场中他与同事的关系。引导学生在平时的技术学习与实践中，组成研究小组，也要能够接纳那些不是那么合得来或者不太愿意合作的合伙人。在今后的职场中，在身处的软件开发团队中，员工形形色色，脾气秉性和行为方式各异，怎样协同攻关、合作共赢，是每个职业人必须处理好的问题。			
1、SQL 语句中的 limit 用法 SELECT * FROM table limit 开始位置 ， 操作条数 SELECT * FROM table limit 0 ， 20 SELECT * FROM table limit 10 ， 20			
2、学习分页的一种公式 (1) 分页原理 所谓分页显示，也就是讲数据库中的结果集，一段一段显示出来			

(2) 需要的条件

怎么分段, 当前在第几段 (每页有几条, 当前再第几页)

前 10 条记录: `select * from table limit 0,10`

第 11 至 20 条记录: `select * from table limit 10,10`

第 21 至 30 条记录: `select * from table limit 20,10`

(3) 得到公式

(当前页数 - 1) X 每页条数, 每页条数

`Select * from table limit ($Page- 1) * $PageSize, $PageSize`

3、parse_url() 解析 URL 函数

`parse_url()` 是讲 URL 解析成有固定键值的数组的函数

`arg=value#anchor");`

`print_r($ua);`

结果:

Array

```
(
    [scheme] => http      //协议
    [host] => hostname   //域名
    [user] => username   //用户名
    [pass] => password   //密码
    [path] => /path      //路径
    [query] => arg=value //取得的值
    [fragment] => anchor //
)
```

4、\$_SERVER["REQUEST_URI"]函数 //除域名以外的值

预定义服务器变量的一种, 所有`$_SERVER`开头的都叫做预定义的服务器变量
`REQUEST_URI`的作用是取得的当前 URL, 也就除域名外后面完整的地址路径。

</home.php?id=23&cid=22>

Echo `$_SERVER["REQUEST_URI"]`

结果为: </home.php?id=23&cid=22>

简单分页

```
$pagesize=5;
$sql="select * from message limit 5, $pagesize";
$query=mysql_query($sql);
while($row=mysql_fetch_array($query))
{
echo "<hr><b>".$row['user']."|".$row['title'];
}
}
```

分页实例

```
/* 分页程序 */
include("conn.php");
include("head.php");
$pagesize=10;//每页需要显示几条信息
$url=$_SERVER["REQUEST_URI");//显示文件目录
$url=parse_url($url);//把地址分析出来,分析成有固定键值的数组。
$url=$url['path'];//echo $url;取出翻页的地址
```

```

$numq=mysql_query("select * from message");//查询出所有记录。
$num=mysql_num_rows($numq);//记录总条数
//echo $num;// 查看结果
if($_GET['page'])//对page的判断
{
    $pageval=$_GET['page'];
    $page=($pageval-1)*$pagesize;//得出page的值
    $page.=',';
}
if($num>$pagesize)
{
    if($pageval<=1) $pageval=1;
    echo "共 $num 条".<a href=$url?page=".( $pageval-1) .
">上一页</a><a href=$url?page=".( $pageval+1) .">下一页</a>";
}
$sql="select * from message limit $page $pagesize";
$query=mysql_query($sql);
while($row=mysql_fetch_array($query))
{
echo "<hr><b>".$row['user']."|".$row['title'];
}

```

PHP 上传原理及应用

1、FORM 标签 enctype 属性

表单中 enctype="multipart/form-data"的意思，是设置表单的 MIME 编码。默认情况，这个编码格式是 application/x-www-form-urlencoded（通过浏览器传递文件），不能用于文件上传；只有使用了 multipart/form-data（可以传递文件，文件是二进制编码格式），才能完整的传递文件数据。

```
<form action="" enctype="multipart/form-data" method="post"
name="upform">
```

上传文件：

```
<input name="upfile" type="file">
<input type="submit" value="上传">
</form>
```

2、\$_FILES 系统函数//将文件转换为数组

\$_FILES['myFile']['name'] 客户端文件的原名称。

\$_FILES['myFile']['type'] 文件的 MIME 类型，例如"image/gif"。

```

image/jpg
image/jpeg
image/png
image/pjpeg
image/gif
image/bmp
image/x-png
application/x-zip-compressed
.....

```

\$_FILES['myFile']['size'] 已上传文件的大小，单位为字节。

`$_FILES['myFile']['tmp_name']` 储存的临时文件名，一般是系统默认。存储路径

`$_FILES['myFile']['error']` 该文件上传相关的错误代码。

0; 文件上传成功。

1; 超过了文件大小 php.ini 中。

2; 超过了文件大小或 MAX_FILE_SIZE 选项指定的值。

3; 文件只有部分被上传。

4; 没有文件被上传。

5; 上传文件大小为 0。

3、move_uploaded_file 函数//上传后移动文件到目标位置的函数

move_uploaded_file(临时文件, 目标位置和文件名);

4、is_uploaded_file 函数//判断上传 MIME 类型的文件函数

is_uploaded_file(MIME);

上传文件实例

```
<?php
//上传文件判断
if(is_uploaded_file($_FILES['upfile']['tmp_name'])) {
$upfile=$_FILES['upfile'];
//print_r($_FILES['upfile']);测试结果
$name=$upfile['name'];
$type=$upfile['type'];
$size=$upfile['size'];
$tmp_name=$upfile['tmp_name'];
$error=$upfile['error'];
//echo $type; //测试输出结果
switch($type){
    case 'image/jpg' : $ok=1;break;
    case 'image/jpeg' : $ok=1;break;
    case 'image/gif' : $ok=1;break;
    case 'image/png' : $ok=1;break;
    case 'image/pjpeg' : $ok=1;break;
}
if($ok && $error==0)
{
    move_uploaded_file($tmp_name, 'up/'.$name);
    echo "上传成功! ";
}
}
?>
<form action="" enctype="multipart/form-data" method="post"
name="upform">
上传文件:
<input name="upfile" type="file">
<input type="submit" value="上传">
</form>
```

文件操作

1、PHP 部分文件操作函数(fopen , fread , filesize, fwrite, fclose)

(1) fopen 打开文件函数

fopen (路径和文件名, 打开方式); R-只读 W-写入 A-读写

(2) fread 读取文件内容

fread (打开的文件, 结束位置); 字节为单位

(3) filesize 读取文件大小, 字节为计量单位

filesize (路径和文件名);

(4) fwrite 写入文件内容

fwrite (路径和文件名, 写入的内容);

(5) fclose 关闭打开的文件

fclose (路径和文件名);

2、unlink() rmdir() 删除函数

unlink() 删除文件函数 : unlink(路径和文件名)

rmdir() 删除目录函数 : rmdir(路径和目录名)

实例讲解:

建立一个模板文件 tmp.htm

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 3.2//EN">
```

```
<html>
```

```
<head>
```

```
<title>标题:{title}</title>
```

```
</head>
```

```
<body>
```

```
此新闻的内容:{content}
```

```
</body>
```

```
</html>
```

生成文件 wenjiancaozuo.php

```
$fp=fopen("tmp.htm","r");//只读打开模板
```

```
$str=fread($fp,filesize("tmp.htm"));//读取模板中的内容分
```

```
//echo $str;测试读取结果
```

```
$str=str_replace("{title}","新标题",$str);
```

```
$str=str_replace("{content}","新内容",$str);//替换内容
```

```
fclose($fp);
```

```
$handle=fopen('news.html','w');
```

```
fwrite($handle,$str);
```

```
fclose($handle);
```

带循环的生成批文件

```
$con=array(array('新闻标题','新闻内容'),array('新闻标题 2','新闻内容 2'),array('新闻标题 3','新闻内容 3'));
```

