



合肥信息技术职业学院
HEFEI INFORMATION TECHNOLOGY UNIVERSITY

教 案

课程名称: PHP 网站开发

授课教师: 孙 波

教务处制

二〇二二年九月

说 明

教案又称课时授课计划，是任课教师的教学实施方案。任课教师应遵循专业人才培养方案制订的培养目标，以教材为蓝本，在熟悉教材、了解学生的基础上，结合教学实际，提前编写设计好每堂课的全部教学活动。具体要求如下：

1. 每项内容都要认真填写，条理要清晰、内容要准确、格式要规范。

2. “教学课次”指本门课程第几次课（根据实际填1、2……）；“教学时数”指完成本次课所用教学时间，理论课通常以学时数为单位（一般2学时），实践课一般以学时数、天数或周数为单位，具体结合教学实际填写。

3. “教学过程设计”等内容要详细填写，各栏目表格不够可自行扩充，但要保持表格的完整性。

4. 教案包括封面、说明、首页、正文（根据需要自行复制），填写或手写前，应删去红色说明文字，填写字体格式为宋体、黑色；需使用A4纸双面打印（或手写），左侧装订。

5. 教案应根据专业技术领域发展、教学要求变化、学生实际水平，以及教师以往的教学后记、批注等进行补充、修改或重写，以保持教学内容的先进性、实用性、完整性和可操作性。

6. 教案须提前撰写教案，上课时应携带教案，学期结束时送交教务处存档；教研室要定期检查教师教案。

教案（首页）

课程名称	PHP网站开发
课程类型	专业技术课程
学时分配	共 56 学时，其中理论 30 学时，实践 26 学时。
使用教材	教材名称：PHP 动态网站开发项目实 主 编：林龙健 出 版 社：机械工业出版社 出版日期：2019 年 10 月
主要参考资料	教材名称：PHP+MySQL Web 项目实战 主 编：沙旭 出 版 社：北京希望电子出版社 出版日期：2020 年 6 月
首次授课时间	2023 年 2 月 9 日
课程教学目的与要求	<p>一、总体目标</p> <p>通过本课程的学习，使学生掌握 PHP 各方面的知识，掌握 Web 应用程序开发的特点和常用的实现方法，具备能够针对某一行业进行网站开发、对开源代码进行二次开发的能力，具备一定的独立网站编程能力。</p> <p>二、具体目标</p> <p>1. 知识目标</p> <p>（1）了解 PHP 的特征及功能，掌握 PHP 的基础知识和核心技术；</p> <p>（2）掌握 PHP 的安装与配置，掌握 PHP 的调试方法，熟悉 PHP 在整站程序中的作用；</p> <p>（3）掌握 PHP 进行 Web 开发的全过程；</p> <p>（4）通过对大量 PHP 开源实例中相关技术点的剖析讲解，使学员了解 PHP 各知识点在 Web 开发中的作用，掌握 PHP 编程技巧；</p> <p>（5）对实际的网页效果和网站应用程序分析，熟悉整个网站的设计思路和架构。</p> <p>2. 能力目标</p> <p>（1）能搭建 PHP 开发环境；</p> <p>（2）会开发简单的动态网页；</p> <p>（3）会对开源代码进行二次开发。</p>
备注	

教案（正文）

教学课次	1	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	动态网站概述				
教学目的 与要求	<p>教学目的：了解静态网页与动态网页的区别； 了解 PHP 的执行流程。</p> <p>教学要求：掌握动态网站运行原理。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：静态网页和动态网页的区别。</p> <p>教学难点：C/S 架构和 B/S 架构。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在之前的课程中我们已经学习过《网页设计与制作》相关课程，能够实现基本的网页布局等效果，大家回忆一下当时我们制作新闻类网站时新闻数据是如何显示在页面中的？当需要更新新闻时需要如何处理？</p> <p>进一步思考：大家在教务系统中进行技能考试报名时，报名数据是如何被记录的？查询期末考试成绩时成绩是从何处获取的？</p> <p style="padding-left: 2em;">（学生回答）</p> <p>老师：网页有静态网页和动态网页之分，要想实现与用户的交互功能则必须使用动态网页技术，本学期我们主要学习 PHP 动态网站开发技术。</p> <p>二、学习新知</p> <p>1.静态网页与动态网页的基本概念、优点及应用场景</p> <p>讲解：静态网页通常是指使用 HTML、CSS、JavaScript、jQuery 等技术编写的网页，扩展名为.html 或.htm。</p> <p>静态网页适用于更新频率较低的宣传、展示性场景中，主要用于一些固定内容的展示。</p> <p>静态网页的“静”主要指不同的用户在不同的时间访问网页，网页的内容都是固定不变的。</p> <p>静态网页的缺点在于网页内容更新不方便和无法与用户实现交互，但也正是这样的特点使其具有响应速度快、安全性能高、可跨平台等优势。</p>				

动态网页中除了包含静态网页的内容之外还包含一些实现特定功能的程序代码，这些程序代码可以实现用户与服务端的交互，可以针对不同用户、不同请求动态生成并显示不同的网页内容。

动态网页的优势主要体现在：交互性、自动更新、可实现个性化服务

2.静态网页和动态网页的区别：

比较项目	静态网页	动态网页
使用技术	HTML、CSS、JavaScript、jQuery	在静态网页基础上加入 ASP、ASP.Net、PHP、JSP 等动态网页技术
文件类型	.html 或 .htm	.asp、.aspx、.jsp、.php 等
更新维护	需修改源代码，工作量大	数据库做支持，无需修改页面，更新维护方便
交互性	交互性差，用户被动接受	可实现更丰富的功能，在用户和网站建设者之间实现信息双向交流
响应速度	响应速度快	需在服务器端执行相应程序，进行数据库等操作，响应速度慢

3. 客户端动态技术和服务器端动态技术

讲解：客户端动态技术不需要与服务器进行交互，实现动态功能的代码往往采用脚本语言形式直接嵌入到网页中。服务器发送给浏览者后，网页在客户端浏览器上直接响应用户的动作，有些应用还需要浏览器安装组件支持。常见的客户端动态技术包括 JavaScript、VBScript、Java Applet、Flash、DHTML 和 ActiveX 等。

服务器端动态技术需要与客户端共同参与，客户通过浏览器发出页面请求后，服务器根据 URL 携带的参数运行服务器端程序，产生的结果页面再返回客户端。一般涉及数据库操作的网页（如注册、登录和查询等）都需要服务器端动态技术程序。动态网页比较注重交互性，即网页会根据客户的要求和选择而动态改变和响应，将浏览器作为客户端界面，这将是今后 Web 发展的趋势。动态网站上主要是一些页面布局，网页的内容大都存储在数据库中，并可以利用一定的技术使动态网页内容生成静态网页内容，方便网站的优化。典型的服务器动态技术有 CGI、ASP/ASP.Net、JSP、PHP 等。

4.PHP 概述

讲解：PHP：“超文本预处理器”，是一种通用、开源、HTML

内嵌式脚本语言。其语法混合了 C 语言、Perl 语言的特点，它通过将程序嵌入到 HTML 文档中去执行，执行速度比 CGI 或者 Perl 更快，具有速度快、开源免费、跨平台、支持面向对象、支持多种数据库等优势。

5.PHP 特点？

讲解：PHP 作为一种服务器端的脚本语言，它的特点主要有以下 8 个方面。

- (1)开放源代码
- (2)基于服务端
- (3)数据库支持
- (4)跨平台
- (5)易于学习
- (6)网络应用
- (7)安全性
- (8)其他特性

6.PHP 系统构成

一个完整的 PHP 系统由以下几个部分构成：

(1)操作系统

网站运行服务器所使用的操作系统，允许 PHP 运行在任何操作系统上。

(2)服务器

PHP 支持多种服务器软件，包括 Apache、IIS 等。

(3)PHP 包

PHP 包实现对 PHP 文件的解析和编译。

(4)数据库系统

实现系统中数据的存储。PHP 支持多种数据库系统。

(5)浏览器

PHP 对浏览器没有任何限制。

7.工作流程

在图 1-1 中完整地展示了用户通过浏览器访问 PHP 网站系统的全过程，从图中可以更加清晰的理清它们之间的关系。PHP 程序的工作流程如下。

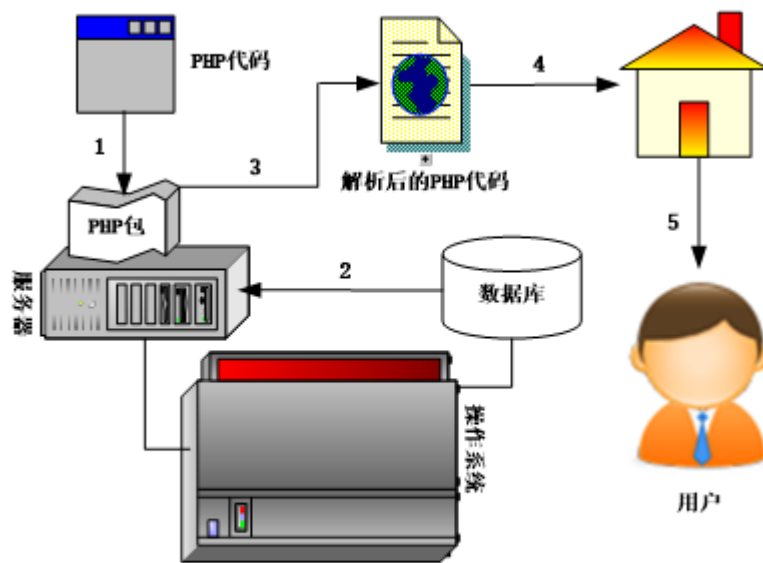
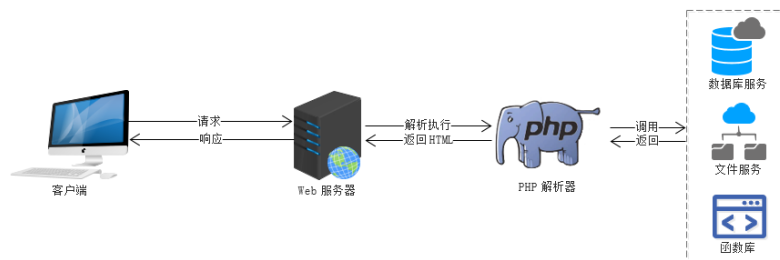


图 1-1 PHP 程序的工作流程

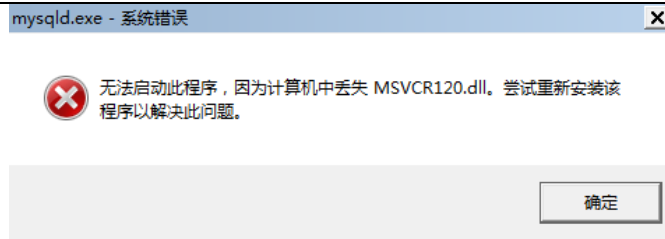
一个 PHP 程序的执行需要浏览器、Web 服务器、PHP 解析器、数据库服务器等共同相互协调完成。



作业/思考题： 理解静态网页和动态网页区别。

教学后记：通过讲解网页类型，区分两种网页。让学生了解动态网页优点和方便，从而让学生对 PHP 产生兴趣。

教学课次	2	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	PHP 开发环境搭建				
教学目的 与要求	<p>教学目的：学会 PHPStudy 安装与使用； 了解 WampServer 使用。</p> <p>教学要求：掌握 PHP 运行环境的搭建与配置； 掌握 PHPStorm 的安装以及断点调试方法。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：PHP 运行环境的搭建与配置；</p> <p>教学难点：PHPStorm 的安装以及断点调试方法。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在之前的课程中我们已经学习过《网页设计与制作》相关课程，能够实现基本的网页布局等效果，大家回忆一下当时我们制作网站时如果要运行网站是否需要安装环境？</p> <p>进一步思考：之前我们所做的网站是静态网站，现在我们要做的是动态网站，动态网站要不要安装运行环境呢？</p> <p>（学生回答）</p> <p>老师：动态网站语言是服务器端语言，我们需要安装运行环境后才能正常运行。这节课我们就来学习安装 PHP 运行环境。</p> <p>二、学习新知</p> <p>1.使用 phpStudy 搭建 PHP 环境</p> <p>讲解：phpStudy 是一款方便易用的 PHP 调试环境程序集成包，当前最新版本是 8.1，支持 Windows 和 Linux 系统，可以实现一键创建网站、FTP、数据库、SSL 等功能，同时支持安全管理、计划任务、文件管理、PHP 多版本共存及切换。</p> <p>演示：PHPStudy 的下载、安装、启动。</p> <p>强调：如果在启动服务的过程中出现下图所示的错误提示，则是由于缺少相关的 VC 运行库而引起的，根据当前操作系统类型、PHP 的版本、Apache 的版本下载安装对应版本的 VC 运行库即可。</p>				



讲解：PHPStudy 的目录结构。

演示：创建本地网站、添加本地域名解析、测试网站

2. 了解 wampserver 搭建 PHP 环境

讲解：WampServer 是一款由法国人开发的 Apache Web 服务器、PHP 解释器以及 MySQL 数据库的整合软件包。免去了开发人员将时间花费在繁琐的配置环境过程，从而腾出更多精力去做开发。

演示：安装 wampserver 的过程很简单，只要一直点击 next 就可以完成安装了：依次按图操作就行了。

2.1 MySQL 以及 PhpMyAdmin 的配置（可以不改密码）

单独安装过 MySQL 的朋友们肯定会发现一个问题，以前安装 MySQL 的时候是需要配置一个 root 的密码的，可是安装 wampserver 的时候，至始至终都没有配置密码的步骤，那岂不是 MySQL 为空密码了？答案是肯定的，确实是空密码。这样岂不是很不安全？非常对，所以我们要修改密码。

2.2 Apache 的配置

虽说 wampserver 已经帮我们集成好了 Apache，但是有些细节的东西还是需要我们自己去完善的，下面就让我们来一一对 wampserver 自带的 Apache 进行完善。

刚装完 wampserver 之后，有没有性子比较急的朋友们可能会尝试通过外网访问我们配置好的 wampserver 服务器，但是会发现通过外网访问会提示权限不够，这是为什么呢？是因为 wampserver 默认是只允许 127.0.0.1 访问的，也就是只允许本机访问，怎么修改呢？依次点击—Apache—httpd.conf，找到如图所示的地方，大概在第 234 行，把“Deny from all”删掉，再把“Allow from 127.0.0.1”改成“Allow from all”，就可以了。

2.3 配置 PHP

PHP 的配置比较简单，只需要进行一些基本设置的修改就可以了，依次点击 —PHP—php.ini，找到这三个地方：

`short_open_tag = Off`（是否允许使用 PHP 代码开始标志的缩写形式（`<? ?>`）。）；

`memory_limit = 128M`（最大使用内存的大小）；

upload_max_filesize = 2M (上传附件的最大值),

第一个需要改成 on, 否则的话很多 php 程序都会运行不了, 后面的两项就按照大家的实际需求更改就行了。

强调: 可能遇到的问题: Apache 服务器默认的 80 端口被 IIS 占用, 导致 wamp 运行失败。

解决的办法:

(1)、找到 wamp 的安装目录—>bins 目录—>Apache 安装目录—>conf 目录下的 httpd.conf 文件 (即 d:\Apache\conf\httpd.conf)。打开它。

(2) .找到 “Listen 80”, 紧接着 Listen 的数字就是端口号, 我们改为 “Listen 8080”。 修改为其它未使用的端口号也行。

(3) .重新启动 Apache, 使新的配置生效。

3.手动安装配置 PHP 环境

讲解: 使用 phpStudy 软件可以快速搭建 PHP 运行环境, 但手动安装 PHP 环境可以个性化配置 PHP 运行环境, 同时有助于更深入的理解 PHP 内部结构和运行机制。手动需要安装配置 PHP 和 Apache。

演示: (1) PHP 的安装与配置 (获取、解压、配置);

(2) Apache 的安装与配置 (获取、解压、配置、安装、启动)

4.PhpStorm 的安装与使用

讲解: PhpStorm 的简单介绍及优点。

演示: PHPStorm 的安装

5.Xdebug 断点调试

讲解: Xdebug 是一个开放源代码的 PHP 程序调试器, 可以用来跟踪、调试和分析 PHP 程序的运行状况。

演示: (1) Xdebug 的下载与配置

(2) 程序调试的过程

三、课堂小结, 布置作业

对这节的主要内容进行总结, 布置作业。最后留几分钟让学生和老师或者同学之间互相讨论, 加深课程内容印象, 及时提出问题、解决问题。

作业/思考题: 课后习题 1、2、3。

教学后记：主要讲解 PHPStudy 集成环境使用，让学生感受安装 PHP 环境非常简单。顺便了解 WampServer。

教学课次	3	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	PHP 基本语法				
教学目的 与要求	<p>教学目的：掌握 PHP 标记、注释、输出语句； 掌握 PHP 数据类型、标识符、关键字、常量、变量、运算符。 教学要求：掌握并使用基本语法。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：常量、变量、运算符。 教学难点：连接运算符、逻辑运算符。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：PHP 是用于开发动态网页的脚本语言，动态网页和静态网页有什么区别？</p> <p style="padding-left: 2em;">(学生回答)</p> <p>老师提问：PHP 执行的流程是什么？</p> <p style="padding-left: 2em;">(学生回答)</p> <p>老师：如果把掌握一门计算机语言比作修建一座宏伟的大厦，那么语法基础就像是大厦的地基，是掌握好这门语言的前提，PHP 语言也不例外。要想熟练运用 PHP 进行网站开发，必须掌握 PHP 基础语法。本章主要对 PHP 常用语法基础进行讲解，包括：PHP 的标记风格、常量、变量、运算符、表达式、流程控制结构等。</p> <p>二、学习新知</p> <p>1. PHP 标记</p> <p>讲解：PHP 在解析代码时，通过寻找开始和结束标记来确定代码的解析范围，PHP 脚本可以放置在文档中的任何位置，通过特定的标记将 PHP 代码和其他代码进行区分。PHP7 支持以下两种标记风格：XML 风格和简短风格。</p> <p>(1) XML 风格</p> <pre style="padding-left: 2em;"><?php //PHP 代码; ?></pre> <p>(2) 简短风格</p> <pre style="padding-left: 2em;"><? //PHP 代码;</pre>				

?>

2. PHP 注释

讲解：注释是对代码的解释和说明文字，一般出现在代码的上方。注释主要对代码的功能、创建者、修改者、时间等内容进行说明。合理利用注释可以提高代码的可读性。

PHP 注释可以分为单行注释、多行注释和#号注释。

单行注释使用//进行注释。

多行注释使用/*...*/进行注释。

#号注释使用#号进行注释。

演示：分别演示三种注释方式，并提示学生们注意相关注意事项。

3. PHP 输出语句

讲解：PHP 输出语句可以将脚本内容显示在浏览器上，方便用户查看程序的执行结果。PHP 常用输出语句包括：echo 语句和 print 语句等。

演示：echo 语句、print 语句的用法。

注意：echo 语句在输出多个字符串时，字符串可以作为多个参数单独传递，也可以连接在一起并作为单个参数传递。作为多个参数传递时，参数之间用逗号分隔。

print 语句每次只能输出一个字符串或者变量的值，且总是返回 1。

4. 数据类型

讲解：在 PHP 程序中支持 9 种数据类型，具体又可细分为：

1) 4 种标量类型：boolean（布尔型）、integer（整型）、float/double（浮点型）、string（字符串）。

2) 3 种复合类型：array（数组）、object（对象）、callable（可调用）。

3) 2 种特殊类型：resource（资源）、NULL（空类型）。PHP 数据类型划分如图 2-1 所示。

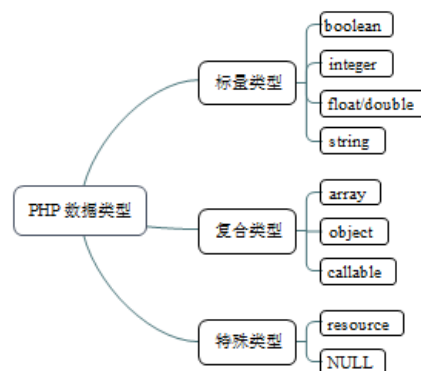


图 2-1 PHP 数据类型

演示：结合代码演示不同数据类型的用法，重点演示布尔型、整型、浮点型和字符串类型。

5. 标识符、关键字

讲解：PHP 中标识符的定义需要遵循以下规则：

- 1) 标识符只能由字母、数字、下划线组成，且不能以数字开头。
- 2) 标识符用作变量、常量或数组的名称时，区分大小写。
- 3) 用作函数名、类名时不区分大小写。

关键字是指 PHP 中一些带有特殊含义的名称，它们是语言结构的一部分。关键字不可以作为常量名、类名或方法名；关键字可以作为变量名使用，但为了避免引起歧义，不推荐此种使用。

6. 常量、变量

讲解：PHP 使用常量来表示那些在程序执行过程中始终保持不变的数据。一个常量一旦被定义，就不能再改变或者取消定义。常量一般采用全大写形式表示，如果常量名称由多个单词构成，采用下划线对单词进行分隔。

演示：结合例 2-1 和例 2-2 讲解常量的使用。

讲解：PHP 用变量保存那些在程序执行过程中可能发生变化的数据。为了便于区分每个变量，开发者可以给每个变量起一个简洁明了、容易记住的名字，也就是“变量名”。

PHP 通过“\$”符号和一个变量名表示变量，变量名遵循上文标识符的命名规则：由数字、字母、下划线组成，且不能以数字开头。此外，PHP 变量名一般采用小写字母，且区分大小写。例如：`$name`、`$_book` 都是合法的变量名称。

演示：结合例 2-3~例 2-5 讲解变量的使用。

7. 运算符

讲解：PHP 运算符提供对操作数的运算方式，它可以对一个及以上操作数进行运算，根据运算数的个数可分为一元运算符、二元运算符和三元运算符。

演示：结合例 2-6~例 2-9 演示字符串运算符、算术运算符、赋值运算符、递增/递减运算符、比较运算符、条件运算符、逻辑运算符、位运算符、错误控制运算符的用法，并比较运算符的优先级。

强调：PHP7 新增的太空船运算符、NULL 合并运算符，错误控制运算符。

8. 类型转换

讲解：不同类型数据之间的自动类型转换和强制类型转换。

演示：结合代码演示不同类型之间的转换。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 课后习题 1、2、3。

教学后记： 主要讲解 PHP 基本语法，让学生体验和 C 语言书写区别。

教学课次	4	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	PHP 控制语句				
教学目的 与要求	教学目的：掌握选择结构的用法。 教学要求：掌握循环结构的用法。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：选择结构、循环结构。 教学难点：循环结构。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：现在大家已经习惯了网上购物，那么大家有没有留意过自己在购物网站的信用等级呢？网站一般会对会员购物实行评分累积，形成不同的信用度。以淘宝为例，用户每在淘宝网上购物一次，至少可以获得一次评分的机会，具体为：“好评”加一分，“中评”零分，“差评”扣一分。如果让我们用计算机程序来写一个判断用户信用等级的程序，应该怎么样来完成呢？</p> <p style="text-align: center;">（学生回答）</p> <p>老师：要想完成这样一个程序，就需要用到我们今天要学习的流程控制语句。流程控制语句用于实现对程序流程的选择、循环、跳转等，在编码时，只有清楚每条语句的执行流程，才能选择合适的流程控制语句来实现想要的功能。合理的流程控制结构能够使程序代码更加清晰、减少代码冗余，有利于提高开发效率。</p> <p>二、学习新知</p> <p>1. 选择结构</p> <p>讲解：选择结构根据选择条件的不同，执行不同的分支语句，从而得到不同的结果。例如：如果学生成绩大于 60，则该学生的成绩及格；否则，成绩不及格。如果淘宝用户的积分在 4~250 之间，信用等级为“红心”；在 250~10000 之间，信用等级为“蓝钻”等。</p> <p>常用的选择结构包括 if 条件语句和 switch 条件语句两种。</p> <p>演示：结合例 2-10、2-11、2-12 讲解 if 语句的用法。</p> <p style="padding-left: 40px;">结合例 2-13 讲解 switch 语句的用法。</p>				

2. 循环结构

讲解：对于一些需要反复执行、且有规律的代码，可以采用循环结构进行编写。循环结构能够使代码结构更加清晰，有效减少重复代码。

循环结构分为：`for` 循环结构、`while` 循环结构 `do...while` 循环三种。

演示：结合例 2-14、2-15、2-16 讲解 `for` 循环的应用。

结合例 2-17 讲解 `while` 循环的用法。

结合例 2-18 讲解 `do...while` 循环的用法。

结合例 2-19 讲解 `while` 循环和 `do...while` 循环的区别。

3. 循环跳出语句

讲解：只要循环条件成立，循环语句便会一直执行下去。如果希望在循环过程中跳出循环，可以采用循环跳出语句。PHP 循环跳出语句包括 `break` 和 `continue` 两种。

`break` 语句可以直接跳出 `for`、`while` 和 `do...while` 循环，需要特别注意的是：当有多层循环嵌套时，`break` 语句只能跳出离得最近的一层循环。

`continue` 语句只能跳出本次循环，并继续进入下一次循环。

演示：结合例 2-20 演示 `break` 语句的用法。

结合例 2-21 演示 `continue` 语句的用法。

4. 案例：打印月历

讲解并演示：首先讲解月历的用处，呈现最终效果。然后，对案例进行分析，总结实现本效果需要的步骤。最后，按照步骤通过代码进行实现。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 1、用“*”输出金字塔形状。2、猜数字游戏。3、取钱币。

教学后记：主要讲解选择结构和循环结构，重点要让学生搞清楚循环嵌套。

教学课次	5	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	数组基本概念				
教学目的 与要求	<p>教学目的：掌握数组的概念和分类。</p> <p>教学要求：掌握定义数组的方法；掌握访问、遍历和删除数组的方法。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：数组的定义和访问。</p> <p>教学难点：遍历数组。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在程序中经常会对一批数据进行操作，比如，在微信运动中，对每天好友的运动步数进行排序，我们如何存储每个好友的步数呢？</p> <p>(学生回答)</p> <p>老师：如果用整型变量来表示每位好友每天的运动步数，有多少个好友就需要定义多少个变量，这样做不仅麻烦，而且容易出错。这时，可以使用数组来解决，从而引入数组的概念。</p> <p>二、学习新知</p> <p>1. 数组的概念和分类</p> <p>讲解：数组是一组有某种共同特性的元素组成的集合，相当于存储多个元素的容器。在 PHP 中，每个元素都包括键（key）和值（value）两个项，每个元素是一个“键值对”（key=>value），键值是成对出现的，是一一对应的关系。其中“键”为元素的识别名称，也被称为数组的下标，可以是数字、字符串或者数字与字符串的组合；而“值”为元素的值，可以定义为任意类型。在 PHP 中，根据维度可以将数组分成一维数组、二维数组和多维数组。根据下标的数据类型，可以将数组分为索引数组和关联数组。</p> <p>强调索引数组和关联数组的存储方式和区别。</p> <p>2. 定义数组</p> <p>讲解：讲解使用 [] 定义数组、使用 array() 定义数组、使用短数组语法定义数组，以及定义二维数组的方法。</p> <p>演示：分别演示不同方法定义数组的语法格式，并提示学生们注意定义二维数组时的注意事项。</p>				

3.访问、输出数组

讲解：访问数组的方式：\$数组名[键名],并演示访问数组元素的方法。

讲解：输出数组的两个函数，分别是 print_r()和 var_dump()。

强调：print_r()和 var_dump()的区别。var_dump()函数不仅输出元素的值，而且输出元素的数据类型；print_r()函数只输出元素的值。

4.遍历数组

讲解：遍历数组的两种方式。

第一种：

```
foreach (array_expression as $value)
{
    statement
}
```

第二种：

```
foreach (array_expression as $key => $value)
{
    statement
}
```

通过例子讲解两种方式遍历数组的区别，在举例演示遍历二维数组的方法。

```
<?php
$students = array(
    '科学 1 班' => array('张亮', '李丽'),
    '科学 2 班' => array('李风', '张森'),
    '科学 3 班' => array('李艳丽', '马良龙')
); //定义二维数组$students
foreach ($students as $key => $value) //循环读二维数组，$value 仍为数组
{
    echo $key . ' ' . $value; //输出第一维度的键名
    foreach ($value as $k => $v) //遍历一维数组$value 中的数据
    {
        echo $k . ' => ' . $v; //输出一维数组的键名和值
        echo ' ' . $v; //输出空格
    }
    echo '<br>'; //输出换行
```

}?>

5.删除数组

讲解：删除数组元素的方法和删除整个数组的方法，并举例演示效果。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 构建数组并完成课后习题。