```
foreach($con as $id=>$val)
```

```
{ $title=$val[0];
```

```
$content=$val[1];
```

```
$path=$id.'.htm';
```

```
$fp=fopen("tmp.htm","r");//只读打开模板
```

```
$str=fread($fp,filesize("tmp.htm"));//读取模板中的内容分
```

```
//echo $str;测试读取结果
```

```
$str=str_replace("{title}","新标题",$str);
```

```
$str=str_replace("{content}","新内容",$str);//替换内容
```

<pre>fclose(\$fp); \$handle=fopen(\$path,'w'); fwrite(\$handle,\$str); fclose(\$handle); echo "生成成功"; } 删除文件 unlink('1.htm');</pre>
作业、讨论题、思考题：制作一个留言板设计分页与文件上传
<p>课后小结：本节介绍了 SQL 语句中的 limit 用法与学习分页的一种公式，parse_url() 解析 URL 函数，\$_SERVER["REQUEST_URI"]，FORM 标签 enctype 属性，\$_FILES 系统函数</p> <p>讲解了文件操作函数 move_uploaded_file 函数，is_uploaded_file 函数，PHP 部分文件操作函数(fopen , fread , filesize, fwrite, fclose)，unlink() rmdir() 删除函数。</p>

《ASP 动态网站开发》课程教案

授课班级		课次	5
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目（教学章、节或主题）： 面向对象开发的学习（一）			
<p>教学目的、要求（含思政育人目标）（分掌握、熟悉、了解三个层次）：</p> <p>了解什么是面向对象，什么是类的概念</p> <p>掌握使用 CLASS 创建一个类及类的成员属性和成员方法，对象中的 \$this 关键字</p> <p>掌握使用 new 函数对象实例化，创建一个和多个对象</p> <p>掌握初始化对象 __construct() 和析构函数 __destruct()</p> <p>了解封装 关键字 :public,protected,private 及封装相关函数：__set() , __get()</p> <p>熟悉继承关键字：extends，PHP 继承的规则，基类方法重载和父类方法访问</p> <p>掌握抽象方法和抽象类的定义，抽象类和方法使用规则</p> <p>通过介绍软件公司 PHP 工程师的招聘条件，让学生了解 PHP 程序开发规范的重要性，培养学生的职业素质和道德规范；</p> <p>软件行业规划解析，培养学生的软件工匠精神。</p>			
<p>教学重点及难点： CLASS 创建一个类及类的成员属性和成员方法，对象中的 \$this 关键字与 new 函数对象实例化，创建一个和多个对象。初始化对象 __construct() 和析构函数 __destruct()。</p>			
教学基本内容及方法手段			

● 思政元素的融入（德育点）

1、通过介绍软件公司 PHP 工程师的招聘条件，让学生了解 PHP 程序开发规范的重要性，培养学生的职业素质和道德规范；

2、软件行业规划解析，培养学生的软件工匠精神。

3、大国战略，技术强国

党的十八届五中全会通过的“十三五”规划《建议》，明确提出实施网络强国战略以及与之密切相关的“互联网+”行动计划。国家正着力实现关键技术自主可控，为维护国家安全、网络安全提供技术保障。中国信息化需求巨大，但在一些关键技术领域如操作系统、芯片技术、CPU 技术等方面，还难以做到自主可控，对国家安全造成威胁。引导学生明确：建设网络强国，不仅仅是靠网络技术，还要有软件技术等其他各类技术的支撑。引导学生作为软件技术专业的一员，应更加明晰专业人才的培养目标，更加明确专业领域内工作岗位和工作内容的社会价值，自觉树立远大职业理想，将职业生涯、职业发展脉络与国家发展的历史进程融合起来。

1、什么是面向对象，什么是类

起初，“面向对象”是专指在程序设计中采用封装、继承、抽象等设计方法。可是，这个定义显然不能再适合现在情况。面向对象的思想已经涉及到软件开发的各个方面。如，面向对象的分析（OOA, Object Oriented Analysis），面向对象的设计（OOD, Object Oriented Design）、以及我们经常说的面向对象的编程实现（OOP, Object Oriented Program）。

传统开发方法存在问题

1. 软件重用性差，2. 软件可维护性差，3. 无很好的扩展性

面向对象的要素

1. 抽象性，2. 封装性，3. 共享性，4. 强调对象结构而不是程序结构

面向对象的三大特点（封装，继承，多态）缺一不可

什么是类

我们可以理解成一个功能集合菜单，我们通过类来实现生成我们的方法

2、使用 CLASS 创建一个类

```
class 方法名 {
```

```
.....
```

```
}
```

3、类的成员属性和成员方法

```
class MyPc{
public $name;
var $price;
function vod() {
return "PHP100视频教程";
}
}
```

4、使用 new 函数对象实例化

```
new 对象名( 参数 )
```

例：

```
class MyPc{
var $key;//public $key 等价
public $name;
function vod()
{
```

```

        echo "面向对象学习";
    }
}
$pc1=new MyPc();
$pc1->key="php";//访问成员属性
echo $pc->key;
$pc1->vod(); //调用成员方法

```

5、创建一个和多个对象

创建一个和多个对象我们只需要多次使用 new 函数，进行对类的实例化

类	对象
<pre> class MyPc{ public \$name; public \$price; function vod() { return "PHP学习教材"; } } </pre>	<pre> \$pc1 = new MyPc(); \$pc2 = new MyPc(); \$pc3 = new MyPc(); </pre>

6、对象中的\$this关键字

\$this 关键字是用来访问当前对象中的对象属性和对象方法的系统变量

<pre> class MyPc{ public \$name; public \$price; function vod() { \$this->name; } \$this->vod() ; } </pre>	<p>我们可以理解为\$this 是在对象中特殊的一种使用构造函数和变量的方法。 \$this 仅能在当前对象中使用</p>
--	--

```

class MyPc{
    public $name;
    function vod()
    {
        echo $this->name.",播放电影". "<br>";
    }
    function game()
    {
        echo $this->vod(). "玩游戏";
    }
}

```

```

$pc1=new MyPc();
$pc1->name="家用电脑! ";//访问成员属性
$pc1->vod(); //调用成员方法
$pc2=new MyPc();
$pc2->name="公司电脑! ";
$pc2->game()

```

7、初始化对象 __construct()

我们在创建一个类有时候需要初始化一些操作，这个时候我们需要使用初始化函数，在 PHP5 中有两种初始化方法。

1、原 PHP4 中我们需要建立一个与类名称相同的一个对象方法：若类改名字则内部初始化，也要改名字。	2、在 PHP5 中继承了 PHP4 但为了更好的针对面向对象开发有了固定函数。
<pre>class MyPc{ function MyPc () { 初始化操作 } }</pre>	<pre>class MyPc{ function__construct() { 初始化操作 } }</pre>

```
class MyPc{
    public $name;
    public $type;
    function __construct($name,$type)
    {
        $this->name=$name;
        $this->type=$type;
    }
    function vod()
    {
        echo $this->name.$this->type.",播放电影". "<br>";
    }
    function game()
    {
        echo $this->vod(). "玩游戏";
    }
}
$pc1=new MyPc("家用电脑","台式机");
$pc1->vod(); //调用成员方法
$pc2=new MyPc("公司电脑","笔记本");
$pc2->game();
```

8、析构函数 __destruct() //能够在对象释放时自动被调用的方法被称为析构函数。

PHP4 支持析构函数，通过登录在 PHP 运行终止时用 register_shutdown_function() 调用的函数，只有类似的实行方法。PHP5 正式支持析构函数，能够在类中指定对象释放时的动作——可以理解为垃圾回收机制。

当对象内部的操作执行完毕的时候，__destruct() 被调用，然后对象所使用的内存被释放出来。**规则：后进先出**

```
class MyPc {
    .....
    function __destruct() {
        .....
    }
}
function __destruct()
```

```

    {
        echo "<br>====". $this->name;
    }

```

输出结果

====公司电脑

====家用电脑

提前释放: \$pc1=null;

9、类的封装和应用

● 封装关键字: public, protected, private

封装是把一些相关的属性和行为隐藏起来, 从而得到保护和安全。

public 表示全局, 类内部外部子类都可以访问;

protected 表示受保护的, 只有本类或子类或父类中可以访问;

private 表示私有的, 只有本类内部可以使用;

```

class MyPc {
    public $name;          //全局属性, 任何位置都可以引用
    protected $price;     //保护属性, 只有本类或子类或父类中可以访问
    private function vod() //私有属性, 只有本类内部可以使用
    {
        $this->name;
    }
    .....
}

```

● 封装相关函数: __set() , __get()

某些特定操作的时候需要访问和赋值封装的类型, 这个时候我们就需要其他的函数帮我们完成这些操作, PHP 为我们提供了个方法名: __set() , __get()

__set() 取得当前类中封装过私有属性或者方法重新执行或赋值操作

__set(\$Name, \$Value)

__get() 取得当前类中封装过属性或方法并转换成共有属性

实例讲解

```

class MyPc {
    private $name;
    function __construct($name='')
    {
        return $this->name=$name;
    }
    function __get($name) //间接访问私有变量
    {
        return $this->name."====="; //可以改变变量的输出方式
    }
    function __set($n, $v) //设置私有变量的值
    {
        if($v=="zzh") {$this->$n=$v;}
    }
}
$pc1=new MyPc("我的电脑");

```

```
$pcl->name="fdsgdsgdsg";
```

```
echo $pcl->name;
```

10、类的继承和应用

● 继承关键字：extends

PHP 类的继承，我们可以理解成共享被继承类的内容。PHP 中使用 extends 单一继承的方法！被继承的类我们叫做父类（基类）继承者成为子类（派生类）。

● PHP 继承的规则

依次被继承，class3 拥有 class1、class2 所以功能和属性，避免方法和属性重名。

```
class root{
    function dayin()//可以用封装类 protected function dayin()
    {
        return "root print <br>";
    }
}

class son extends root {
    function dayin2()
    {
return "son print <br>";//return $this->dayin()."son print <br>";
    }
}