教学后记：不宜讲过于复杂，不然学生不易搞懂。主要讲下一维数组和几个数组函数即可。

教学课次	6	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	数组例题				
教学目的 与要求	<p>教学目的：掌握常用数组函数的使用。</p> <p>教学要求：掌握超全局数组变量的使用；通过案例掌握综合使用数组解决问题的方法。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：数组函数的使用。</p> <p>教学难点：综合运用数组解决问题的方法。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：对于大量的数组元素，是否有一些函数可以对其进行操作呢？</p> <p>(学生回答)</p> <p>老师：是的，PHP 提供了丰富且功能强大的数组函数，包括统计函数、数组指针函数、检索函数、排序函数、数组元素操作函数等等，从而引入数组函数的使用。</p> <p>二、学习新知</p> <p>1. 常用数组函数</p> <p>(1) 统计函数</p> <p>讲解：讲解统计函数的功能。</p> <p>演示：举例并演示 count()、max()、min()、array_sum()、array_product()、array_count_values()这些函数的使用。</p> <p>(2) 数组指针函数</p> <p>讲解：PHP 内置了管理数组“当前指针”的函数，使用这些函数可以移动数组“当前指针”并读取数组元素，从而实现数组的遍历，讲解数组指针函数的功能。</p> <p>演示：举例并演示 current()、key()、next()、prev()、end()、reset()这些函数的使用。</p> <p>(3) 检索函数</p> <p>讲解：PHP 提供的数组检索函数主要是对数组元素的“键”或“值”进行查询，主要讲解检索函数的使用。</p> <p>演示：举例并演示 array_keys()、array_values()、in_array()、array_search()、array_key_exists()、array_unique()这些函数的使用。</p>				

(4) 排序函数

讲解：在操作数组时，经常对数组中的元素进行排序，讲解多种数组排序函数。

演示：举例并演示 sort()、asort()、rsort()、arsort()、ksort()、krsort()、shuffle()、array_reverse()这些函数的使用。

(5) 数组元素操作函数

讲解：数组元素操作函数的使用。

演示：举例并演示 array_push()、array_pop()、array_unshift()、array_shift()这些函数的使用。

(6) 其他函数

讲解：其他函数的使用。

演示：举例并演示 is_array()、array_merge()、array_rand()、range()这些函数的使用。

2.超全局数组变量

讲解：PHP 提供的预定义数组变量包含了来自 Web 服务器、客户端、运行环境和用户输入的数据，这些数组非常特别，通常被称为自动全局变量或者超全局变量。重点讲解\$_GET、\$_POST、\$_REQUEST、\$_SERVER、\$_ENV、\$_COOKIE、\$_SESSION、\$GLOBALS 中的\$_SERVER，其他的将在后续章节讲解。

举例：

```
<?php
header('content-type:text/html;charset=utf-8');
echo '当前服务器的主机名是：' .
    $_SERVER['SERVER_NAME'];
echo '<br/>';
echo '客户端连接到主机所使用的端口是：' .
    $_SERVER['SERVER_PORT'];
echo '<br/>';
echo '脚本文件的名称是：' .
    $_SERVER['SCRIPT_FILENAME'];
echo '<br/>';
echo '当前程序相对路径是：' . $_SERVER['PHP_SELF'];
echo '<br/>';
echo '文件存在的位置：' .
    $_SERVER['DOCUMENT_ROOT'];
echo '<br/>';
echo '访问页面时的请求方法：' .
    $_SERVER['REQUEST_METHOD'];
?>
```

3.案例：统计学习时长

- (1) 案例呈现：呈现本节案例
- (2) 案例分析：分析案例所需要的知识点
- (3) 案例实现：讲解案例实现的过程。

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Title</title>
  <style>
    /*CSS 代码在此省略，完整代码请参考配套源代码*/
  </style>
</head>
<body>
<table width="600" height="100">
  <th colspan="6">学习时长统计</th>
  <tr>
    <td>学号</td>
    <td>姓名</td>
    <td>班级</td>
    <td>在线学习时间(h)</td>
    <td>线下学习时间(h)</td>
    <td>学习总时间(h)</td>
  </tr>
  <!-- 定义学生数组-->
  <?php
    $student = array(
      array('id' => '20081031', 'name' => '杜云', 'class' => '
科学 1 班', 'Onlinetime' => 30, 'offlinetime' => 26),
      array('id' => '20081032', 'name' => '王亮', 'class' => '
科学 1 班', 'Onlinetime' => 28, 'offlinetime' => 30),
      array('id' => '20081033', 'name' => '张凤', 'class' => '
科学 1 班', 'Onlinetime' => 26, 'offlinetime' => 32),
      array('id' => '20081231', 'name' => '王良', 'class' => '
科学 2 班', 'Onlinetime' => 36, 'offlinetime' => 20),
      array('id' => '20081232', 'name' => '李军', 'class' => '
科学 2 班', 'Onlinetime' => 45, 'offlinetime' => 16),
    );
    $sum = 0; //统计学习总时间的
              变量，初始值为 0
    foreach ($student as $value) { //遍历数组中的每
                                  一个元素
```

```

?>
<tr>
    <td><?php echo $value['id'] ?></td>
    <td><?php echo $value['name'] ?></td>
    <td><?php echo $value['class'] ?></td>
    <td><?php echo $value['Onlinetime'] ?></td>
    <td><?php echo $value['offlinetime'] ?></td>
    <td><?php $everytime = $value['Onlinetime']
+ $value['offlinetime'];
        echo $everytime; ?></td>
    <?php
        $sum = $sum + $everytime;           //累
        加求和，统计所有学生的总学时
    ?>
</tr>
<?php } ?>
<tr>
    <td colspan="6"> 合 计 :<?php echo
count($student) ?>人,总学时:<?php echo $sum; ?>小时
</td>
</tr>
</table>
</body>
</html>

```

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 课后习题 1、2。

教学后记：不宜讲过于复杂，不然学生不易搞懂。主要讲下一维数组和几个常用数组函数即可。

教学课次	7	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	函数概念				
教学目的 与要求	教学目的：掌握函数概念。 教学要求：掌握如何自定义函数。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：函数概念。 教学难点：自定义函数。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在项目实际开发的过程中，经常需要重复执行某些操作，比如数据更新、数据查询、数据排序等，如果每个模块的操作都重新编写一次代码的话，会如何？</p> <p>(学生回答)</p> <p>老师：没错，这样不仅加大了开发人员的工作量和开发时间，而且对后续的维护也有较大影响。因此，在程序开发中，可以使用函数来完成这种重复性的操作，从而引入函数的概念。</p> <p>二、学习新知</p> <p>1. 函数的概念和分类</p> <p>讲解：函数是一段实现特定功能的代码段，只需要编写一次，需要使用时直接调用该函数就可实现指定的功能，大大提高程序员的开发效率，提高代码的可读性。PHP 的函数可以分为三种：自定义函数、PHP 内置函数和可变函数。</p> <p>2.自定义函数</p> <p>(1) 函数的定义和调用</p> <p>讲解：函数定义的格式和调用的方式。</p> <p>举例：无参函数调用。</p> <pre><?php function output() //定义函数 output() { echo "*****"; //函数体 } output(); //调用函数 output() ?></pre>				

举例：有参函数调用。

```
<?php
function introduce($name)      //定义函数 introduce ()
{
    echo "我的姓名是:". $name;  //函数体
}
echo introduce('张三');        //调用函数 introduce ()
?>
```

(2) 函数的返回值

讲解、举例、演示：带返回值的函数的调用过程。

(3) 函数的参数

讲解：值传递、引用传递和默认值传递的过程。

举例：函数参数值传递。

```
<?php
function autoadd($n)           //定义函数 autoadd ()
{
    $n = $n + 1;
    return $n;
}
$age = 9;
echo autoadd($age);            //调用函数 autoadd ()
echo '<br>';
echo $age;
?>
```

举例：函数参数引用传递。

```
<?php
function autoadd(&$n)          //定义函数 autoadd ()
{
    $n = $n + 1;
    return $n;
}
$age = 9;
echo autoadd($age);            //调用函数 autoadd ()
echo '<br>';
echo $age;
?>
```

举例：设置函数参数默认值。

```
<?php
function autoadd($a, $b = 1)   //定义函数 autoadd ()
{
```

```

    $a = $a + $b;
    return $a;
}
$age = 9;
echo autoadd($age, 2);           //调用函数 autoadd ()
echo '<br>';
echo autoadd($age);             //调用函数 autoadd ()
echo '<br>';
echo $age;
?>

```

(4) 变量的作用域

讲解：举例讲解局部变量与全局变量的区别。

(5) 嵌套调用和递归调用

讲解：举例讲解嵌套调用和递归调用的含义和执行过程。

3. 可变函数和匿名函数

讲解：可变函数和匿名函数的含义，通过例题讲解其使用方法和执行过程。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 自定义求和函数。

教学后记： 主要讲解函数概念，如何自定义。重点解释清楚在什么情况下要自定义函数，以及函数的形参和实参。

教学课次	8	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	内置函数				
教学目的 与要求	<p>教学目的：握 PHP 常用的内置函数的使用，包括数学函数、日期时间函数和字符串函数。</p> <p>教学要求：掌握文件包含语句的使用。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：日期时间函数、字符串函数。</p> <p>教学难点：日期时间函数、字符串函数。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：上节课中我们学习了自定义函数的定义和调用，那么有没有封装好的函数，可以直接拿过来调用呢？</p> <p>(学生回答)</p> <p>老师：是的，有一些函数是系统封装好的，在 PHP 中有一些内置函数，本节我们学习数学函数、日期时间函数和字符串函数，从而引入内置函数的介绍。</p> <p>二、学习新知</p> <p>1. 数学函数</p> <p>讲解：PHP 内置了一系列的数学函数，用于获取最大值、最小值、绝对值、平方根、生成随机数等等。</p> <p>演示：abs()、ceil()、floor()、max()、min()、pow()、pi()、sqrt()、round()、rand()这些函数的使用。</p> <p>2. 日期时间函数</p> <p>讲解：使用 PHP 开发 Web 应用程序时，经常会处理和时间相关的问题，比如，用户访问网站的时间、购买产品下订单的时间、用户登录的时间等等。PHP 提供了日期时间函数，来满足开发者的需求。</p> <p>演示：time ()、date ()、mktime()、strtotime ()、microtime ()函数的语法格式和功能，并演示其含义。</p> <p>强调概念：</p> <p>所谓 Unix 时间戳，是一种时间的表示方式，定义了从格林威治时间 1970 年 1 月 1 日 00 时 00 分 00 秒起到现在的总秒数，以 32 位二进制数表示，其中 1970 年 1 月 1 日零点也叫 Unix 纪元。</p>				

重点讲解例子：商品秒杀倒计时。

```
<?php
date_default_timezone_set('PRC');
//秒杀的开始时间，以字符串存放到变量$starttimestr 中
$starttimestr = '2020-4-14 12:00:00';
//秒杀的结束时间，以字符串存放到变量$endtimestr 中
$endtimestr = '2020-4-14 12:15:00';
//将存放开始时间的字符串变量$starttimestr 转换成时间戳，存放到变量$starttime 中
$starttime = strtotime($starttimestr);
//将存放结束时间的字符串变量$endtimestr 转换成时间戳，存放到变量$endtime 中
$endtime = strtotime($endtimestr);
//获得当前时间，存放到变量$nowtime 中
$nowtime = time();
//如果当前时间小于秒杀活动的开始时间，提示活动还未开始
if ($nowtime < $starttime) {
    die("活动还没开始,活动时间是： {$starttimestr}至 {$endtimestr}");
}
//如果当前时间小于等于秒杀活动的结束时间，计算剩余的时间
if ($endtime >= $nowtime) {
    $lefttime = $endtime - $nowtime;    //计算实际剩下的时间（秒）
    $leftmin = intval($lefttime / 60);
    $leftsec = $lefttime % 60;
    echo '活动还剩' . $leftmin . '分' . $leftsec . '秒';
}
//如果当前时间大于秒杀活动的结束时间，提示活动已经结束
else {
    $lefttime = 0;
    die('活动已经结束！ ');
}
?>
```

3.字符串函数

讲解：

在实际的程序开发中，经常需要对字符串进行处理，PHP 提供了丰富的字符串函数，比如获取字符串的长度、去掉字符串首尾空格、分割字符串等等。

演示: strlen()、substr()、strcmp ()、strstr ()、substr_count ()、str_ireplace ()、ltrim ()、 rtrim ()、trim ()、explode ()、implode ()、strev ()、str_repeat ()、strchr () 等函数的功能和含义。

4.PHP 文件包含语句

讲解: include 语句与 require 语句的含义、使用和区别。

演示: include 语句和 require 语句的使用。

5.案例讲解: 随机验证码的生成

- (1) 案例呈现: 显示案例的最终效果。
- (2) 案例分析: 分析案例所用的知识点。
- (3) 案例实现: 讲解并演示案例实现的过程。