$p=new son();
echo $p->dayin();
```

● 基类方法重载和父类方法访问

因为属于向下继承的原理，基类不能使用派生类里内容，这个时基类的一些方法不能完成我们的一些派生类的功能，我们就可以进行方法重载避免了，新建方法带来的混乱。

方法重载我们也可以理解方法覆盖，在派生类里使用与基类方法重名的方法名称执行重载，重载时我们需要调用原始基类内容，再增加新内容，我们可以使用基类名 :: 方法名称

```
class son extends root {
    function dayin()
    {
        return root::dayin()."son print <br>";
    }
}
```

11、类的抽象方法和类

● 抽象关键字：abstract

抽象就是无法确切的说明，但又有一定的概念或者名称，在 PHP 中声明一个抽象类或者方法我们需要使用 abstract 关键字。

● 抽象方法和抽象类的定义

一个类中至少有一个方法是抽象的，我们称之为抽象类。所以如果定义抽象类首先定义抽象方法。

```
abstract class class1{
    abstract function fun1();
    .....
}
```

- 1、类中至少有一个抽象方法。
- 2、抽象方法不允许有{ }。
- 3、抽象方法前面必须要加 abstract。

● 抽象类和方法使用规则

抽象类的几个特点：

- 1、不能被实例化，只能被继承。
- 2、继承的派生类当中要把所有抽象方法重载才能实例化。

不可实例化	可实例化
<pre>abstract class cl1{ abstract function fun1(); }</pre>	<pre>class cl2 extends cl1{ //继承 function fun1() { //重载所有抽象方法 {}}</pre>

```
abstract class cl1{
    abstract function fun1();
    abstract function fun2();
    abstract function fun3();
    function ok()
    {}
}
class cl2 extends cl1{
    function fun1()//重载抽象方法
    {echo "第一个";}
    function fun2()
    {echo "第二个";}
    function fun3()
    {echo "第三个";}}
$p=new cl2();
$p->fun1();
```

作业、讨论题、思考题：建立一个类，定义三个属性和三个方法

课后小结：本节课介绍了什么是面向对象，什么是类，使用 CLASS 创建一个类，类的成员属性和成员方法，使用 new 函数对象实例化。创建一个和多个对象，对象中的 \$this 关键字，初始化对象 __construct()，析构函数 __destruct()。

讲解了封装关键字：public,protected,private,封装相关函数：__set()，__get()

，继承关键字：extends, PHP 继承的规则，基类方法重载和父类方法访问，抽象关键字：abstract, 抽象方法和抽象类的定义，抽象类和方法使用规则。

《ASP 动态网站开发》课程教案

授课班级		课次	6
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目（教学章、节或主题）： PHP 面向对象开发的学习（二）与制作自己的 PHP+MYSQL 的类			
<p>教学目的、要求（含思政育人目标）（分掌握、熟悉、了解三个层次）：</p> <p>了解关键字：final, self, static, const, interface, implements, instanceof。 掌握多态的介绍与优势、多态的简单应用。 熟悉对象描述的配置，对象方法的异常处理及克隆对象的应用、自动载入对象的方法。 熟悉常用 MYSQL 函数、使用类封装常用 SQL 操作，简化代码及认识面向对象在开发中的重要作用。</p> <p>理解团队协作的重要性，了解团结合作是计算机应用程序开发小组成员所遵循的基本规范之一；</p> <p>理解分类学的思想是人类解决复杂问题时最常用的方法之一，在学习生活中做好分类计划，合理规划时间；</p> <p>理解沟通能力是一个人生存与发展的必备技能，了解软件开发过程中，沟通能力的重要性，学习高效沟通的三大秘诀。</p>			
教学重点及难点：MYSQL 函数、使用类封装常用 SQL 操作，简化代码及认识面向对象在开发中的重要作用与关键字的应用。			
教学基本内容及方法手段			
<p>● 思政元素的融入（德育点）</p> <p>1、软件开发设计的人员组成及分工—引出：团结协作精神；</p> <p>2、分类学原理—引出：有效进行时间管理；</p> <p>3、一个往届毕业生的工作真实经历—引出：良好的沟通能力的重要性。</p> <p>4、典型应用——多元互动，媒体融合</p> <p>党的十八大以来，以习近平同志为核心的党中央高度重视传统媒体和新兴媒体的融合发展，习近平总书记多次在不同场合强调要利用新技术、新应用创新媒体传播方式。引导学生关注最新的传播技术、传播方式，这些都是信息技术在新闻传播行业的典型应用。引导学生了解融媒体系统平台的建设，离不开软件技术——特别是 PHP 技术，使学生被技术的魅力所吸引，建立起专业自豪感。</p> <p>类中常用关键字</p> <p>1、关键字：final</p> <p>用来定义类和方法的一个重要关键字，当定义类的时候该类将不能被继承，当用来定义方法的时候该方法将不能被重载</p> <pre>final class class1{ //该类将不能被继承</pre>			

```

final function fun1() { //该方法将不能被重载
    .....
}
.....
}

```

实例讲解

```

final class MyPc { //加上 final 后就不能被继承
    public $name='我的电脑';
    final function power() //加上 final 后就不能重载
    {
        echo $this->name.", 电脑打开中.....";
    }
}
class my extends MyPc {
    // function power() { //重载抽象方法
//echo MyPc::power(). "*****";
//}
}

```

```
$p=new my();
```

```
$p->power();
```

2、关键字：self

用来用类访问当前类中的内容的关键字，类似与\$this 关键字，但\$this 是需要类实例化以后才可以使⤵用，self 可以直接访问当前类中的内部成员

self :: 类内部成员（属性或者方法）等价与 类名 :: 类内部成员（属性或者方法）

注意：因为没有实例化类访问内部属性或者方法是没有意义的，所以 self 一般用来访问类中的：静态成员，常量，或者其他定义内容。

3、关键字：static

用来定义类的静态属性或方法，可以在类未被实例化时使用，静态属性单独占用内存而不会因创建多个对象时而导致同样的方法或者属性重复占用。

```

class class1 {
    static $name;
    static function fun1() { //静态方法内部禁止出现非静态内容
        .....
    }
}

```

类内部访问静态成员 类名称 :: 静态成员 self :: 静态成员

类外部访问静态成员 类名称 :: 静态成员

实例讲解

```

final class MyPc { //加上 final 后就不能被继承
static $name='我的电脑';
    static function power() //静态方法，使用$this 关键字会出错
    {
        //echo $this->name.", 电脑打开中.....";this 关键字就不能使用
        echo self::$name.", 电脑打开中....."; //用 self 方法访问静态属性。
    }
}

```

```
//echo MyPc::$name;//静态属性可以被访问输出
//MyPc::$name="你的电脑";//静态属性可以被访问修改
$p=new MyPc();
$p->power();
```

4、关键字：const

用来定义类中的常量，类似 PHP 外部定义常量的关键字 define()；Const 只能修饰类当中的成员属性！

```
class class1{
    const NAME;
    function fun1(){
        .....
    }
}
```

实例讲解

```
final class MyPc{
const NAME='我的电脑';
    static function power()
    {
        echo self::NAME.",电脑打开中.....";
    }
}
```

```
$p=new MyPc();
$p->power();
```

注意事项：

- 1、建议大写常量
- 2、常量不使用\$符号

类中接口的应用

1、接口的介绍与创建

关键字：interface

接口：一种成员属性全部为抽象的特殊抽象类，在程序中同为规范的作用。

抽象类

- 1、类中至少有一个抽象方法
- 2、抽象方法前需加 abstract

接口

- 1、类中全部为抽象方法
- 2、抽象方法前不用加 abstract
- 3、接口抽象方法属性为 public
- 4、成员属性必须为常量

共同点：

- 1、本身都不能被实例化，必须被继承或者引用。
- 2、继承或者引用后，需要把所有抽象方法重载方可使用。

接口有自己的创建关键字：interface 与普通类的创建类似

```
interface demo { //声明接口类
    const NAME="名字"; //对象属性必须为常量
    function Fun1();
    function Fun2();//对象方法全部为抽象方法，前面无须加 abstract
```

}

实例讲解

```
interface demo{
    const NAME="名称";//声明一个变量试试
    function fun1();//前加一个 abstract 试试，建一个普通的方法试试
}
```

2、接口的引用与规范

关键字：implements

接口引用区别与之前我们的学的类继承关键字 extends, 继承只能是单一性, 而接口可以使用关键字: implements 多个引用并用逗号, 分开。

设已有接口 demo , demo2, demo3 父类 root

1、普通类引用接口例子:

```
class mypc implements demo , demo2 ,demo3{
    .....
}
```

2、抽象类引用接口例子:

```
abstract class mypc implements demo , demo2 , demo3{
    .....
}
```

3、继承父类引用接口并存例子:

```
class mypc extends root implements demo , demo2 , demo3{
    .....
} //先继承后接口，单继承多接口
```

4、接口与接口继承:

```
interface demo3 extends demo{
    .....
}
```

类中多态的应用**1、多态的介绍与优势**

介绍: 多态性是继抽象和继承后, 面向对象语言的第三个特征。从字面上理解, 多态的意思是“多种形态”, 简单来说, 多态是具有表现多种形态的能力的特征, 在 OO 中是指“语言具有根据对象的类型以不同方式处理。”

优势: OOP 的模式并不仅仅是把很多函数和功能集合起来, 目的而是使用类, 继承, 多态的方式描述我们生活中的一种情况。从而使得我们的代码更具有“物”的意义。帮助我们减少一些重复性的代码和条件语句的判断。

2、运算符：instanceof

PHP 一个类型运算符。instanceof 用来测定一个给定的对象是否来自指定的对象类。

```
class A { }
class B { }
$thing = new A;
if ($thing instanceof A) { //判断变量$thing 是否来自类 A
    echo 'A';
}
if ($thing instanceof B) {
```

```

    echo 'B' ;
}
3、多态的简单应用
interface myusb{
    function type();
    function alert();
}
class zip implements myusb{
    function type() {
        echo"2.0";
    }
    function alert()
    {echo"正在检查———U 盘驱动";}
}
class mp3 implements myusb{
    function type() {
        echo"1.0";
    }
    function alert()
    {echo"正在检查———Mp3 盘驱动";}
}
class mypc{
    function pcusb($what)
    {
        $what->type()."<br>";
        $what->alert();
    }
}
$p=new mypc();
$zip=new zip();
$mp3=new mp3();
$p->pcusb($mp3); //插上不同的设备，就有不同的输出

```

总结篇

1、对象描述的配置

方法名 __toString()

我们可以直接打印对象句柄，从而获得该方法的基本信息或其他内容。

```

class My{
function __toString ( ){
    echo "我所要显示的内容";
    }
}
$p = new My();
echo $p;

```

输出结果：我所要显示的内容

2、对象方法的异常处理 方法名 __call(\$funname, \$arr_value)

调用一些不存在的对象方法的异常处理，使程序继续正常运行。

```
class My{
function __call($n,$v){
    echo "错误的方法名". $n;
    echo "错误的参数".Print_r($v);// 因为参数会有多个所以用数组的方式保存
    } //打印的时候也要就用 print_r 输出。
}
```

4、克隆对象的应用

方法名 __clone()

关键字 clone

通过克隆的方式我们可以在内存中生成两个一样的对象或升级原对象。

```
class My{
function __clone( ){
    echo "克隆时自动调用的函数";
    }
}
$a=new My()
$b=$a; //内存中只有一个对象
$a=new My()
$b= clone $a; //内存中会有两个对象
```

5、自动载入对象的方法

方法名 __autoload()

快速取得对象名称并自动载入进当前页面

function __autoload(\$class_n) { //该方法是在类以外单独的方法也就是我们所谓的构造函数。

```
include($class_n.".php"); //将会被自动调用，执行如下
}
    $p=new Mypc();
    $d=new demo();
    include("mypc.php");
    include("demo.php");
```

实例讲解

```
class my{
    public $name="我的类";
    function __toString()//反映出类里边的基本信息
    {
        return $this->name."这是一个空类";
    }
    function __call($n,$v)
    {
        echo "不存在的方法是: ".$n."<br>";
        echo "错误的值是: ".print_r($v);
    }
    function __destruct() {
        echo"<br>清理一个对象";
    }
}
```

```

    }
    function __clone()
    {
        $this->name="你的值";
    }
}
$p=new my();
//echo $p;//没有__toString方法之前看看结果
//$p->demo("demo",6);//系统会崩溃,引入__call($n,$v)方法
$b=clone $p;//区别$b= $p
echo $b->name;//值与 echo $p->name;不同
echo $p->name;

```

实例讲解

建立两个 php 文件,各自包含一个类

```

Function __autoload($name)
{
include( "$name.php" );
}

```

```

$d=new demo();
$d->fun1();
$t=new test();
$t->fun2();

```

制作自己的 PHP+MYSQL 的类

- 1、巩固学习常用 MYSQL 函数
- 2、使用类封装常用 SQL 操作,简化代码
- 3、认识面向对象在开发中的重要作用

用到的知识:

- 1、private 私有属性关键字
- 2、__construct 初始化方法名
- 3、常用 mysql 函数
- 4、常用 SQL 语句

实例开发:

```

class mysql{
    private $host;
    private $name;
    private $pass;
    private $table;
    private $ut;
    function __construct($host,$name,$pass,$table,$ut){
        $this->host=$host;
        $this->name=$name;
        $this->pass=$pass;
        $this->table=$table;
        $this->ut=$ut;
        $this->connect();
    }
    function connect(){

```

```

$link=mysql_connect($this->host,$this->name,$this->pass) or die
($this->error());
mysql_select_db($this->table,$link) or die("没该数据库：
".$this->table);
mysql_query("SET NAMES '$this->ut'");
}
function query($sql, $type = '') {
    if(!($query = mysql_query($sql))) $this->show('Say:', $sql);
    return $query;
}
function show($message = '', $sql = '') {
    if(!$sql) echo $message;
    else echo $message.'<br>'.$sql;
}
function affected_rows() {
    return mysql_affected_rows();
}
function result($query, $row) {
    return mysql_result($query, $row);
}
Function num_rows($query) {
    return @mysql_num_rows($query);
}
function num_fields($query) {
    return mysql_num_fields($query);
}
function free_result($query) {
    return mysql_free_result($query);
}
function insert_id() {
    return mysql_insert_id();
}
function fetch_row($query) {
    return mysql_fetch_row($query);
}
function version() {
    return mysql_get_server_info();
}
function close() {
    return mysql_close();
}
//=====
function fn_insert($table,$name,$value) {
    $this->query("insert into $table ($name) value ($value)");
}

```

```

}
$db = new mysql('localhost','root','','liuyanban','GBK');
$db->fn_insert('message','id,user,title,content,lastdate',' ','呵呵','
留言标题','今天心情不错',now());

```

作业、讨论题、思考题:

课后小结: 本节介绍了关键字 : final, self, static, const, interface, implements, instanceof 的应用。介绍了多态的介绍与优势、多态的简单应用。讲解了对象描述的配置, 对象方法的异常处理及克隆对象的应用、自动载入对象的方法。介绍了常用 MYSQL 函数、使用类封装常用 SQL 操作, 简化代码及认识面向对象在开发中的重要作用。

《ASP 动态网站开发》课程教案

授课班级		课次	7
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目 (教学章、节或主题): PHPMyAdmin 备份, 导入, 创建, 修改, 删除教程 PHP5 中 Cookie 与 Session 详解与 PHP5 中图片验证码的制作			
教学目的、要求 (含思政育人目标) (分掌握、熟悉、了解三个层次): 了解 PHPMyAdmin 简单配置和安装及 数据库的操作。 了解 Cookie 和 Session 简介与区别及 Cookie 与 Session 的配置与应用 熟悉启用 PHP 中 GD 库、部分 GD 库函数的介绍 掌握 GD+SESSION 制作 PHP 验证码及 image 与 header 输出的介绍 了解 imageline 与 imagesetpixel 函数、imaggottext 函数调用字体写入文字 掌握 PHP 验证码插入中文的方法 优秀学长工作经验分享, 树立正确的技能观, 努力提高自己的技能, 为社会和人民造福; 通过类和对象解决生活中的实际问题, 加深学生对专业知识技能学习的认可度与专注度。			
教学重点及难点:GD+SESSION 制作 PHP 验证码及 Cookie 与 Session 的配置与应用。			
教学基本内容及方法手段			

● 思政元素的融入（德育点）

- 1、优秀学长工作经验分享，树立正确的技能观，努力提高自己的技能，为社会和人民造福；
- 2、通过类和对象解决生活中的实际问题，加深学生对专业知识技能学习的认可度与专注度。
- 3、典型应用——党建平台，学习强国

信息技术，特别是软件技术在结合了互联网技术之后，将以传统方式开展的党建工作，升级为信息化、网络化、数据化的形式，软件技术能够助力新时代党建工作有力而高效地开展。

“学习强国”平台是由中共中央宣传部主管，以习近平新时代中国特色社会主义思想 and 党的十九大精神为主要内容，立足全体党员、面向全社会的优质平台，极大地满足了互联网条件下广大党员干部和人民群众多样化、自主化、便捷化的学习需求。

PHPMyAdmin 备份, 导入, 创建, 修改, 删除教程

1、PHPMyAdmin 简单配置和安装

- (1) 下载 PHPMyAdmin 压缩包，直接解压到网站或开发目录即可
- (2) 修改 PHPMyAdmin 配置文件

```
$cfg['PmaAbsoluteUri'] = '';
```

.....

```
$cfg['Servers'][$i]['host'] = 'localhost'; // MySQL hostname or IP address
```

```
$cfg['Servers'][$i]['port'] = ''; // MySQL port - leave blank for default port
```

.....