```
<?php
function RandNum($length)
{
    $a = range('a', 'z');
    $b = range('A', 'Z');
    $c = range(0, 9);
    $arr = array_merge($a, $b, $c);
    $n=count($arr);
    $str="";
    for($i=0;$i<=$length-1;$i++)
    {
        $str=$arr[rand(0,$n-1)];
    }
    return $str;
}
echo RandNum(5);
?>
```

三、课堂小结, 布置作业

对这节的主要内容进行总结, 布置作业。最后留几分钟让学生和老师或者同学之间互相讨论, 加深课程内容印象, 及时提出问题、解决问题。

作业/思考题: 课后习题。

教学后记: 解释清楚什么是时间戳, 它的作用, 在什么时候要用到它。

教学课次	9	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	数据交互				
教学目的 与要求	教学目的：掌握 GET 方式提交和获取表单数据。 教学要求：实现该实训相关要求。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：GET、POST 方式提交和获取表单数据。 教学难点：获取一组单选按钮、复选框的值。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：我们平时在登录网站的过程中，经常需要输入用户名、密码等信息，在我们提交了这些信息之后，服务器是如何获取到我们提交的数据的呢？</p> <p>（学生回答）</p> <p>老师：在网站开发过程中，前台页面和后台数据库之间通常需要数据传递，称之为数据交互。例如：在用户注册时，需要把表单中的数据提交到服务器，经过服务端脚本处理之后，存储到数据库中；在编辑用户信息时，又需要把用户数据从数据库中读取出来，呈现在页面中。</p> <p>二、学习新知</p> <p>1. 页面间参数传递</p> <p>问题：在查看所有学生信息时，有时需要对个别学生的信息进行修改，一般通过超链接跳转至编辑页面进行处理。在修改不同学生的信息时，均是链接到同一个编辑页面，此时，为了区分不同学生，需要把学生的学号传递给编辑页面。应该如何传递参数？</p> <p>讲解：页面间传递参数的方法，可以通过在 URL 后面使用“？”号连接“key=value”的键值对实现，其中 key 为参数名，value 为参数值。当传递多个参数时用&号进行连接。</p> <p>演示：结合例 5-1 演示如何在页面之间传递参数。</p> <p>2. 表单数据交互</p> <p>提问：复习表单的提交方式有哪两种？表单的 action 属性的作用？</p>				

(学生回答)

讲解：当表单以 GET 方式发送数据时，表单数据以键值对的形式附加在 URL 后面发送给服务器，服务端通过超全局变量\$_GET 读取数据。

演示：结合例 5-2、5-3 演示表单以 get 请求方式发生数据时，服务器获取表单数据的方法。分别讲解表单获取单个数据和多个数据的方式。

讲解：当表单以 POST 方式向服务端发送数据时，PHP 通过超全局变量\$_POST 来获取表单提交的数据。

演示：结合例 5-4 演示表单以 post 请求方式发生数据时，服务器获取表单数据的方法。

通过例 5-5 演示服务器获取复选框数据的方法。

3. 案例：考试答题

讲解：无纸化办公以快捷方便、省时省力、成本低、效率高等优点得到各界人士的青睐。对于一些客观题目，在线考试答题是一个非常好的考试方式，它不仅可以实现自动阅卷，还能对考试结果进行精准分析。

演示考试答题的页面运行效果，并分析考试答题需要实现的功能：

- 1) 显示考试答题页面。
- 2) 通过 PHP 脚本获取用户提交的选项。
- 3) 在页面显示用户提交的选项。

总结实现考试答题的步骤：

- 1) 创建考试答题页面，设置每个选项的 name 值。
- 2) 给表单添加提交动作，单击“交卷”按钮跳转到 PHP 处理页面。
- 3) 利用 PHP 脚本获取用户提交的选项，并显示在页面中。

演示：按照实现步骤依次进行演示，特别注意选项值的设置，以及为什么要这么设置。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 完成实训报告一。

教学后记： 授课时根据学生对表单掌握情况，建议在把详细表单讲解一下。

教学课次	10	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	Mysql 数据库				
教学目的 与要求	<p>教学目的：了解 MySQL 数据库的特点、下载与安装； 掌握 MySQL 数据库的创建、查看、修改和删除。</p> <p>教学要求：掌握 MySQL 数据库表的管理；掌握 MySQL 数据库记录的管理。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：字符集和校对规则的理解，多表查询。</p> <p>教学难点：字符集和校对规则的理解，多表查询。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：有同学知道网站上的数据在哪里存放的吗？</p> <p>(学生回答，老师总结)</p> <p>老师：数据存放在数据库中，本章我们将学习 MySQL 关系型数据库，进而引入。</p> <p>二、学习新知</p> <p>1. MySQL 概述</p> <p>讲解：讲解 MySQL 数据库的由来、特点，下载与安装的方法，以及如何启动和关闭 MySQL 服务器。</p> <p>2. MySQL 数据库的管理</p> <p>(1) 创建数据库</p> <p>(2) 查看数据库</p> <p>(3) 修改数据库</p> <p>(4) 删除数据库</p> <p>讲解：</p> <ul style="list-style-type: none"> ➤讲解其语法格式； ➤通过示例并演示讲解其实现过程； ➤强调注意点。 <p>a:创建数据库时一般指明字符集，校对规则一般默认即可。</p> <p>b:如果要删除的数据库不存在，则删除会失败。如果使用 if exists, 则可以避免删除不存在的数据库时出现的 MySQL 错误信息。</p>				

3.MySQL 数据库表的管理

(1) 数据类型

讲解：数据类型的分类和每种数据类型使用的场合，并举例说明。

(2) 创建数据表

(3) 修改数据表

(4) 删除数据表

讲解：

- 讲解其语法格式；
- 通过示例并演示讲解其实现过程；
- 强调删除时的注意点：删除数据表的操作要谨慎，一旦删除数据表，那么表中的数据将会全部清除。

4.MySQL 数据表记录的管理

(1) 数据表记录的添加

讲解：添加一条记录和多条记录的方法。

(2) 数据表记录的更新

(3) 数据表记录的删除

讲解：

- 讲解其语法格式；
- 通过示例并演示讲解其实现过程；
- 对比删除时使用 truncate 和 delete 的区别：使用关键字 TRUNCATE 删除整个表数据的效率会更高。

(4) 数据表记录的查询

讲解并演示：

a,单表查询的语法格式，如何查询行、查询列、查询满足某些条件的记录、如何分组查询、聚合函数的使用、分组后数据记录的筛选、显示某几条记录等等。

b,多表查询的语法格式，连接谓词和 join 连接的语法，子查询，比较子查询，exists 关键词的使用、union 关键字的使用等。

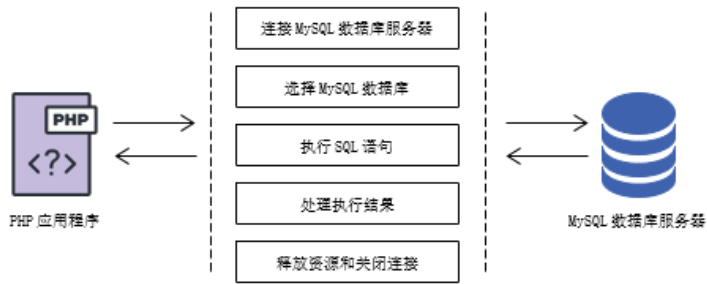
三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 完成实践操作。

教学后记：数据库知识，后面还有课程单独学习，本课主要讲解 MySql 基本使用。

教学课次	11	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	PHP 操作数据库				
教学目的 与要求	<p>教学目的：了解 PHP 中常用的数据库扩展。</p> <p>教学要求：掌握 PHP 操作数据库的一般步骤；掌握 PHP 操作数据库的方法。</p>				
教学方法 与手段	<p>教学手段：LEEPEE 教学法 项目驱动 案例教学</p> <p>教学方法：多媒体</p>				
教学重点 与难点	<p>教学重点：PHP 操作数据库的一般步骤。</p> <p>教学难点：PHP 操作数据库的方法。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>引入新课</p> <p>1.老师组织学生回顾在 MySQL 中如何实现数据的增删查改操作，进而提出问题：如何在 PHP 中实现对数据库中数据的增删查改操作？</p> <p>2.老师演示一个 PHP 显示数据和增加数据的案例，让学生继续思考在刚才的案例中数据从何处来？数据又是如何写入数据表中去？激发学生的学习兴趣。</p> <p>二、学习新知</p> <p>1.PHP 中常用的数据库扩展</p> <p>PHP 本身并不能直接操作数据库，而是要借助相应的数据库扩展来实现 PHP 应用和数据库之间的交互。PHP 中常用的数据库扩展有 MySQL、MySQLi 和 PDO 三种方式。</p> <p>重点分别介绍三种扩展方式的特点。</p> <p>2.PHP 操作数据库的一般步骤</p> <p>组织学生回忆去图书馆借还书的步骤：</p> <p>找到图书馆→刷卡认证→选择书库→进行借还操作→离开图书馆</p> <p>引出 PHP 操作数据库的一般步骤：连接数据库服务器、选择数据库、执行 SQL 语句、处理执行结果、关闭连接和释放资源五个步骤</p>				



3.PHP 操作数据库的方法

(1) 连接数据库服务器

讲解：MySQLi 扩展提供了 `mysqli_connect()`函数来进行连接，其基本语法如下：

```
mysqli mysqli_connect([string $host[, string $username[,
    string $password[,
    string $dbname
    [, string $port[,string $socket]]]]]);
```

执行成功返回一个 MySQLi 对象，表示到 MySQL 服务器的连接；执行失败则返回 `false`。

演示：连接数据库服务器

(2) 删除 Cookie

讲解：一台数据库服务器上可创建多个数据库，使用 PHP 进行数据库操作时需指定本次操作的数据库名称。在使用 `mysqli_connect()`函数连接数据库服务器时可通过第 4 个参数指定要操作的数据库名称，也可以在连接数据库服务器后使用 `mysqli_select_db()`函数选择要操作的数据库。`mysqli_select_db()`函数的语法如下：

```
bool mysqli_select_db(mysqli $link, string $dbname);
```

(3) 设置编码方式

PHP 访问 MySQL 时使用的字符集可以使用 `mysqli_set_charset()`函数设置，其语法如下：

```
bool mysqli_set_charset(mysqli $link,string $charset);
```

强调：为避免乱码，建议将 PHP 文件的 header 头编码、网页的 `<meta>`标记、PHP 访问 MySQL 时使用的字符集、MySQL 数据库的字符集设置为统一的字符集。

(4) 执行 SQL 语句

MySQLi 扩展提供了 `mysqli_query()`函数实现对数据库发送、执

行 SQL 语句。其语法如下：

```
mixed mysqli_query(mysqli $link,string $query[,int $resultmode]);
```

(5) 读取结果集

讲解：mysqli_query() 函数执行查询类 SQL 语句返回的 mysqli_result 对象不能直接被使用，需要将对象中的数据取出保存在数组中才能在页面中进行输出显示。MySQLi 扩展提供的处理结果集的函数有 mysqli_fetch_assoc()、mysqli_fetch_row()、mysqli_fetch_array()、mysqli_fetch_object()、mysqli_fetch_all()。

演示：例 7-1 比较读取结果集的几种方法

(6) 释放资源和关闭连接

mysqli_free_result() 函数用于关闭结果集对象，以释放系统资源，其语法如下：

```
void mysqli_free_result(resource $result)
```

mysqli_close() 函数用于断开与 MySQL 数据库服务器的连接，其语法如下：