```
$cfg['Servers'][$i]['user'] = 'root'; // MySQL user
```

```
$cfg['Servers'][$i]['password'] = ''; // MySQL password (only needed
```

2、PHPMyAdmin 数据库的操作（十项功能）

1. 创建一个 mysql 数据库

创建数据库时选择好编码：gbk2312_chinese_ci

2. 创建一个 mysql 数据库表并添加字段

输入表名和表中的字段数量，注释的输入字段下面会出现注释的内容。

3. 原表中插入新字段

点击表结构，可以添加字段以及字段的位置。

4. 修改字段类型、功能、名称

选择字段的小铅笔图标进行修改。批量修改，先选择一些字段，在选择字段下面的小铅笔。

5. 修改表类型名称编码

选择表点击操作，可以修改表名及表名注释，可以复制表，更改排序。

6. 修改数据库基本信息和配置

选择数据库点击操作，可以修改数据库的基本信息。

7. PHPMyAdmin 一些常有选项卡的使用

浏览、结构、SQL、搜索（练习 SQL 语句、选择字段及操作员）、插入。

8. 导出几种类型的表或数据库

导出类型及另存为文件，数据库的导出结构必选，数据选择。压缩格式导出。

9. 导入表或数据库

导入方式直接将 SQL 语句直接复制上或者导入导出文件也可以。

10. 数据简单优化和一些其他操作

选择全部表点击优化表即可，将多余的垃圾删除掉。清空（数据库中的表恢复到建表的原始状态）、删除（删除一些多余的表）、打印预览、检查表、优化表、修复表、修复表、分析表。

PHP5 中 Cookie 与 Session 详解

1、Cookie 和 Session 简介与区别

在非常多时候，我们需要跟踪浏览者在整个网站的活动，对他们身份进行自动或半自动的识别（也就是平时常说的网站登陆之类的功能），这时候，我们常采用 Cookie 与 Session 来跟踪和判断

区别

Session 信息是存放在 server 端，但 session id 是存放在 client cookie 的，当然 php 的 session 存放方法是多样化的，这样就算禁用。

cookie 一样可以跟踪 Cookie 是完全保持在客户端的如：IE firefox 当客户端禁止 cookie 时将不能再使用。

2、Cookie 的配置与应用

Setcookie(string name, string value, int expire, string path, string domain, int secure);

其中 name 是 cookie 变量名称标识，你在 php 中将能象使用普通变量名相同来用他引用 cookie 变量。value 是 cookie 变量的初始值，expire 表示该 cookie 变量的有效时间；path 为该 cookie 变量的相关路径；domain 表示 cookie 变量的网站；secure 则需在 https 的安全传输时才有效。

SetCookie("Cookie" (名称, 必须), "cookievalue" (值, 必须), time()+3600 (毫秒 time()+3600=1 小时), "/forum" (保存路径), ".php100.com" (保存的域 HTTPS), 1);

接收和处理 Cookie

PHP 对 Cookie 的接收和处理的支持非常好，是完全自动的，跟 FORM 变量的原则一样，特别简单。

比如设置一个名为 MyCookie 的 Cookie, PHP 会自动从 WEB 服务器接收的 HTTP 头里把它分析出来，并形成一個与普通变量一样的变量，名为 \$myCookie，这个变量的值就是 Cookie 的值。数组同样适用。另外一个办法是引用 PHP 的全局变量 \$HTTP_COOKIE_VARS 数组。

分别举例如下：（假设这些都在以前的页面里设置过了，并且仍然有效）

```
echo $MyCookie;
echo $CookieArray[0];
echo $_COOKIE["MyCookie"];
echo $HTTP_COOKIE_VARS["MyCookie"];
```

删除 Cookie

要删除一个已经存在的 Cookie，有两个办法：

- 1、SetCookie("Cookie", "");
- 2、SetCookie("Cookie", "value", time()-1 / time());

使用 Cookie 的限制

- 1、必须在 HTML 文件的内容输出之前设置；
- 2、不同的浏览器对 Cookie 的处理不一致，且有时会出现错误的结果。
- 3、限制是在客户端的。一个浏览器能创建的 Cookie 数量最多为 30 个，并且每个不能超过 4KB，每个 WEB 站点能设置的 Cookie 总数不能超过 20 个。

3、Session 的配置与应用

```

session_start(); //初始化 session. 需在文件头部
$_SESSION[name]=value; //配置 session
echo $_SESSION[name]; //使用 session
isset($_SESSION[name]); //判断
unset($_SESSION[name]); //删除
session_destroy(); //消耗所有 session

```

4、用 Cookie 和 Session 做登陆

文件名: COOKIE.PHP 与 session.php

```

<?php
if(@$_POST['name']&&$_POST['password'])
{
setcookie('id',$_POST['name'],time()+3600);//没设置时间时, 关闭浏览器
cookie失效。
setcookie('pass',$_POST['password'],time()+3600);
echo"<script>location.href='session.php'</script>";
}
if(@$_COOKIE['id']&&$_COOKIE['pass'])
{
echo"登录成功<br>用户名: " . $_COOKIE['id'] . "<br>密码:" . $_COOKIE['pass'];
echo "<br><a href='session.php?out=out'>退出</a>";
}
if(@$_GET['out'])
{
setcookie('id','');
setcookie('pass','');
echo"<script>location.href='session.php'</script>";
}
?>
<form action="" method="post" >
  用户: <input type="text" name="name" /><br>
  密码: <input type="password" name="password" /><br>
  <input type="submit" value="登陆"/>
</form>

```

session.php

```

<?php
session_start();
if(@$_POST['name']&&$_POST['password'])
{
$_SESSION['id']=$_POST['name'];
$_SESSION['pass']=$_POST['password'];
}
if(@$_SESSION['id']&&$_SESSION['pass'])
{
echo"登录成功<br>用户名: " . $_SESSION['id'] . "<br>密
码:" . $_SESSION['pass'];

```

```

echo "<br><a href=' session. php?out=out' >退出</a>";
}
if(@$_GET[' out' ])
{
unset($_SESSION[' id' ]);
unset($_SESSION[' pass' ]);
}
?>
<form action="" method="post" >
  用户:  <input type="text" name="name" /><br>
  密码:  <input type="password" name="password" /><br>
  <input type="submit" value="登陆"/>
</form>

```

PHP5 中图片验证码的制作

1、启用 PHP 中 GD 库

GD 库提供了一系列用来处理图片的 API，使用 GD 库可以处理图片，或者生成图片。

在网站上 GD 库通常用来生成缩略图或者用来对图片加水印或者对网站数据生成报表。

- 通过 WAMPserver 启用 GD 库
- 通过 PHP. ini 启用 GD 库，将：extension=php_gd2. dll 去掉

2、部分 GD 库函数的介绍

1、imagecreatetruecolor 新建一个真彩色图像

imagecreatetruecolor (int x_size, int y_size) // x 就是宽 ， y 就是高

2、imagecolorallocate 为一幅图像分配颜色(调色板)

imagecolorallocate (resource image, int red, int green, int blue)// 三原色

3、imagestring 绘图函数

imagestring (resource image, font, int x, int y, 内容 , 颜色)

3、学习：随机函数，十六进制函数

1 、 rand 随机函数

rand ([int min, int max]) //rand (1,4) 随机 1-4 之间的数

2、dechex 十进制转换为十六进制

dechex (十进制数) //十六进制 1 ~ f

4、GD+SESSION 制作 PHP 验证码

生成随机数->创建图片->随机数写进图片->保存在 SESSION 中

che. php 文件

```

session_start();
//生成随机数
$rand='';
for($i=0;$i<4;$i++)
{
    $rand.=dechex(rand(1,15));
}
$_SESSION[' check_pic' ]=$rand;

```

```
//创建一个图片
$im=imagecreatetruecolor(100, 30);
//设置颜色
$bg=imagecolorallocate($im, 0, 0, 0); //设置背景颜色
$te=imagecolorallocate($im, 255, 0, 255); //设置字体颜色
//把字符串写在图像左上角
imagestring($im, rand(1, 6), rand(3, 70), rand(1, 20), $rand, $te);
//输出图像
header("content-type: image/jpeg");
imagejpeg($im); //将二进制编码转换成图片格式
```

submit.php 文件//验证码验证文件

```
<?php
session_start();
if($_POST['check'])
{
    if($_SESSION['check_pic']==$_POST['check'])
    {
        echo "验证码正确".$_SESSION['check_pic'];
    }else
    {
        echo "验证码错误".$_SESSION['check_pic'];
    }
}
?>
<form action="" method="post" >
<input type="text" name="check" /><br>
<input type="submit" value="提交" />
</form>
```

5、image 与 header 输出的介绍

PHP 的 HEADER 是定义头的动作，新 PHP5 中支持三种类型：

header (*) /*可以是以下三种格式

Content-Type: xxxx/yyyy

Location: xxxx/yyyy/zzzz

Status: nnn xxxxxx

内容文件的类型

如: image/gif image/jpeg image/png

GD 库中有对应的 image 类型

imagejpeg(*) imagegif(*) imagepng(*)

6、imageline 与 imagesetpixel 函数

imageline 画线函数

imageline (resource image, int x1, int y1, int x2, int y2, int color)

例子:

```
for($i=0;$i<3;$i++){
```

```
$te=imagecolorallocate($im,rand(0,255),rand(0,255),rand(0,255));
```

```

    imageline($im, 0, rand(0, 25), 100, 30, $te);
}
imagesetpixel 画点函数
imagesetpixel ( resource image, int x, int y, int color )

```

例子:

```

for($i=0;$i<200;$i++) {
    imagesetpixel($im, rand()%100, rand()%30, $te);
}

```

7、imagettftext 函数调用字体写入文字

```

imagettftext 带字体的写入函数
imagettftext ( resource image, float size, float angle, int x, int y, int
color, string fontfile, string text )
imagettftext($im, 12, 10, 20, 20, $te, 'SIMLI.ttf', $arr);

```

8、PHP 验证码插入中文的方法

iconv("gb2312", "UTF-8", "新年快乐!"); //首先要将文字转换成 UTF8 格式。然后使用 imagettftext 调用一种字体, 再将转换好的 UTF8 格式的中文写入图片中!

```

$arr=iconv("gbk", "utf-8", "新年快乐");
imagettftext($im, 12, 10, 20, 20, $te, 'SIMLI.ttf', $arr);

```

中文验证码例子

```

session_start();
//创建一个图片
$im=imagecreatetruecolor(100, 30);
//设置颜色
$bg=imagecolorallocate($im, 0, 0, 0); //设置背景颜色
$te=imagecolorallocate($im, 255, 0, 255); //设置字体颜色
//载入中文字体
$a=array(1=>"搭建好", 2=>"我爱你", 3=>"兴旺幸福", 4=>"我的幸福");
$arr=iconv("gbk", "utf-8", $a[rand(1, 4)]);
imagettftext($im, 12, 10, 20, 20, $te, 'SIMLI.ttf', $arr);
$_SESSION['check_pic']=$a[rand(1, 4)];
//输出图像
header("content-type: image/jpeg");
imagejpeg($im);

```

作业、讨论题、思考题: 制作一个中文验证码。

课后小结: 本节课介绍了 PHPMyAdmin 简单配置和安装与 PHPMyAdmin 数据库的操作(十项功能), Cookie 和 Session 简介与区别、Cookie 的配置与应用、Session 的配置与应用。

还讲解了用 Cookie 和 Session 做登陆, 启用 PHP 中 GD 库及部分 GD 库函数的介绍, 学习: 随机函数, 十六进制函数, GD+SESSION 制作 PHP 验证码, image 与 header 输出的介绍, imageline 与 imagesetpixel 函数, imagettftext 函数调用字体写入文字, PHP 验证码插入中文的方法。

《ASP 动态网站开发》课程教案

授课班级		课次	8
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目（教学章、节或主题）： PHP 中正则表达式学习及应用			
教学目的、要求（含思政育人目标）（分掌握、熟悉、了解三个层次）： 了解正则表达式的介绍和作用及 PHP 中两个常用的正则函数 掌握正则表达式中包括的元素、“原子”、“元字符”、“模式修正符”。 掌握正则表达式在 PHP 中的作用。 中国软件杯介绍； 国赛获奖者分享，提高学生自我学习和持续学习的意识和能力； 理解全局观的重要性，培养大局意识。以往届学生毕业设计的案例一引出大局观的重要性。			
教学重点及难点：正则表达式中的“元字符”的应用			
教学基本内容及方法手段			
<ul style="list-style-type: none"> ● 思政元素的融入（德育点） <ol style="list-style-type: none"> 1、中国软件杯介绍； 2、国赛获奖者分享，提高学生自我学习和持续学习的意识和能力； 3、理解思考问题的周详性，学会未雨绸缪的前瞻性，培养不拖沓的好性格。 ● 课程思政教学素材 <ol style="list-style-type: none"> 1、理解全局观的重要性，培养大局意识。以往届学生毕业设计的案例一引出大局观的重要性。 2、典型应用——电子商务，创新创业 <p style="margin-left: 20px;">网络电商模式方兴未艾，在层出不穷的新业态中，电商平台、大数据平台起到至关重要的作用，MySQL 技术在开发这些平台的过程中大有作为。引导学生提升专业认同感，稳定专业心态，矢志不渝地学习好本专业的知识，掌握好本专业的技能；作为未来的软件行业从业者，还应稳定职业心态，相信自己在软件行业领域内将有所作为、大有可为，从而在精进专业技术的道路上不断奋进。</p> <p>PHP 中正则表达式学习及应用</p> <p>1、正则表达式的介绍和作用</p> <p>什么是正则表达式</p> <p>在编写处理字符串的程序或网页时，经常会有查找符合某些复杂规则的字符串的需要。正则表达式就是用于描述这些规则的语法。</p> <p>例：在判断用户邮件地址格式、手机号码格式或者采集别人网页内容时</p> <p>主要的作用是：分割、匹配、查找、替换</p> <p>注：正则表达式对于一个程序员来讲是至关重要的一个知识点，所以学好正则 是每一个程序员必须具备的。不仅可以帮助我们完成一些通过函数无法实现的工作，还可以帮助我们减轻很多工作量。</p> <p>2、PHP 中两个常用的正则函数</p> <p>preg_match 正则函数，以 perl 语言为基础</p> <p>preg_match (mode, string subject , array matches)//一对模式匹配符</p>			

```
preg_match('/abc/', 'asdafadsfabcd', $arr);
print_r($arr);
ereg 正则函数，以 POSIX 基础 (Unix、Script)
ereg ( mode, string subject , array regs )
ereg('abc', 'asdafadsfabcd', $arr); //不需要模式定界符
print_r($arr);
```

3、正则表达式中包括的元素

- (1)、原子 (普通字符: a-z A-Z 0-9、原子表、转义字符)
- (2)、元字符 (有特殊功能的字符)
- (3)、模式修正符 (系统内置部分字符 i、m、S、U...)

4、正则表达式中的“原子”

```
$mode=' /add/';
$str="aaadddd";
echo"<hr>";
if(preg_match($mode, $str, $arr))
{
echo "匹配成功". $arr[0];
}else
{
echo "匹配不成功";
}
```

①a-z A-Z _ 0-9 //最常见的字符

②(abc) (skd) //用圆括号包含起来的单元符号，作为一个整体匹配，存入内存中。

③[abcs] [^abd] //用方括号包含的原子表，原子表中的^代表排除或相反内容

④转义字符

\d 包含所有数字[0-9]

\D 除所有数字外[^0-9]

\w 包含所有英文字符[a-zA-Z_0-9]

\W 除所有英文字符外[^a-zA-Z_0-9]

\s 包含空白区域如回车、换行、分页等 [\f\n\r]

5、正则表达式中的“元字符”

* 匹配前一个内容的 0 次 1 次或多次

. 匹配内容的 0 次 1 次或多次，但不包含回车换行

+ 匹配前一个内容的 1 次或多次

? 匹配前一个内容的 0 次或 1 次

| 选择匹配类似 PHP 中的 | (因为这个运算符是弱类型导致前面最为整体匹配)

^ 匹配字符串首部内容

\$ 匹配字符串尾部内容

\b 匹配单词边界，边界可以是空格或者特殊符合

\B 匹配除带单词边界意外内容

{m} 匹配前一个内容的重复次数为 m 次

{m,} 匹配前一个内容的重复次数大于等于 m 次

{m,n} 匹配前一个内容的重复次数 m 次到 n 次

() 合并整体匹配，并放入内存，可使用 \1 \2... 依次获取

```
// $mode=' /go(abc)g\\1le/'
```

6、正则表达式中的“模式修正符”

1、运算顺序 依然遵循从左到→右的运算规则

() 圆括号因为是内存处理所以最高

* ? + { } 重复匹配内容其次

^ \$ \b 边界处理第三

| 条件处理第四

最后按照运算顺序计算匹配

2、模式修正符

模式修正符是为正则表达式增强和补充的一个功能，使用在正则之外，/ 正则 / U (模式修正符)。

常用修正符：

i 正则内容在匹配时候不区分大小写（默认是区分的）

```
$mode=' /[a-z]/i';
```

```
$str="B";
```

m 在匹配首内容或者尾内容时候采用多行识别匹配

```
$mode=' /^aaaa/m';
```

```
$str="00000\naaaa0000";
```

```
$mode=' /aaa$/m';
```

```
$str="00000aaa\n0000";
```

s 将转义回车取消是为单行匹配如. 匹配的时候

```
$mode=' /444(.*)222/s';
```

x 忽略正则中的空白

```
$mode=' /3 3 3/x';
```

A 强制从头开始匹配

```
$mode=' /^444/mA';
```

```
$str="00000\n4440000\n222000";
```

D 强制\$匹配尾部无任何内容 \n

```
$mode=' /2000$/D';
```

```
$str="00000\n4440000\n222000\n";
```

U 禁止贪婪匹配 只跟踪到最近的一个匹配符并结束，常用在采集程序上的正则表达式.