```
bool mysqli_close(mysqli $link)
```

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：完成实训日志二。

教学后记：重点解释清楚连接数据库、从数据库中调出一个表、从表中调出一条记录。一定要让学生搞懂，可以重复讲解。

教学课次	12	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它																																		
授课内容	数据搜索																																						
教学目的 与要求	教学目的：掌握数据显示的实现方法。 教学要求：掌握数据搜索的实现方法。																																						
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书																																						
教学重点 与难点	教学重点：数据显示的实现方法。 教学难点：数据搜索的实现方法。																																						
教 学 过 程 设 计	教 学 内 容																																						
	<p>一、引入新课</p> <p>展示本节课要实现的案例效果，引导学生讨论案例的应用场景并简单思考分析其实现的流程，引入新课。</p> <p>二、学习新知</p> <p>1.数据显示</p>  <p>The screenshot shows a web browser window with the title '学生党员信息' (Student Party Member Information). The browser address bar shows 'demo.com/list.php'. The page content includes a table with the following data:</p> <table border="1" data-bbox="483 1218 1235 1451"> <thead> <tr> <th>学号</th> <th>姓名</th> <th>性别</th> <th>班级</th> <th>入党时间</th> </tr> </thead> <tbody> <tr> <td>20171131</td> <td>杜云</td> <td>女</td> <td>计算机科学与技术1班</td> <td>2018-05-25</td> </tr> <tr> <td>20171135</td> <td>张凤</td> <td>女</td> <td>计算机科学与技术1班</td> <td>2018-05-26</td> </tr> <tr> <td>20171157</td> <td>王亮</td> <td>男</td> <td>计算机科学与技术1班</td> <td>2019-05-15</td> </tr> <tr> <td>20171240</td> <td>李军</td> <td>男</td> <td>计算机科学与技术2班</td> <td>2019-05-16</td> </tr> <tr> <td>20181266</td> <td>王欣</td> <td>男</td> <td>计算机科学与技术2班</td> <td>2019-05-23</td> </tr> <tr> <td>20181330</td> <td>张军</td> <td>男</td> <td>软件工程1班</td> <td>2019-05-15</td> </tr> </tbody> </table> <p>(1) 准备数据</p> <p>创建一张学生党员信息表 memberInfo，表中包含学号、姓名、性别、班级、入党时间字段。</p> <p>(2) 分离数据库连接</p> <p>为避免频繁书写数据库连接代码，提高代码的可维护性，通常将数据库连接单独写带一个文件中，在需要进行数据库操作前通过 include、include_once、require 或 require_once 等语句以文件包含的形式将其引入即可。</p> <p>(3) 制作数据文件</p> <p>数据文件中需要首先加载数据库连接文件 conn.php，然后执行相关查询语句，之后使用 mysqli_fetch_all()函数将结果集中的数据读</p>					学号	姓名	性别	班级	入党时间	20171131	杜云	女	计算机科学与技术1班	2018-05-25	20171135	张凤	女	计算机科学与技术1班	2018-05-26	20171157	王亮	男	计算机科学与技术1班	2019-05-15	20171240	李军	男	计算机科学与技术2班	2019-05-16	20181266	王欣	男	计算机科学与技术2班	2019-05-23	20181330	张军	男	软件工程1班
学号	姓名	性别	班级	入党时间																																			
20171131	杜云	女	计算机科学与技术1班	2018-05-25																																			
20171135	张凤	女	计算机科学与技术1班	2018-05-26																																			
20171157	王亮	男	计算机科学与技术1班	2019-05-15																																			
20171240	李军	男	计算机科学与技术2班	2019-05-16																																			
20181266	王欣	男	计算机科学与技术2班	2019-05-23																																			
20181330	张军	男	软件工程1班	2019-05-15																																			

取到二维数组\$dataArr 中，最后加载视图文件进行显示即可。

(4) 制作视图文件

在视图文件中制作一个两行五列的表格，其中表格第一行为表头，第二行为循环体。在视图文件中内嵌 PHP 代码，遍历输出数据文件查询得到的数据数组\$dataArr 即可实现数据显示

2.数据搜索

随着信息量增多，要想精确地查看某一个党员信息就比较困难，为实现方便、快捷地查找所需党员信息，在数据显示的基础上增加数据搜索功能。初始状态仍显示所有党员信息，同时可根据党员姓名进行模糊搜索。

分析搜索功能的实现原理：根据用户输入的关键词使用 LIKE 进行模糊查询即可。

(1) 添加搜索表单

在数据显示功能的视图文件中增加一个表单，表单中包含一个文本框和一个提交按钮。此外，为便于后续在对搜索结果进行分页时同时传递搜索关键词参数，在这里采用 GET 方式提交表单。

(2) 构造模糊查询

由于初始状态显示所有党员信息，因此定义 SQL 语句初值为查询 memberInfo 中所有信息，如果通过 GET 方式获取到用户提交的数据则在原有基础上拼接 where 条件构造模糊查询语句。由于用户可通过表单提交任意数据，出于安全角度考虑使用 mysqli_real_escape_string()函数对 SQL 语句中的特殊字符进行转义处理。

(3) 多关键词搜索

多关键字搜索是指用户可以一次输入两个或两个以上的关键字进行搜索，关键字之间一般用空格分隔。与单关键字搜索相比，这种搜索方式更为灵活，在实际应用中使用更为广泛。


多关键词搜索的原理：首先使用 explode()函数将用户提交的关键字根据“空格”分隔符拆分为数组，然后遍历这个关键字数组，在循环体中构造多个 where 查询条件，多个条件之间为 or 关系。在循环结束之后使用 rtrim()函数去除右侧多余的“or”即可拼接成完整的查询语句。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 完成上课实践。

教学后记： 主要介绍数据库的搜索，主要说下搜索为什么要用 GET 来提交。学生在分页搜索上，有点不理解，后面可多说明一下。

教学课次	13	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它														
授课内容	数据分页																		
教学目的 与要求	教学目的：掌握数据分页的原理。 教学要求：掌握数据分页的实现方法。																		
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书																		
教学重点 与难点	教学重点：掌握数据分页的原理。 教学难点：掌握数据分页的实现方法。																		
教 学 过 程 设 计	教 学 内 容																		
	<p>一、引入新课</p> <p>展示本节课要实现的案例效果，引导学生讨论案例的应用场景并简单思考分析其实现的流程，引入新课。</p> <p>二、学习新知</p> <p>学生党员数量会不断增多，在一个页面中一次显示所有学生党员信息不仅可读性差，也会影响页面加载的速度。一般地，当数据量较大时会对数据进行分页显示。</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;">  <p style="text-align: center;">学生党员信息</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>学号</th> <th>姓名</th> <th>性别</th> <th>班级</th> <th>入党时间</th> </tr> </thead> <tbody> <tr> <td>20171131</td> <td>杜云</td> <td>女</td> <td>计算机科学与技术1班</td> <td>2018-05-25</td> </tr> <tr> <td>20171135</td> <td>张凤</td> <td>女</td> <td>计算机科学与技术1班</td> <td>2018-05-26</td> </tr> </tbody> </table> <p style="font-size: small; text-align: center;">每页2条数据 当前第1页 共3页 首页 上一页 下一页 末页</p> </div> <p>1.两种数据分页方式</p> <p>由数据分页效果引出数据分页的两种方式：物理分页和逻辑分页。</p> <p>进一步介绍物理分页和逻辑分页的概念和优、缺点。</p> <p>(1) 物理分页依赖于数据库这个物理实体，在 MySQL 数据库中依赖 limit 子句来实现。物理分页在每次翻页时都需要访问数据库以获取当前页的数据，所占用的内存空间较少，但需要频繁操作数据库，因此对数据库服务器造成的压力较大。</p> <p>(2) 逻辑分页则是一次从数据库中读取全部数据，然后由程序人员通过相应的代码控制每页应该显示的数据范围。逻辑分页减少了对数据库服务器的访问次数，但却需占用较多的内存开销，且不</p>					学号	姓名	性别	班级	入党时间	20171131	杜云	女	计算机科学与技术1班	2018-05-25	20171135	张凤	女	计算机科学与技术1班
学号	姓名	性别	班级	入党时间															
20171131	杜云	女	计算机科学与技术1班	2018-05-25															
20171135	张凤	女	计算机科学与技术1班	2018-05-26															

能实时反映出数据库中的最新数据变化，适合数据量较小且数据变化频率较低的情况。

2. 分页的原理

借助 LIMIT 子句可用于强制 SELECT 语句返回指定的记录数。

在 LIMIT 子句中需要确定偏移量和记录数，记录数由每页显示的数据量来确定，偏移量和当前的页数有关，偏移量 $offset=(page-1)*pageSize$ ，因此只需要确定当前是第几页即可实现数据分页，而页码可在浏览数据时通过 GET 方式传递。

3. 实现数据分页

- 1) 定义每页显示的记录数；
- 2) 查询数据表中的总记录数，将总记录数除以每页显示的记录数并向上取整得到总页码；
- 3) 对传入的页码进行最大、最小临界值验证，判断页码的合法性；
- 4) 计算当前页读取数据的偏移量；
- 5) 执行带有 LIMIT 子句的 SQL 语句并处理结果集；
- 6) 加载视图文件进行显示。

演示：数据分页的实现

4. 查询参数的处理

演示：根据以上讲解对搜索结果进行分页，在翻页时出现所有的数据而并非搜索结果的第二页数据，组织学生思考出现这种情况的原因是什么？

教师总结：由于在翻页时没有传递搜索的关键词导致搜索关键词丢失，故显示的是对所有数据的分页结果。

教师提问：如何在翻页的同时传递查询参数？

讲解：在处理这类查询参数时，首先通过 `$_GET` 来获取所有的 GET 传值，在得到的数组 `$params` 中包含了当前的页码参数 `page` 和其他参数，由于翻页时需传递新的 `page` 参数，故使用 `unset()` 函数删除 `$params` 中的原有 `page` 参数，然后使用 `http_build_query()` 函数将数组 `$data` 生成为一个经过 URL encode 的请求字符串，最后再拼接上新的 `page` 参数值。

演示：对搜索结果进行分页

5. 对分页导航进行优化

学生党员信息

学号	姓名	性别	班级	入党时间	修改	删除
20171380	李芳	女	软件工程2班	2019-05-15	修改	删除

每页1条数据 当前第6页 共11页

1 ... 3 4 5 6 7 8 9 ... 11

假设以当前页码为中心显示左右各 3 页的页码, 定义 \$pageOffset 表示页码偏移量, \$pageHtml 表示分页导航对应的 HTML 内容。如果当前页码减去偏移量大于 1 则显示首页页码, 如果当前页码加上偏移量小于总页码则显示尾页页码, 然后循环输出以当前页码为中心的左右页码即可。

强调: 在循环输出过程中需要对页码的边界值进行验证, 避免出现非法页码。对于当前页码添加样式表中定义的 current 样式, 以高亮显示当前页码。

5. 分页效率的优化

可在查询过程中记录上一页数据中最后一条数据的 ID, 假设上一页最后一条数据的 ID 是 198711, 然后通过下面的语句来读取当前页 10 条数据:


```
select * from memberInfo where id>198711 order by id desc
limit 10
```

三、课堂小结, 布置作业

对这节的主要内容进行总结, 布置作业。最后留几分钟让学生和老师或者同学之间互相讨论, 加深课程内容印象, 及时提出问题、解决问题。

作业/思考题: 完成课程综合作业。

教学后记: 主要解释清楚分页原理, 不然学生不易理解。

教学课次	14	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	数据的增删改查				
教学目的 与要求	教学目的：掌握数据添加的实现方法。 教学要求：掌握数据修改的实现方法；掌握数据删除的实现方法。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：掌握数据添加的实现方法；掌握数据修改的实现方法； 掌握数据删除的实现方法。 教学难点：掌握数据添加的实现方法。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>展示本节课要实现的案例效果，引导学生讨论案例的应用场景并简单思考分析其实现的流程，引入新课。</p> <p>二、学习新知</p> <p>1.数据添加</p> <div style="text-align: center;">  </div> <p>(1) 制作添加表单</p> <p>在表单中需要添加文本框、单选按钮、下拉列表、日期等表单元素，并为每个表单元素添加 name 属性值以便于获取用户输入的内容。当提交表单时，以 POST 方式提交给当前页面</p> <p>(2) 处理表单数据</p> <p>在数据文件中，首先通过\$_POST 判断有无提交表单。如果没有提交表单则加载表单视图文件，显示添加页面；如果提交了表单则首先获取用户填写的相应数据，然后拼接组成一个完整的 INSERT 语句，最后执行该语句实现数据添加功能。</p> <p>(3) 优化添加语句</p>				

在实现数据添加功能时，如需填写的信息比较多，则在数据文件中需要一一获取用户填写的信息并拼接 SQL 语句，操作较为繁琐且容易出错。针对添加单条数据的操作，可使用 insert into set 语句代替 insert 语句实现自动拼接。insert into set 语句的语法如下：

```
insert into table set field1 = value1,field2 = value2...
```

首先使用\$_POST 一次获取用户提交的所有数据，然后遍历该数组，在循环体中自动拼接完成 insert into set 语句中 field 和 value 的对应关系。循环结束后使用 rtrim()函数去除语句最右侧多余的逗号即可得到完整的添加语句。

2.数据修改

(1) 制作“修改”链接

单击该链接跳转到数据修改页面 modify.php，同时使用 GET 方式传递当前数据所对应的主键值，以便于在数据修改页面显示该信息的原始数据。

(2) 制作修改表单

修改表单与数据添加的表单相同，只是在表单中通过添加 value 属性来显示该信息的原始数据。

(3) 实现数据修改

在数据修改的数据文件中判断是否提交表单，如未提交表单则根据接收的 mebID 值查询原始数据信息；如提交表单则做数据更新操作。实现数据更新的过程与数据添加的过程类似，需要用 update 语句代替 insert into set 语句，且要注意拼接更新条件。

强调：执行更新语句 update 时一定要添加 where 条件，否则将会更新所有数据，且操作不可逆。

3.数据删除

The screenshot shows a web browser window with the title '学生党员信息' and the URL 'demo.com/paginate.php'. The page content includes a table with the following data:

学号	姓名	性别	班级	入党时间	修改	删除
20171131	杜云	女	计算机科学与技术1班	2018-05-25	修改	删除
20171135	张凤	女	计算机科学与技术1班	2018-05-26	修改	删除

Below the table, there is a pagination bar: '每页2条数据 当前第1页 共3页 首页 上一页 下一页 末页'.

(1) 制作“删除”链接

单击“删除”链接时通过 GET 方式传递要删除信息的主键值，同时为避免误操作使用 JavaScript 中的 confirm() 方法弹出消息框再次提醒用户是否确定删除。

(2) 实现删除操作

接收 GET 方式传递的主键值，然后拼接完整的 delete 语句并执行即可完成数据删除功能。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：完成课程综合作业。

教学后记：学生在添加和修改语句容易出错，主要是数据类型问题。学生初学在加不加引号出错率高。

教学课次	15	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	PHP 操作数据库常见错误				
教学目的 与要求	教学目的：了解 PHP 操作数据库常见错误及排错方法。 教学要求：掌握数据库操作中的常见 Web 安全问题与防御措施。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：讲练结合和演示法。 教学难点：利用多媒体计算机辅助教学。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>7.5 节安排学生课前自主学习。</p> <p>(1) 演示一个用户登录的案例，在用户名处输入 1' or 1=1#，密码处输入任意内容，点击“登录”按钮进行登录，结果登录成功。</p> <p>(2) 组织学生思考讨论为什么会登录成功？</p> <p>(3) 教师将案例中本次执行的查询语句进行打印输出并总结出出现这种情况的原因。</p> <p>二、学习新知</p> <p>1.SQL 注入</p> <p>攻击者通过把非法的 SQL 命令插入到 Web 表单中或页面请求查询字符串中提交给服务器，最终达到欺骗服务器执行恶意 SQL 语句的目的。SQL 注入包含常规注入、宽字节注入、Base64 编码注入等。</p> <p>(1) 原理分析</p> <p>开发者没有对用户提交的数据进行过滤或者过滤被攻击者绕过，从而在将获取的用户数据拼接到 SQL 语句中执行时，被用户插入的恶意 SQL 代码非法操作数据库，导致敏感信息泄露或数据库被破坏，甚至其他更为严重的后果。</p> <p>(2) 防御方法</p> <p>针对 SQL 注入产生的原因，避免将用户传递的数据直接拼接在 SQL 语句中即可防止 SQL 注入。在 PHP 操作数据库中 MySQLi 扩展和 PDO 扩展均提供了相应的预处理函数来实现参数化查询。</p> <p>1) 创建预编译对象</p> <p>mysqli_prepare() 函数用于做好执行 SQL 语句的准备，其基本语法如下：</p>				

`mysqli_stmt mysqli_prepare (mysqli $link , string $query)`

强调：通过预编译，SQL 引擎会预先进行语法分析，产生语法树，生成执行计划，此后无论用户提交什么数据都只会作为字符串字面值参数而不会作为 SQL 命令来执行，因此预编译可以有效防御 SQL 注入。

2) 绑定参数

`mysqli_stmt_bind_param()` 函数用于将变量作为参数绑定到 `mysqli_prepare()` 创建的预编译 SQL 语句中，其基本语法如下：

`bool mysqli_stmt_bind_param (mysqli_stmt $stmt , string $types , mixed &$var1 [, mixed &$...])`

3) 执行预处理语句

`mysqli_stmt_execute()` 函数用于执行由 `mysqli_prepare()` 函数创建的预编译对象，执行时所有的参数标记将被替换成绑定的参数，其基本语法如下：

`bool mysqli_stmt_execute (mysqli_stmt $stmt)`

`mysqli_stmt_execute()` 函数中 `stmt` 表示由 `mysqli_prepare()` 创建的预编译对象，该函数执行成功返回 `true`，执行失败返回 `false`。如果执行的语句是 `UPDATE`、`DELETE` 或 `INSERT` 等非查询语句，则可以使用 `mysqli_stmt_affected_rows()` 函数来确定受影响行的总数。

如果执行的是 `SELECT` 等类型的查询语句，则可以使用 `mysqli_stmt_get_result()` 函数从执行后的预编译对象中获取结果集，其基本语法如下：

`mysqli_result mysqli_stmt_get_result (mysqli_stmt $stmt)`

演示：使用参数化查询改进用户登录案例并再次使用课前的测试数据进行测试。

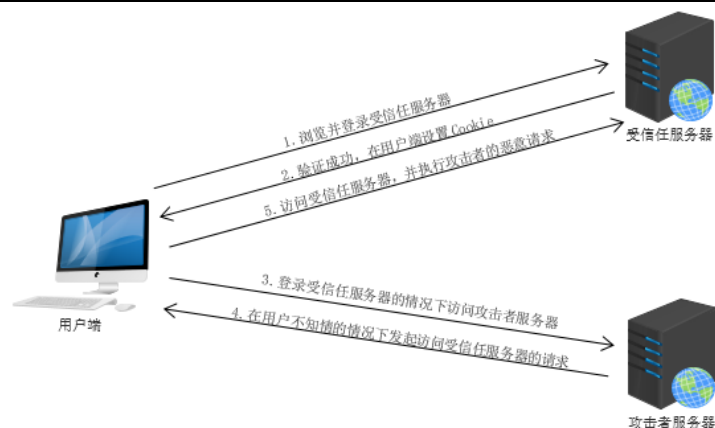
强调：在参数化查询中，模糊匹配的通配符在绑定参数时进行处理。

2.CSRF 跨站请求伪造

攻击者盗用了用户的身份，并以用户的名义向第三方网站发送恶意请求。通过 CSRF 可以盗用用户的身份发邮件、发短信、进行交易转账等。

(1) 原理分析

CSRF 的原理是在用户已登录受信任服务器的前提下，攻击者诱导访问攻击者服务器，然后在攻击者服务器返回的页面中嵌入代码，从而在用户不知情的情况下以受害者的名义向受信任服务器发起请求，实现在并未授权的情况下执行在权限保护之下的操作。



(2) 防御方法

防御的基本思想是在请求中放入攻击者不能伪造的信息，并且该信息不存于 Cookie 中，然后在受信任服务器端以此信息作为判断正常访问和非法访问的标准。常用的防御 CSRF 的方法有验证 HTTP Referer 字段、token 验证、在 HTTP 头中自定义属性等。

1) 生成 token

用户访问页面时，生成一个伪随机数 token，将其保存在服务器 Session 中，同时在页面中隐藏保存该随机数，用户正常发起请求时会一并携带 token 值，而通过 CSRF 发起的欺骗性攻击中由于无法事先获取 token，导致不能正常回传 token 值而被服务器拒绝请求。

强调：加入 token 验证后要考虑“并行会话的兼容”，即如果用户在一个站点上同时打开了多个不同的表单，CSRF 保护措施不应该影响到他对任何表单的提交，因此生成的 token 应该保存在数组中，从而避免多个 token 之间的相互覆盖。

2) Web 表单保存 token

在 Web 表单中通过隐藏域保存生成的 token 值。

3) 验证 token


在服务端接收到 token 后验证该值是否有效，验证有效后为防止重复使用应及时将其删除。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：完成课后习题。

教学后记：主要讲解数据库常见错误及排除方法，让学生感觉如果代码书写不严谨，可能会出现 BUG。

教学课次	16	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	Cookie 管理				
教学目的 与要求	教学目的：理解 Cookie 的工作原理。 教学要求：掌握 Cookie 的使用方法。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：Cookie 的工作原理。 教学难点：Cookie 的使用方法。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在登录 163 邮箱时有一个“十天内免登录”的选项，如果勾选该选项则在下一次访问邮箱时无需输入用户名和密码即可正常登录邮箱系统，请大家思考这是如何实现的？</p> <div style="text-align: center;">  </div> <p>(学生回答)</p> <p>老师：Cookie 是在客户端浏览器存储用户信息的一种机制，通过 Cookie 可以实现对用户的跟踪和识别，常用在记录用户登录信息、保存用户浏览历史记录、定制主题等应用场景中。</p> <p>二、学习新知</p> <p>1.Cookie 的管理</p> <p>(1) 概念和特点</p> <p>讲解：Cookie 是服务端保存在客户端浏览器中的一种文本信息，该信息由服务器生成并以 key-value 键值对的形式发送给客户端，客户端浏览器再将其以文本文件的形式保存在一个文件夹中。当客户端再一次向服务器发起会话请求时会自动携带保存在 Cookie 中的信息，服务器辨别该 Cookie 信息，并以此 Cookie 信息来识别用户。</p> <p>以生活中商家在客户消费后免费赠送 VIP 卡为例进一步阐述 Cookie 的概念和特点。</p>				

Cookie 具有：存储在客户端、存储在客户端、对用户透明地工作、大多以明文形式进行存储、会跟踪用户的访问信息等特点。

(2) 工作原理

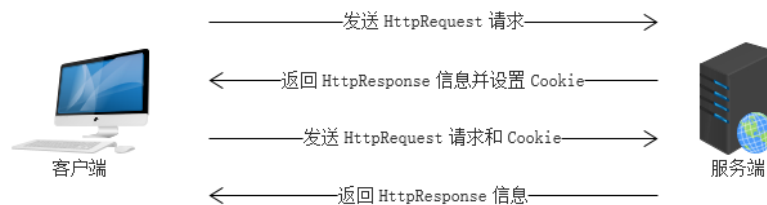
讲解：Cookie 的设置及发送会经历以下四个步骤：

1) 客户端发送 HttpRequest 请求到服务端。

2) 服务端返回 HttpResponse 响应到客户端，并在头部信息中包含了要设置的 Cookie 信息。客户端接收到相应信息后保存 Cookie 信息。

3) 客户端再次向服务端发送 HttpRequest 请求，并在头部请求信息中包含之前设置的 Cookie 信息。

4) 服务器通过 Cookie 信息识别用户，并返回 HttpResponse 响应信息。



(3) 设置 Cookie

讲解：通过 `setcookie()` 或 `setrawcookie()` 函数来设置 Cookie，`setcookie()` 函数的语法格式如下：