```
$mode=' /<(.*?)>/U';
```

```
$str="asdddddas<b>ssss</b>fdasfds"; //查看源文件
```

7、正则表达式在 PHP 中的应用

1、匹配功能

preg_match_all 全部匹配函数

```
preg_match_all ( string pattern, string subject, array matches [, int flags] )
```

对结果排序使 \$matches[0] 为全部模式匹配的数组

用途：截取比较详细的内容，采集网页，分析文本

```
$mode=' /{(.*?)}/U';
```

```
$str="asdddddas {name} ssss {sex} fdas {age} fds";
```

```
echo"<hr>";
```

```
preg_match_all($mode, $str, $arr);
```

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

源文件如下

```
Array
(
    [0] => Array
        (
            [0] => {name}
            [1] => {sex}
            [2] => {age}
        )
    [1] => Array
        (
            [0] => name
            [1] => sex
            [2] => age
        )
)
```

2、替换功能

preg_replace 正则替换函数

```
preg_replace ( mixed pattern, mixed replacement, mixed subject [, int limit] )
```

通过正则表达式来替换相关内容，类似之前学过的 str_replace 字符串替换，但功能要强于它。

提示：1、替换内容可以是一个正则也可以是数组正则

```
$str="姓名： {name}<br>性别： {sex}<br>年龄： {age}";
```

```
$mode=array("/{name}/",("/{sex}/",("/{age}/");
```

```
$met=array("王京峰", "男", "33 岁");
```

```
echo"<hr>";
```

```
echo preg_replace($mode, $met, $str);
```

2、替换内容可以通过修正符 e 来解决替换执行内容

```
$str="2222a222333d2133124t6768768";
```

```
$mode=' /([a-z])/ie' ;//字符 e 增加之前后的变化
```

```
echo"<hr>";
```

```
echo @preg_replace($mode, "md5(\\1)", $str);
```

```
echo @preg_replace($mode, "md5(\\1)", $str, 1);
```

用途：替换一些比较复杂的内容上，也可以用于内容的转换上

3、分割功能

preg_split 正则切割

```
preg_split ( string pattern, string subject [, int limit [, int flags]] )
```

通过正则表达式来切割相关内容，类似之前学过的 explode 切割函数，但 explode 只能按照一种方式切割有局限性。

用途：太广

```
$str="Microsoft Office Word 2003 将使计算机上的文档阅读?工作变得前所未有的简单。现在 Word 可以根据屏幕的尺寸和分辨率优化显示。同时，一种新的阅读版式、视图也提高了文档可读性。";
```

```

$mode='/[.,、?]/';
echo"<hr>";
$arr=preg_split($mode,$str);
print_r($arr);

```

作业、讨论题、思考题： 练习使用“原子”与“元字符”书写正则表达式

课后小结： 本节课介绍了正则表达式的介绍和作用及 PHP 中两个常用的正则函数，正则表达式中包括的元素、正则表达式中的“原子”、正则表达式中的“元字符”、正则表达式中的“模式修正符”：1、运算顺序 2、模式修正符。正则表达式在 PHP 中的应用 1、匹配功能 、2、替换功能、3、分割功能。

《ASP 动态网站开发》课程教案

授课班级		课次	9
授课方式	理论课 <input checked="" type="checkbox"/> 讨论课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其他 <input type="checkbox"/>	课时安排	3
授课题目（教学章、节或主题）： PHP 模板引擎 Smarty			
教学目的、要求（含思政育人目标）（分掌握、熟悉、了解三个层次）： 了解 Smarty 介绍及应用的优缺点及 Smarty 的配置和应用。 了解什么是 Smarty 变量操作符及如何使用 Smarty 变量操作符 掌握用 foreach 数组的处理、include 多功能使用、IF 条件语句的使用。 掌握 Smarty 缓存的配置及 Smarty 缓存的使用和清除与 Smarty 局部缓存。 软件行业领军人物的奋斗故事分享，培养学生诚实、守信、坚忍不拔的性格； 通过完成一个简单的 MySQL 项目，提高学生在沟通表达、自我学习和团队协作方面的能力； 理解思考问题的周详性，学会未雨绸缪的前瞻性，培养不拖沓的好性格。			
教学重点及难点： Smarty 变量操作符与 Smarty 缓存的配置及 Smarty 缓存的使用和清除与 Smarty 局部缓存。			
教学基本内容及方法手段			

● 思政元素的融入（德育点）

- 1、软件行业领军人物的奋斗故事分享，培养学生诚实、守信、坚忍不拔的性格；
- 2、通过完成一个简单的 MySQL 项目，提高学生在沟通表达、自我学习和团队协作方面的能力。

● 课程思政教学素材

- 1、以实际工作中异常处理的案例—引出未雨绸缪的重要性，培养好性格。
- 2、典型应用——抗疫防疫，信息助力

新冠肺炎疫情来袭，全国人民乃至全球人类都要积极应对，做好防疫抗疫。在党中央的正确领导下，这场打响在中华大地上的防疫抗疫人民战争必定打赢。疫情期间，信息技术特别是基于软件技术设计开发的各类应用系统，乃至软件技术支持下的各类大数据系统、各类人工智能应用，为防疫抗疫带来巨大助力。为防止疫情在学校蔓延，延期开学是重要举措，停课不停教、不停学既是战疫情应急之举，也是互联网+ 教育的重要成果应用展示。信息技术将原本的线下教育教学活动搬到线上，彰显出技术的魅力。引导学生深入认识信息技术和信息技术行业的从业者在此次抗疫、防疫中的意义和贡献。

PHP 模板引擎 Smarty

1、Smarty 介绍及应用的优缺点

什么是 smarty?

Smarty 是一个使用 PHP 写出来的 PHP 模板引擎，目的是要使用 PHP 程序同美工分离，使的程序员改变程序的逻辑内容时不会影响到美工的页面设计，美工重新修改页面时不会影响到程序的程序逻辑，这在多人合作的项目中显的尤为重要。（也易于程序的多样式开发）

Smarty 优点

1. 速度快：相对其他模板引擎。
2. 编译型：采用 smarty 编写的程序在运行时要编译成一个非模板技术的 PHP 文件。
- 3 缓存技术：它可以将用户最终看到的 HTML 文件缓存成一个静态的 HTML 页。
4. 插件技术：smarty 可以自定义插件。

不适合使用 smarty 的地方

1. 需要实时更新的内容。例如像股票显示，它需要经常对数据进行更新。
2. 小项目。小项目因为项目简单而美工与程序员兼于一人的项目。

2、Smarty 的配置 //smarty_inc.php 配置文件

```
include_once("Smarty/Sm
$smarty = new Smarty(); //建立 smarty 实例对象$smarty
$smarty-> caching=false; //是否使用缓存，项目在调试期间，不建议启用缓存
$smarty->template_dir = "./templates"; //设置模板目录
$smarty->compile_dir = "./templates_c"; //设置编译目录
$smarty->cache_dir = "./smarty_cache"; //缓存文件夹
//-----
//左右边界符，默认为{}，但实际应用当中容易与 JavaScript 相冲突
//-----
$smarty->left_delimiter = "{";
$smarty->right_delimiter = "}";
```

3、Smarty 的应用：变量，循环…

```
$smarty->assign("模板变量", "值(数组/变量)");
```

```
$smarty->display("模板名称");
index.php 引入我们配置的 Smarty_inc.php
$smarty->assign("name", "PHP 学习制作网站"); //进行模板变量替换
$smarty->display("index.htm"); // 该文件就是模板文件，应该在模板目录下
index.htm
```

```
<html>
<title>{$name}</title>
</html>
```

例子: Index.php

```
include("smarty_inc.php");
$name="php 网站制作";
$smarty->assign("title", $name);
$smarty->display("index.htm");
```

Index.htm

```
<html>
<b><font color=red>{$title}</font></b>
</html>
```

```
$smarty->assign("模板变量", "数组");
```

数组是最常用的方式，可以帮助我们循环列表，和快速访问打理数据，如果要循环数组我们可以使用 Smarty 内置的方法 section

```
{section name=s loop=$stu}
{$stu[s].name}
{sectionelse}
无内容
{/section}
```

例子: index.php

```
include("smarty_inc.php");
$name[]=array("name"=>"新闻第一条", "date"=>"2010-09-08");
$name[]=array("name"=>"新闻第二条", "date"=>"2010-08-08");
$name[]=array("name"=>"新闻第三条", "date"=>"2010-10-08");
$name[]=array("name"=>"新闻第四条", "date"=>"2010-11-08");
$name[]=array("name"=>"新闻第五条", "date"=>"2010-12-08");
$row=array("标题", "作者", "当前页");
$smarty->assign("title", $name);
$smarty->assign("row", $row);
$smarty->display("index.htm");
```

index.htm