```
bool setcookie ( string $name [, string $value = "" [, int
                $expire = 0 [, string $path
                = "" [, string
                $domain = "" [, bool $secure = false [, bool $httponly =
                false ]]]]] )
```

`setcookie()` 函数创建 Cookie 成功返回 `true`，否则返回 `false`

演示：例 8-1 保存商品浏览记录

(4) 读取 Cookie

讲解：通过超全局变量 `$_COOKIE` 来获取客户端已存储的 Cookie 信息，如果不指定要获取的 `name` 值，则可以获取到存储的所有 Cookie 信息。

强调：如果设置 Cookie 和读取 Cookie 在同一个页面中，那么在设置 Cookie 之后并不能立即获取到 Cookie 信息。原因是在这一次请求中服务端返回了需要设置的 Cookie 信息，此次设置的 Cookie 信息只有在下一次发起访问请求时才会发送到服务端，此时才能读取到 Cookie 信息。

演示：例 8-2 显示商品浏览记录

(5) 删除 Cookie

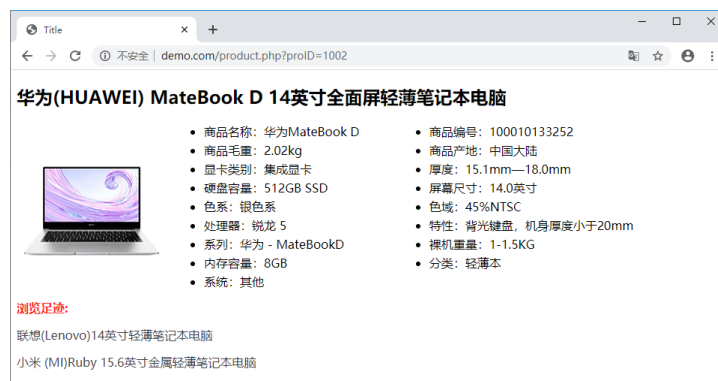
讲解：对于没有设置失效期的 Cookie，在关闭浏览器时会自动删除该 Cookie 信息；而设置了失效期的 Cookie，使用 setcookie() 函数删除 Cookie，只需将 setcookie() 函数中 value 参数设置为空，将 expire 参数设置为一个小于当前系统的时间即可。

2. 案例：商品浏览足迹

(1) 案例呈现

使用 Cookie 技术实现如下图所示的“商品浏览足迹”案例。在案例中主要实现以下功能：

- 1) 记录商品的浏览历史，并在页面中显示浏览足迹。
- 2) 浏览历史中不重复显示同一商品信息。
- 3) 按照用户浏览的先后顺序逆序显示浏览记录。



(2) 案例分析

- 1) 接收 GET 方式传递的商品 ID，并对商品 ID 进行验证。
- 2) 根据商品 ID 查询当前浏览商品的信息。
- 3) 判断之前是否已经存在浏览记录，如果不存在则将当前商品 ID 存入 Cookie 并转步骤 9。
- 4) 如之前已有浏览记录，则读取 Cookie 中的商品 ID 信息，并根据商品 ID 信息查询商品详情。
- 5) 将 Cookie 中存储的商品 ID 字符串根据逗号拆分为数组。
- 6) 在数组头部添加元素，以保证最后浏览的记录在最前面显示。
- 7) 删除数组中的重复商品 ID。
- 8) 将商品 ID 数组转换为以逗号分隔的字符串，并将此字符串存入 Cookie。
- 9) 加载视图文件，显示商品详情。

(3) 案例实现

- 1) 数据准备
- 2) 业务逻辑
- 3) 视图显示

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

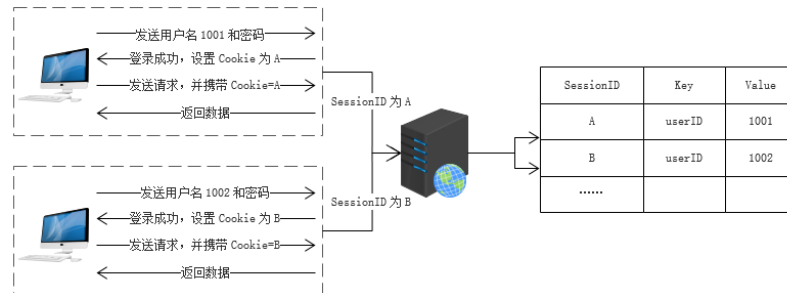
作业/思考题：使用 Cookie 实现网页更换主题的功能。

教学后记：本节课讲解 COOKIE 普通使用即可，不易讲过多深入知识。

教学课次	17	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	Session 的工作原理				
教学目的 与要求	教学目的：理解 Session 的工作原理。 教学要求：掌握 Session 的使用方法。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	重点：Session 的工作原理； 难点：Session 的使用方法。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：</p> <p>问题一：在教务管理系统中查询成绩之前是否必须登录？系统如何知道当前用户是否已登录？</p> <p>问题二：在教务管理系统中点击“查看成绩”链接即可显示当前已登录的用户，用户登录和查看成绩是在两个不同的页面，在成绩显示页面是如何知道当前已登录用户的信息？</p> <p>（学生回答）</p> <p>老师：Session 即会话，是针对 HTTP 的局限性而提出的另一种保持客户端和服务端会话状态的机制。与 Cookie 相比较而言，保存在 Session 中的数据存储在服务端，数据的安全性更高且不受存储长度的限制，但是在某一段时间内随着访问用户的增多会对服务器的性能有所影响。</p> <p>二、学习新知</p> <p>1. Session 的管理</p> <p>（1）概念</p> <p>讲解：Session 指的是从进入网站到关闭浏览器这段时间内的会话，是一个特定的时间概念。由于存储在 Session 中的值可以在生命周期中被当前站点的所有页面访问，因此可在访问者与网站之间建立一种“对话”机制，实现用户访问的连续性。Session 常被用在需要用户登录的场景中，当用户登录成功之后在服务端为其保存 Session，在访问其他功能页面时将 Session 作为判断用户是否有权限访问的凭证来使用。</p>				

(2) 工作原理

讲解：服务器存储的每一个 Session 都有一个唯一的标识，称之为 SessionID。SessionID 是一个由 PHP 随机生成的加密数字，这个 SessionID 会通过 HTTP 响应头返回，并以 Cookie 的方式保存在客户端。当该用户再次发起请求时会自动携带保存在 Cookie 中的 SessionID 发送到服务端，服务端接收到请求之后就会依据 SessionID 找到相应的 Session，从而识别该用户。



结合商场中的智能存包柜的工作流程讲解 Session 的工作流程。

(3) 设置 Session

讲解：在使用 Session 之前必须先启动一个会话，在 PHP 中使用 `session_start()` 函数来开启一个会话，其语法格式如下：

```
bool session_start(void)
```

开启会话后就可以通过超全局变量 `$_SESSION` 来保存 Session 信息，直接给该数组新增一个元素即可。在使用 Session 时会自动对要设置的值进行编码和解码，因此 Session 中可以存储任何数据类型。

演示：例 8-3 存储用户信息

(4) 读取 Session

讲解：读取 Session 时同样需要先使用 `session_start()` 开启会话，然后使用全局数组 `$_SESSION`，并根据创建 Session 时命名的数组键名进行读取。

演示：例 8-4 读取 Session 中用户信息

(5) 删除 Cookie

讲解：与删除普通数组元素相同，可以使用 `unset()` 函数删除 Session 中的某个值。如果需要删除 Session 中的所有信息，可直接将一个空数组赋值给 `$_SESSION`，以释放服务器内存中保存的 Session 信息。

演示：例 8-5 删除 Session 中用户信息

2. 案例：用户登录

(1) 案例呈现

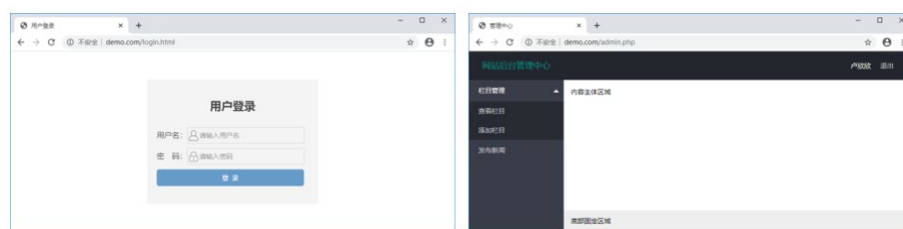
本节中使用 Session 技术实现如下图所示的“用户登录”案例。在

案例中主要实现以下功能：

1) 对用户输入的用户名、密码进行验证，验证通过后跳转到如图 8-6 所示的管理页面，验证失败时给出相应提示信息。

2) 如果用户在未登录状态下直接通过 URL 方式请求管理页面则强制跳转到登录页面进行登录。

3) 对用户的密码进行 MD5 加密，并使用加盐的方式增强其安全性，即在密码中混入一段随机的字符串，这个随机的字符串称为盐。加盐后可在一定程度上防止通过查表法、反向查表法和彩虹表来破解密码。



(2) 案例分析

1) 制作一个包含文本框、密码框、提交按钮的表单。

2) 接收表单提交的 POST 信息，并根据获取的用户名查询用户表中与此用户名匹配的密码、姓名、盐。

3) 判断步骤 2 的查询结果是否为空，若为空则提示用户名不存在，返回登录页面。不为空则转步骤 4。

4) 将用户填写的密码和查询到的盐采用注册时相同的算法进行加密处理，并与从数据表中查询到的密码进行比较。若不相等则提示密码错误，返回登录页面。否则登录成功转步骤 5。

5) 开启 Session，将用户 ID 和用户姓名分别存入 Session，并跳转到管理中心页面。

6) 在管理中心页面开启 Session，判断之前已设置的 Session 值是否存在。若存在则正常显示，若不存在则提示尚未登录或登录信息已失效，终止程序运行并强制重新登录。

(3) 案例实现

1) 数据准备

2) 制作登录页面

3) 登录业务逻辑

4) 验证是否登录

强调：本案例中将 4 位盐值直接拼接在原始密码后再进行 MD5 加密存储，这仅仅是为了方便讲解，在真实项目中不可采用这种短盐值和简易组合。

3.组织学生讨论总结 Cookie 和 Session 的区别

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：（1）使用 Session 实现网页更换主题的功能。

（2）综合运用 Cookie 和 Session 相关知识，实现可“7 天内免登陆”的用户自动登录功能。

教学后记：本节课讲解 Ssession 和 Cookie 有相似之处，所以要讲解清楚两者区别。

教学课次	19	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	文件基本操作				
教学目的 与要求	教学目的：掌握创建目录、删除目录；掌握移动目录、遍历目录。 教学要求：掌握文件的读写以及基本操作函数。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：目录的移动和遍历，文件的读写。 教学难点：目录的遍历，文件的读写。				
教 学 过 程 设 计	教 学 内 容				
	<p>引入新课</p> <p>老师提问：在电脑中经常会使用哪些文件或目录操作？ (学生回答)</p> <p>老师：新建、删除文件夹，查看文件夹下的内容，创建、删除文件，复制文件等。</p> <p>二、学习新知</p> <p>1. 创建目录、删除目录</p> <p>讲解：使用 <code>mkdir()</code> 创建目录，使用 <code>rmdir()</code> 删除空目录和非空目录。</p> <p>演示：分别演示非递归创建和递归创建目录，删除空目录的实现。</p> <p>2. 移动目录</p> <p>讲解：<code>file_exists()</code>、<code>basename()</code> 和 <code>rename()</code> 函数的使用，以及使用这 3 个函数实现移动目录功能的步骤。</p> <p>演示：演示移动目录的实现，并且提醒学生不同操作系统路径分隔符的区别。</p> <p>3. 遍历目录</p> <p>讲解：<code>opendir()</code>、<code>readdir()</code>、<code>closedir()</code>、<code>rewinddir()</code> 函数的使用，以及实现遍历目录功能的步骤。</p> <p>演示：遍历目录的实现，并且提醒学生在遍历任何一个目录时，都会包括“.”和“..”两个特殊的目录，“.”表示当前目录，“..”表示上一级目录。</p> <p>4. 创建文件</p>				

讲解：使用 `fopen()` 函数创建和打开文件，以及 `mode` 参数的常用值。

演示：以不同的方式打开文件，并且提醒学生为了移植性考虑，应该以“b”作为 `mode` 参数的最后一个字符。

5. 读写文件

讲解：PHP 中常用的读写文件的函数，重点讲解 `fgets()`、`file_get_contents()`、`readfile()`、`fwrite()` 和 `file_put_contents()` 函数的使用。

演示：使用 `fgets()`、`file_get_contents()`、`readfile()`、`fwrite()` 和 `file_put_contents()` 函数读写文件。

6. 文件的基本操作函数

讲解：`copy()`、`unlink()` 和 `rename()` 函数的使用。

演示：复制、删除、重命名文件的实现

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：完成课后习题。

教学后记：本节课有些难度，可以让学生大概了解一下，不用一定完全掌握。

教学课次	20	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	文件上传和下载				
教学目的 与要求	教学目的：概念掌握文件的上传和下载。 教学要求：通过综合案例掌握文件的常见操作。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：文件的上传、文件管理器的实现。 教学难点：文件的上传后改名。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：大家经常经常会在一些网站上传文件，思考下这是如何实现。</p> <p>(学生回答)</p> <p>老师：在上传文件的时候首先用户需要在客户端上传文件，然后服务器端接收上传的文件并处理。</p> <p>二、学习新知</p> <p>1. 文件的上传</p> <p>讲解：上传文件的实现过程，详细讲解客户端选择上传文件和服务器处理上传文件的实现。</p> <p>演示：图片上传的实现过程。</p> <p>2. 文件的下载</p> <p>讲解：常见响应消息头，文件下载的实现步骤。</p> <p>演示：下载文件的实现。</p> <p>3. 案例：文件管理器</p> <p>讲解：向学生展示案例的实现效果，分析案例的实现步骤，并演示该案例的实现。</p> <p>三、课堂小结，布置作业</p> <p>对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。</p>				

作业/思考题：实践操作 9。

教学后记：普通上传讲完后，可以讲解一下给上传文件重命名，感觉学生喜欢听，并且可以回顾之前部分函数使用。

教学课次	21	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	图形绘制				
教学目的 与要求	教学目的：掌握画布的创建和输出；掌握文本和图像的绘制。 教学要求：通过验证码案例掌握图像基本函数的使用。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：验证码案例的实现。 教学难点：文本和图像的绘制。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在注册、登录网站时，为了提高网站的安全性，避免用户的恶意攻击，经常需要输入各种各样的验证码，思考下验证码图片的实现思路。</p> <p>(学生回答)</p> <p>老师：验证码图片是通过在图像上绘制文本实现的，GD 库是 PHP 处理图像的内置扩展库，它提供了一系列处理图像的函数。</p> <p>二、学习新知</p> <p>1. 创建画布、设置颜色</p> <p>讲解：PHP 中创建画布的函数以及如何为图像设置颜色，分别讲解创建一个空白画布、基于图片创建画布的方法。</p> <p>演示：创建画布函数的使用，以及为画布设置背景色。</p> <p>2. 输出和销毁图像</p> <p>讲解：将图像输出到浏览器或者保存到文件中的方法，以及在图像资源使用完毕后应该使用 <code>imagedestroy()</code> 销毁图像。</p> <p>演示：将图像输出到浏览器或保存的文件中的方法。</p> <p>3. 绘制文本</p> <p>讲解：PHP 中常用的绘制文本函数，主要讲解通过 <code>imagestring()</code>、<code>imagechar()</code> 和 <code>imagefttext()</code> 的使用。</p> <p>演示：使用不同函数绘制文本的效果。</p> <p>4. 绘制图像</p> <p>讲解：介绍画布中的坐标系统，注意和数学坐标系的区别。讲解常用的图像绘制函数：<code>imagepixel()</code>、<code>imageline()</code>、</p>				

`imagerectangle()`、`imageellipse()`。

演示：图像绘制函数的实现效果。

5. 案例：验证码

讲解：介绍验证码的使用场景，分析常见验证码的组成元素。向学生展示案例的实现效果，分析案例的实现步骤，并演示案例的实现过程。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 各种效果的验证码是如何实现的。

教学后记：本节课有些难度，可以让学生大概了解一下，不用一定完全掌握。会修改验证码效果即可。

教学课次	22	教学 时数	2	教学 课型	<input type="checkbox"/> 理论课 <input checked="" type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	图形的基本操作				
教学目的 与要求	<p>教学目的：掌握图像的基本操作：添加水印、图片缩放、图片旋转等。</p> <p>教学要求：通过案例综合运用图像操作。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：图片缩放、添加水印、相册管理器的实现。</p> <p>教学难点：图片缩放、添加水印。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在展示大量图片的网站中（类似百度图片），经常会使用什么方法减少加载时间？另外，为了确定图片版权一般还会怎么做？</p> <p>（学生回答）</p> <p>老师：将大图片缩放为小图片，可以减少加载时间。为了避免网站中的图片被他人盗用，一般会在图片中添加水印以确定版权。</p> <p>二、学习新知</p> <p>1. 图片缩放</p> <p>讲解：介绍图片缩放的意义，分析实现步骤，讲解图片缩放实现中所涉及函数的使用：<code>getimagesize()</code>、<code>imagecopyresampled()</code>。</p> <p>演示：图片缩放的实现过程。</p> <p>2. 图片剪裁</p> <p>讲解：图片缩放的实现原理，分析实现步骤。</p> <p>演示：图片剪裁的实现过程。</p> <p>3. 添加水印</p> <p>讲解：介绍文字水印和图片水印，分析图片水印的实现步骤，讲解 <code>imagecopy()</code> 函数的使用。</p> <p>演示：添加水印的实现过程。</p> <p>4. 图片旋转和翻转</p> <p>讲解：介绍图片旋转和翻转的区别，分析实现方法，讲解 <code>imagerotate()</code> 函数的使用。</p> <p>演示：图片旋转和翻转的实现过程。</p>				

5.案例：相册管理器

讲解：向学生展示案例的实现效果，分析案例的实现步骤，并演示该案例的实现。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 实践操作 10.5。

教学后记：本节课有些难度，可以让学生大概了解一下，不用一定完全掌握。

。

教学课次	23	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	类的定义				
教学目的 与要求	教学目的：理解面向对象与面向过程的区别；掌握类、对象的定义。 教学要求：掌握构造方法和析构方法；掌握类的封装和继承。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：类的定义、构造方法、封装、继承。 教学难点：方法重写、属性覆盖。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>面向对象编程是一种程序设计思想，它把处理问题需要的相关数据和数据处理方法组合成“对象”、从对象的角度对问题进行建模并解决问题。面向对象编程具有可读性强、易维护、易扩展等优点。本章主要介绍 PHP 中面向对象的封装、继承、多态三大特性，以及抽象类、接口、魔术方法、单例模式和常用类的封装。</p> <p>二、学习新知</p> <p>1. 程序设计方法</p> <p>讲解：面向过程编程是一种以“过程”为中心的程序设计方法，面向过程编程把问题分解为若干过程，每一个过程就是解决问题的一个步骤，把每个步骤按照预定的先后顺序依次执行即可解决问题。采用面向对象的思想进行程序设计时，首先从问题角度出发，把问题分解成若干个对象，每个对象中包含着问题的相关数据和数据处理方法，通过对象之间的交互解决问题。</p> <p>以下棋为例说明两者的区别。</p> <p>2. 类的定义</p> <p>讲解：类是对一组具有相似特征的事物的抽象描述，抽象描述就是把这些事物的共同特征提取出来。例如：人都具有身高、体重、姓名等特征，人还有吃饭、睡觉、工作等行为，因此可以把这些特征和行为抽象出来，构成一个类。</p> <p>PHP 中使用 <code>class</code> 关键字对类进行声明，类名由字母、数字、下划线组成，不能以数字开头，类名不区分大小写。</p> <p>演示：类的定义。</p> <p>讲解：类主要由成员属性和成员方法组成，成员属性用来描述类固有的特征；成员方法用来描述类具有的行为，即类可以做什么事情。例如：员工类包含员工 ID、员工姓名等属性，包含设置 ID、设置姓名等方法。</p>				

演示:通过例 11-1 演示类的定义及成员属性和成员方法的添加。

3. 创建对象

讲解:对象是类的具象化,也称对象为类的实例化。类是抽象的,而对象是具体的。例如:人是一个“类”,但是具体的某一个人则是一个“对象”;“汽车”是一个类,具体的某一辆汽车则是一个“对象”。PHP 中使用 new 运算符创建对象。

演示:通过例 11-2 演示对象的创建及使用。

4. 构造方法和析构方法

讲解:构造方法和析构方法的作用:构造方法用于完成初始化工作,析构方法可以做一些释放资源的工作。构造方法的名称: `__construct()`。

演示:通过例 11-3 演示构造方法的使用。

5. 类的封装

讲解:封装的概念:封装是指通过对现实的事物的共同特征进行分析后,把抽象出的数据和数据操作方法组合起来,形成“类”的过程。封装性使得在类外部的程序不需要知道类的具体实现细节,只要掌握类的使用方法即可。无论类内部如何改变,都不会影响类的使用。

三个访问修饰符 `public`、`private`、`protected` 的作用及使用权限。

6. 类的继承

讲解:继承的概念:继承是指从一个类中派生出一个或多个类。如果一个类 B 继承自另外一个类 A,那么就可以说:类 A 是类 B 的父类,类 B 是类 A 的子类。

继承的作用:子类可以继承父类的属性和方法,从而实现对父类的扩展。

PHP 中子类通过 `extends` 关键字继承父类。

演示:结合例 11-4 讲解如何实现继承。

强调:如果子类没有显式声明构造方法,则子类对象在初始化的时候会自动调用父类的构造方法。

7. 方法重写与属性覆盖

讲解:子类也可以通过在子类内部声明和父类同名的方法或属性,实现对父类方法的重写及对父类属性的覆盖。

演示:结合例 11-5 讲解子类如何重写父类方法和覆盖父类的属性。通过例 11-6 讲解子类在构造方法中调用父类构造方法进行父类初始化。

三、课堂小结,布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 课后习题 1、2、3。

教学后记：本节课有些难度，普通学生大概了解一下，不用一定完全掌握。学习效果好的同学可以深入理解一些。

教学课次	24	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它							
授课内容	类的方法和属性											
教学目的 与要求	教学目的：掌握静态方法、静态属性；掌握抽象类与接口。 教学要求：掌握多态的应用。											
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书											
教学重点 与难点	教学重点：抽象类、接口、多态。 教学难点：多态的应用。											
教 学 过 程 设 计	教 学 内 容											