```
<html>
{$row[0]} | {$row[1]} | {$row[2]}
<hr>
{section name=list loop=$title}
<b><font color=red>
```

标题 | 作者 | 当前页

新闻第一条-2010-09-08
新闻第二条-2010-08-08
新闻第三条-2010-10-08
新闻第四条-2010-11-08
新闻第五条-2010-12-08

```
{ $title[list].name } - { $title[list].date }
</font></b><br>
{/section}
</html>
```

4、什么是 Smarty 变量操作符?

php 模板引擎 smarty 内置的一些操作函数，我们称之为**变量操作符**，变量操作符可用于操作变量，自定义函数和字符。(跟我 PHP 中常用的 PHP 内部函数类似)，他可以帮助我们完成很多比较实用的功能，如：**首字母大写，拆分，替换，截取...**，更像是我们提前写好的一些 PHP 常用类...

5、如何使用 Smarty 变量操作符

语法中使用“|”应用变量操作符，多个参数用“:”分隔开来

例子:

```
{ $name | truncate:10:"..." } //截取前十个字符多则用...代替
```

6、介绍常用的 20 个变量符

capitalize [首字母大写]

count_characters [计算字符数]

cat [连接字符串]

count_paragraphs [计算段落数]

count_sentences [计算句数]

count_words [计算词数]

date_format [时间格式]

```
{ $name | date_format:' %Y-%m-%d' } <br> //可以调用内置函数
```

```
等价于 { $smarty.now | date_format:' %Y-%m-%d' } <br>
```

default [默认]

escape [转码]

indent [缩进]

lower [小写]

nl2br [换行符替换成
]

regex_replace [正则替换]

replace [替换]

spacify [插空]

string_format [字符串格式化]

strip [去除(多余空格)]

strip_tags [去除 html 标签]

truncate [截取]

upper [大写]

wordwrap [行宽约束]

例子讲解

Index.htm

```
原始内容: { $name } <br><hr>
```

```
{ $name | capitalize } <br>
```

```
{ $name | count_characters } <br>
```

```
{ $name | cat:"smarty 变量演示" } <br>
```

```
{ $name | count_paragraphs } <br>
```

```
{ $name | count_sentences } <br>
```

```

{$name|count_words}<br>
{$name|default:"没有时间"}<br>
{$name|escape:'url'}<br>
{$name|indent:10:'&nbsp;'}<br>
{$name|lower}<br>
{$name|upper}<br>
{$name|replace:"is":"***"}<br>
{$name|spacify}<br>// {$name|spacify:"_"}<br>
$smarty->assign
{$name|string_format:"%.2f"}<br>// {$name|string_format:"%d"}<br>
{$name|strip:""}<br> // {$name|strip:"_"}<br>
{$name|strip_tag}<br>
{$name|truncate:30:"..."}<br>
{$name|wordwrap:10:"<br>"}<br>

```

Index.php

```

include("smarty_inc.php");
$value="it is work and,it is php study";
$smarty->assign("name",$value);
$smarty->display("index.htm");

```

7、foreach 数组的处理

与我们在 PHP 中的 foreach 用法类似，同样是用来遍历数组。在实际操作中与 section 功能一样，但 foreach 在 Smarty 一般用来处理一维数组。

=====无键值数组

```

{foreach from=$name item=id}
id: {$id}<br>
{/foreach}

```

例子: index.htm

```

{foreach from=$name item=id}
数组内容: {$id}<br>
{/foreach}

```

Index.php

```

include("smarty_inc.php");
$value=array(3,4,5,2,6,7,8,9,12,32);
$smarty->assign("name",$value);
$smarty->display("index.htm");

```

=====有键值数组

```

{foreach key=j item=v from=$name }//key 键值 item 内容 from 数组
{$j}: {$v}<br>
{/foreach}

```

例子: index.htm

```

{foreach from=$name item=id key=k}
数组内容: {$k}-{$id}<br>
{/foreach}

```

Index.php

```

include("smarty_inc.php");

```

```
$value=array('a'=>"PHP",'j'=>"JSP",'c'=>"c++",'s'=>"SQL");
$smarty->assign("name",$value);
$smarty->display("index.htm");
```

8、include 多功能使用

```
{include file="header.htm"}
{include file="D:\www\head.htm"}//php 中不可以访问绝对路径文件
{include file='head.htm' title="Main Menu"}
```

例子: head.htm

```
<title>{$title}</title>
```

这是一个导航文件!!!

Index.htm

```
{include file='head.htm' title='这是个首页'}
```

Index.php

```
include("smarty_inc.php");
$smarty->display("index.htm");
```

9、IF 条件语句的使用

```
{if $name=='ok'}
{else} //同样支持 elseif
{/if} //结束时不一样
```

例子: Index.htm

```
{if $name==''}
```

什么都没有!!!

```
{else}
```

里面有东西!!!

```
{/if}
```

Index.php

```
include("smarty_inc.php");
$value="sdfdsf";
$smarty->assign("name",$value);
$smarty->display("index.htm");
```

10、literal strip 文本的处理技巧

literal 数据将被当作文本处理,此时模板将忽略其内部的所有字符信息. 该特性用于显示有可能包含大括号等字符信息的 javascript 脚本

```
{literal}
<script language=javascript>//不建议大家把 javascript 写在页面上
```

.....

```
</script>
```

```
{/literal}
```

```
{literal}
```

```
<script language="JavaScript" >
```

```
function check()
```

```
{
    {
        alert("请填写用户名");
        return true; }
}
```

```

alert("测试");
</script>
{/literal}

```

strip 标记中数据的首尾空格和回车。这样可以保证模板容易理解且不用担心多余的空格导致问题。

一般放在 HTML 代码的开始和结束

```
{strip}
```

```
....
```

```
{/strip}
```

以下代码放在 {strip} {/strip} 与不放在其中的区别：查看源文件比较一下

```

<table width="200" border="1">
  <tr>
    <td colspan="2">this is a test ! </td>
  </tr>
  <tr>
    <td>welcome</td>
  </tr>
</table>

```

11、Smarty 缓存的配置

```

$smarty->cache_dir = "/caches/"; //缓存目录
$smarty->caching = true; //开启缓存,为 false 的时候缓存无效
$smarty->cache_lifetime = 60; //缓存时间
//配置好后会自动建立一个 caches 文件夹,查看缓存文件

```

12、Smarty 缓存的使用和清除

```

$smarty->display('cache.tpl', cache_id); //创建带 ID 的缓存
$smarty->clear_all_cache(); //清除所有缓存
$smarty->clear_cache('index.htm'); //清除 index.tpl 的缓存
$smarty->clear_cache('index.htm', cache_id); //清除指定 id 的缓存

```

例子: index.htm

```

{$id}
Index.php
include("smarty_inc.php");
@$id=$_GET['id'];
$smarty->assign("id",$id);
$smarty->display("index.htm",$id);
$smarty->clear_all_cache();//清除所有缓存
//$smarty->clear_cache('index.htm',$id); //清除指定 id 的缓存

```

13、Smarty 局部缓存

insert 函数默认是不缓存的。并且这个属性不能修改。

```

index.htm
{foreach from=$name item=id key=k}
数组内容: {$k}-{$id}<br>
{/foreach}
<div>{insert name="get_time"}</div>index.php
include("smarty_inc.php");

```

```

$value=array(' a' =>"PHP", ' j' =>"JSP", ' c' =>"cxzvcxz", ' S' =>"SQL");
$smarty->assign("name", $value);
$smarty->display("index.htm");
function insert_get_time() {
    return date("Y-m-d H:m:s");
}

```

smarty_block 函数功能更加强大，使用方法同上

```
{blockname}
```

```
没有缓存的: {$smarty.now}
```

```
{/blockname}
```

14、MYSQL 与 Smarty 的应用

通过连接一个数据库，具体学习 MYSQL 与 Smarty 的应用

例子 index.php

```

<?php
include("smarty_inc.php");
include("mysql_inc.php");
if(@$_GET[' id' ]){
$sql="SELECT * FROM message where id=".$_GET[' id' ];
$query=$db->query($sql);
$row=$db->fetch_row($query);
}
$smarty->assign("row", $row);
$smarty->display(' index.htm' );
?>
Index.htm
标题: {$row[1]}

```

作业、讨论题、思考题： smarty 模板的应用。

课后小结：本节课介绍了 Smarty 介绍及应用的优缺点，Smarty 的配置、Smarty 的应用：变量，循环…，什么是 Smarty 变量操作符？如何使用 Smarty 变量操作符，介绍常用的 20 个变量符。

讲解了 foreach 数组的处理，include 多功能使用，IF 条件语句的使用以及 literal strip 文本的处理技巧。

举例介绍了 Smarty 缓存的配置，Smarty 缓存的使用和清除，Smarty 局部缓存，MYSQL 与 Smarty 的应用。