	<p>一、引入新课</p> <p>老师提问：面向对象的三大特性是什么？ (学生回答)</p> <p>老师：上一节课我们已经学习了类的封装和继承，我们今天将学习类的多态。在学习多态之前，首先让我们来学习一些和多态相关的必要知识，主要包括静态成员、抽象类、接口。</p> <p>二、学习新知</p> <p>1. 静态方法与属性</p> <p>讲解：在面向对象中，有些属性和方法不属于某个具体的对象而属于类，这些属性和方法称为静态属性和静态方法。凡是访问类的地方，都可以访问它的静态方法和属性。</p> <p>在静态方法中不可以访问类的普通属性，但可以访问类的静态属性。在类外部一般通过类名和“::”访问静态成员，不同位置访问静态成员的方法如下表所示。</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <caption>表 11-2 访问静态成员</caption> <thead> <tr> <th>访问位置</th> <th>使用的类名或关键字</th> </tr> </thead> <tbody> <tr> <td>类外部</td> <td>类名</td> </tr> <tr> <td>类内部</td> <td>类名、self、static</td> </tr> <tr> <td>子类访问父类的静态成员</td> <td>parent、static</td> </tr> </tbody> </table> <p>演示：通过例 11-7 演示静态属性和静态方法的用法。</p> <p>2. 抽象类</p>					访问位置	使用的类名或关键字	类外部	类名	类内部	类名、self、static	子类访问父类的静态成员
访问位置	使用的类名或关键字											
类外部	类名											
类内部	类名、self、static											
子类访问父类的静态成员	parent、static											

讲解：抽象类是无法进行实例化的类，它一般用于对类进行更高层次的抽象。PHP 用 `abstract` 关键字定义抽象类，抽象类可以拥有普通的成员属性和成员方法，但至少应该包含一个抽象方法，抽象方法也需要使用 `abstract` 关键字修饰。

演示：通过例 11-8 演示抽象类的定义及应用。

3. 接口

讲解：接口用来提供一组相关的交互功能。接口只能定义功能，但是并不对功能进行具体实现。例如：笔记本电脑生产好后，需要连接鼠标、U 盘等，所以厂家一般会在电脑上预留几个 USB 插槽，这些插槽都需要遵循 USB 接口协议，鼠标厂商如果希望自己生产的鼠标能够适配该笔记本电脑，必须设计出能支持 USB 的接口；同理，U 盘厂商也必须遵循该 USB 接口的协议。因此，接口可以看作交互双方的一个约定，具体的实现由交互实体各自完成即可。PHP 使用 `interface` 关键字定义接口。

演示：通过例 11-9 演示接口的定义及用法。

4. 多态

讲解：多态的概念：多态是指同一个行为应用在不同的对象上会有不同的表现。例如：动物类 `Animal` 拥有一个“发声”的方法，子类 `Dog` 在实现该方法时发出“汪汪汪”的声音，子类 `Cat` 在实现该方法时发出“喵喵喵”的声音。

演示：通过例 11-10 演示多态的应用。

5. 网约车案例

讲解：本例以网约车为例，讲解网约车类、顺风车类、租车类的实现，并通过它们之间的继承关系实现车辆信息显示、费用计算等。

分析成员属性和成员方法：

(1) 成员属性：

对于父类而言，主要包含通用属性即可。网约车一般需要记录车主姓名、车的颜色、汽车类型、车牌号、每次行程的公里数等，实际应用中可能还需要记录车辆总里程数、购买日期、车座数量等。为了简化模型，父类 `NetCar` 中仅考虑车主姓名、汽车颜色、汽车类型、车牌号四个属性属性。

顺风车是网约车的一种形式，顺风车司机为兼职司机。顺风车类一般要考虑起始地点、目的地、出发时间、每公里单价、是否拼车等，此处同样做了简化处理，由于是否拼车决定了不同的计价方式，因此，顺风车包含每公里单价、总路程公里数、是否拼车三个属性。

租车也是网约车的一种形式，租车一般要考虑租车时间、租车单价、租车形式等，常见的租车形式包括按月租和按天租，此处仅考虑按天租的情况。租车类包含单日价格和租车天数两个属性。

(2) 成员方法

本例要完成显示网约车信息和计算用车费用两个功能。由于每个子类都需要显示车辆基本信息，而这些基本信息已经包含在父类的属性中，因此，父类应该包含一个显示车辆信息的方法。由于不同的网约车类型拥有不同的计算费用方法，此处可以构造一个车辆操作接口，并在接口中定义一个计算费用方法，顺风车类和租车类分别通过继承该接口实现各自的费用计算方法。

演示：根据刚才的分析构造网约车类、顺风车类、租车类。并通过构造类的实例进行测试。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 课后习题。

教学后记：主要讲解类的方法和属性。本节课有些难度，普通学生大概了解一下，不用一定完全掌握。学习效果好的同学可以深入理解一些。

教学课次	25	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	面向对象的编程				
教学目的 与要求	<p>教学目的掌握__set()方法和__get()方法的用法；掌握__clone()方法的用法。</p> <p>教学方法：理解__call()方法和__callStatic()方法的用法；掌握__autoload()方法的用法。</p>				
教学方法 与手段	<p>教学方法：LEEPEE 教学法 项目驱动 案例教学</p> <p>手段：多媒体，板书</p>				
教学重点 与难点	<p>教学重点：__set()、__get()、__clone()。</p> <p>教学难点：__call()、__autoload()。</p>				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：面向对象的三大特性是什么？ (学生回答)</p> <p>老师提问：构造一个类时，一般在类的内部需要包含哪些内容？ (学生回答)</p> <p>老师提问：类的成员方法中，有哪些是以两条下划线开头的？ (学生回答)</p> <p>老师：PHP 可以通过实现魔术方法完成特定的功能。魔术方法的名称一般以两条下划线开头，例如：__construct()、__destruct()，这些方法的定义在创建类时默认已经存在，魔术方法的访问修饰符均为 public。</p> <p>二、学习新知</p> <p>1. __set()和__get()方法</p> <p>讲解：__set()用于对那些类中不存在的或不可见的属性进行赋值，该方法包含两个参数：第1个参数代表属性名称，第2个参数代表属性的值。在类外对属性进行访问时，PHP会自动调用__get()方法获取访问权限为 protected、private 的属性的值，以及类中不存在的属性的值。</p> <p>演示：通过例 11-11 演示__set()方法和__get()方法的用法。</p> <p>2. __clone()方法</p> <p>讲解：__clone()方法在使用 clone 关键字进行克隆操作时会自动被调用，主要用于解决对象中特殊属性的复制操作。</p>				

演示：通过例 11-12 演示 `__clone()` 方法的用法。

3. `__call()` 和 `__callStatic()` 方法

讲解：当程序调用类中未声明或没有访问权限的方法时，PHP 会自动调用 `__call()` 方法。该方法包含两个参数，分别为方法名和方法参数。当调用类中未声明或没有访问权限的静态方法时，PHP 会自动调用 `__callStatic()` 方法。`__callStatic()` 方法的参数和 `__call()` 方法相同。

演示：通过例 11-13、11-14 演示 `__call()` 和 `__callStatic()` 方法的用法。

4. `__autoload()` 方法

讲解：`__autoload()` 方法用于批量引入外部类文件。当程序需要引入外部类时，会自动调用 `__autoload()` 方法。该方法包含一个参数，即待加载的类名。`__autoload()` 方法会根据参数中的类名在指定路径下查找类文件，并自动加载该类文件。

演示：通过例 11-15 讲解 `__autoload()` 方法的用法。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 课后习题。

教学后记： 主要讲解面向对象中类的方法。本节课有些难度，普通学生大概了解一下，不用一定完全掌握。学习效果好的同学可以深入理解一些。

。

教学课次	26	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	单例模式				
教学目的 与要求	教学目的：掌握采用单例模式构建类的方法。 教学方法：掌握采用单例模式构建类的方法。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：单例模式 教学难点：单例模式				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：学习完面向对象后，大家觉得面向对象的优点是什么？</p> <p>(学生回答)</p> <p>老师：面向对象具有可维护、可复用、可扩展的优点，它最强大的地方在于：随着业务越来越复杂，面向对象依然能够保持程序结构良好，而面向过程则可能使程序越来越臃肿。面向对象之所以能够做到这一点，最主要的一个原因就是设计模式。</p> <p>二、学习新知</p> <p>1. 设计模式</p> <p>讲解：软件设计模式，也称为设计模式，是一套被反复使用、多数人知晓的、经过分类编目的、代码设计经验的总结。设计模式可以看作是针对特定环境下某一类问题给出的解决方案，它往往是经过无数人的反复验证，从而总结出的一套技术方案。使用设计模式是为了可重用代码、让代码更容易被他人理解，保证代码可靠性、程序的重用性。</p> <p>2. 单例模式的概念</p> <p>讲解：单例模式的概念：单例模式是指一个类只能拥有一个实例对象的设计模式。在某些情况下，希望某个类在整个项目中只创建一个对象，例如：系统需要有一个配置类来存储某些配置信息，如果该类可以多次进行实例化，则无法知晓具体是哪个类对象修改了配置文件，最好的方式是只让该类实例化一次，使用单例模式可以很方便的解决这个问题。此外，PHP 是服务端脚本语言，PHP 程序中一般会包含大量数据库操作，使用单例模式可以避免重复创建数</p>				

数据库连接对象。

3. 构造单例模式

讲解：根据单例模式的特点，逐一分析如何构造单例模式。要想构造单例模式，可通过以下三个步骤：

1) 私有化构造方法

在类外部实例化类对象时，一般要通过 `new` 运算符创建对象，此时一定会调用构造方法。如果把构造方法设为私有，则在类外部无法通过 `new` 运算符创建对象，从而强制该类无法从外部进行实例化。

2) 私有化克隆方法

除了通过 `new` 运算符获得类的对象外，还可以通过 `clone` 关键字克隆对象。克隆对象时一定会调用 `__clone()` 方法，因此，将 `__clone()` 方法设置为私有可以避免对象被克隆。

3) 定义静态方法

在私有化构造方法和克隆方法之后，无法通过构造方法或者克隆方法创建类的对象，而类的非静态方法必须由对象调用，因此只能通过静态方法获得类的对象。为了避免外部类修改实例对象，需要添加一个私有静态属性来保存类的实例，这样外部类就无法访问该实例对象。演示：通过例 11-16 构造单例模式。

4. 单例模式的代码实现

演示：结合例 11-16 演示单例模式的构建过程。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：单例模式的代码实现结合例 11-16 演示单例模式的构建过程。

教学后记：主要介绍单例模式。本节课有些难度，普通学生大概了解一下，不用一定完全掌握。学习效果好的同学可以深入理解一些。

教学课次	27	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	数据库操作类				
教学目的 与要求	教学目的：掌握数据库操作类。 教学方法：掌握文件上传类。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：数据库操作类。 教学难点：对类进行封装。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：在 PHP 操作数据库章节，我们已经学习过使用 MySQLi 扩展操作数据库的方法。现在，我们学习过面向对象的知识后，能不能使用面向对象的方式操作数据库呢？</p> <p>(学生回答)</p> <p>老师：由于面向对象的封装性、继承性和多态性，使得采用面向对象思想设计的程序具有易复用、易维护、易扩展等特点。本节主要讲解数据库操作类、文件上传类的封装和使用。</p> <p>二、学习新知</p> <p>1. 数据库操作类</p> <p>讲解：前面章节已经学习过使用 MySQLi 扩展提供的数据库操作函数进行增、删、改、查的基本方法，这些方法在使用时常常需要重复书写，不便于在多个项目中进行复用。为了便于访问数据库，可以把访问数据库的操作封装成一个数据库操作类。</p> <p>分析需要完成的功能：PHP 在操作数据库时，一般包含连接数据库、设置字符集、执行 SQL 语句、查询数据库表并返回数据和关闭数据库连接等操作。</p> <p>分析成员属性：在访问不同的数据库时，使用的主机名、用户名、密码、数据库名称、字符集编码等也不尽相同，这些参数需要在创建对象时动态传递进来，因此，可以把这些基本信息作为数据库操作类的私有成员属性。</p> <p>在实际使用数据库操作类的时候，如果每次都对其进行实例化，则每个数据库对象都需要和数据库连接一次。为了避免频繁创建数据库操作对象、降低服务器压力，本例采用单例模式构建数据库操作类，因此类中需要有一个私有静态实例属性。此外，在设置字符</p>				

集、执行 SQL 语句等操作时，都需要用到数据库连接对象，因此把数据库连接对象也设置为类的属性。

分析成员方法：为了实现连接数据库、设置字符集、执行 SQL 语句、查询数据库表并返回数据和关闭数据库连接等操作，数据库操作类需要给每个操作设计一个对应的方法。主要包括：构造方法、初始化属性方法、连接数据库方法、设置字符集方法、执行 SQL 语句方法、查询单条数据方法、查询多条数据的方法、关闭数据库连接的方法。

演示：通过例 11-17 对数据库操作类的实现过程进行演示，并对数据库操作类进行测试。

2. 文件上传类

讲解：在提交表单时，有时需要进行文件上传操作。前面章节已经学习过借助超全局变量 `$_FILES` 实现上传文件的方法，为了便于在不同项目中复用文件上传程序，可以把文件上传封装成类。本节主要介绍文件上传类的设计及实现。

分析完成文件上传的步骤：当表单中有文件需要上传时，PHP 脚本一般需要经过获取文件信息、出错时给出提示、判断文件类型和大小、移动临时文件至指定位置等步骤完成上传操作。接下来分析文件上传类的成员属性和成员方法。

分析成员属性：为了避免频繁创建文件上传类的对象，文件上传类也采用单例模式进行设计，因此该类中必须包含一个私有静态实例对象。在文件上传时，需要对上传路径、文件最大字节数等进行设定，因此类中需要包含文件大小、文件大小的上限等属性。服务器对上传文件一般采用统一的命名规则，本例在命名时用到了文件前缀、文件后缀、原名称等属性，最后生成文件的新名称。此外，移动上传文件时还需要用到临时文件，因此也把上传的临时文件作为类的一个属性。

分析成员方法：为了实现文件上传，需要依次完成判断上传路径是否可用、获取文件信息、检验文件大小、检验文件类型、移动文件至指定目录等过程。因此，需要对上述过程分别设计对应的方法，并依次在 `upload()` 方法中调用。程序在获得 `UploadFile` 类的实例后，只需要调用 `upload()` 方法即可实现文件上传。

演示：通过例 11-18 演示文件上传类的封装，并对文件上传类进行测试。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题：课后习题。

教学后记：本章介绍了数据库操作类。本节课有些难度，是之前数据库操作的扩展。主要针对学习效果的同学。

教学课次	28	教学 时数	2	教学 课型	<input checked="" type="checkbox"/> 理论课 <input type="checkbox"/> 实训课 <input type="checkbox"/> 实验课 <input type="checkbox"/> 习题课 <input type="checkbox"/> 其它
授课内容	验证码类				
教学目的 与要求	教学目的：掌握验证码类。 教学方法：掌握验证码类。				
教学方法 与手段	教学方法：LEEPEE 教学法 项目驱动 案例教学 手段：多媒体，板书				
教学重点 与难点	教学重点：验证码类的实现。 教学难点：对验证码类的封装。				
教 学 过 程 设 计	教 学 内 容				
	<p>一、引入新课</p> <p>老师提问：上次课我们学习了数据库操作类和文件上传类的封装及使用。大家觉得在封装类的过程中，最重要的事情是什么？最困难的又是什么？</p> <p>(学生回答)</p> <p>老师：经过上次课的学习，大家基本已经掌握了封装类的方法。本节课我们继续来学习验证码类的封装。</p> <p>二、学习新知</p> <p>1. 验证码类</p> <p>讲解：在网站上进行表单数据提交时，为了保护网络安全、避免机器暴力破解数据密码等危害，一般会在表单提交之前要求用户输入验证码进行验证。前面章节已经学习过使用 GD 库中的函数生成验证码。为了便于对生成验证码的代码进行复用，可以把生成验证码的过程封装成类。本节主要讲解验证码类的封装过程。</p> <p>分析验证码生成的步骤：生成验证码一般包含创建图像、生成验证码字符串、绘制验证码、绘制干扰点、绘制干扰线条、显示图片等步骤。接下来分析验证码类中需要包含的成员属性和成员方法。</p> <p>分析成员属性：在生成验证码图片时，需要指定验证码的宽度、高度、字符个数、字符类型、干扰点数目、干扰线数目等信息，这些均作为验证码类的属性。在绘制图像、绘制线条等操作中都需要用到图像对象，在绘制字符串、保存到 session 中时，均需要用到验证码字符串，所以把图像对象和验证码字符串也设置为该类的属性。</p> <p>分析成员方法：根据验证码生成的步骤，验证码类中应该包含创建图像、生成验证码字符串、绘制验证码、绘制干扰点、绘制干扰线条、显示图片等方法。</p> <p>演示：通过例 11-19 演示验证码类的封装，并对验证码类进行测试。</p>				

试。

三、课堂小结，布置作业

对这节的主要内容进行总结，布置作业。最后留几分钟让学生和老师或者同学之间互相讨论，加深课程内容印象，及时提出问题、解决问题。

作业/思考题： 课后习题 11。

教学后记：本章介绍了验证码类，可以多做个验证码的效果，如汉字、计算等，感觉学生挺感兴趣的。