

《细说PHP》

图书基本信息

书名：《细说PHP》

13位ISBN编号：9787121185632

10位ISBN编号：7121185636

出版时间：2012-10

出版社：电子工业出版社

作者：高洛峰

页数：810

版权说明：本站所提供下载的PDF图书仅提供预览和简介以及在线试读，请支持正版图书。

更多资源请访问：www.tushu000.com

内容概要

《细说PHP(第2版)》共六个部分，分为30个章节，每一章都是PHP独立知识点的总结。内容涵盖了动态网站开发的前台技术（HTML+CSS）、PHP编程语言的语法、PHP的常用功能模块和实用技巧、MySQL数据库的设计与应用、PHP面向对象的程序设计思想、数据库抽象层PDO、Smarty模板技术、Web开发的设计模式、自定义框架BroPHP、Web项目开发整个流程等目前PHP开发中最主流的技术。每一章中都有大量的实用示例，以及详尽的注释，加速读者的理解和学习，也为每章的技术点设置了大量的自测试题。最后以一个比较完整的、采用面向对象思想，以及通过MVC模式设计，并结合Smarty模板，基于BroPHP框架的CMS系统为案例，详细介绍了Web系统开发从设计到部署的各个细节，便于更好地进行开发实践。

作者简介

现任北京LAMP兄弟连（易第优教育）教学总监，五年软件开发工作经验，四年IT教学工作经验。曾在多家上市企业担任过技术总监、项目经理、高级软件工程师等职务，具有扎实的技术功底。国内资深的高级技术认证讲师，从事IT培训以来累计授课长达6000课时以上，培训近千名学员成功走向IT岗位，具有丰富的IT教学经验。熟悉掌握Unix/Linux/Windows等多种操作系统，一直从事嵌入式、C\C++、J2EE、J2ME、J2SE、PHP及Oracle数据库等技术研发工作。其中，对Web系统软件开发深有研究，近年来致力于推广LAMP开源平台系列软件。

书籍目录

第1部分 web开发入门篇

第1章 lamp网站构建

1.1 介绍网站给你认识

1.1.1 web应用的优势

1.1.2 web 2.0时代的互联网

1.1.3 web开发标准

1.1.4 认识脚本语言

1.1.5 了解http协议

1.2 动态网站开发所需的web构件

1.2.1 客户端浏览器

1.2.2 超文本标记语言html

1.2.3 层叠样式表css

1.2.4 客户端脚本编程语言javascript

1.2.5 web服务器

1.2.6 服务器端编程语言

1.2.7 数据库管理系统

1.3 几种主流的web应用程序平台

1.3.1 web应用程序开发平台对比分析

1.3.2 动态网站开发平台技术比较

1.4 web的工作原理

1.4.1 情景1：服务器不带应用程序服务器和数据库

1.4.2 情景2：带应用程序服务器的web服务器

1.4.3 情景3：浏览器访问服务器端的数据库

1.5 lamp网站开发组合概述

1.5.1 linux操作系统

1.5.2 web服务器apache

1.5.3 mysql数据库管理系统

1.5.4 php后台脚本编程语言

1.5.5 lamp发展趋势

1.6 学php需要学习什么内容

1.6.1 学php之前的准备

1.6.2 学php时需要了解或掌握的内容

1.6.3 优秀的web程序员是怎样练成的

1.7 小结

本章必须掌握的知识点

本章需要了解的内容

第2章 html的设计与应用

2.1 网页制作概述

2.1.1 html基础

2.1.2 简单html实例制作

2.2 html语言的语法

2.2.1 html标签和元素

2.2.2 html语法不区分字母大小写

2.2.3 html标签属性

2.2.4 html颜色值的设置

2.2.5 html文档注释

2.2.6 html代码格式

- 2.2.7 html字符实体
- 2.3 html文件的主体结构
- 2.4 html文档头部元素[head]
 - 2.4.1 [title]元素
 - 2.4.2 [base]元素
 - 2.4.3 [link]元素
 - 2.4.4 [meta]元素
- 2.5 html文档主体标记
- 2.6 文字版面的编辑
 - 2.6.1 格式标签
 - 2.6.2 文本标签
- 2.7 创建图像和链接
 - 2.7.1 插入图片
 - 2.7.2 建立锚点和超链接
- 2.8 使用html表格
- 2.9 html框架结构
- 2.10 html表单设计
- 2.11 小结
- 本章必须掌握的知识点
- 本章需要了解的内容
- 本章需要拓展的内容
- 本章的学习建议
- 第3章 层叠样式表css
 - 3.1 css简介
 - 3.2 css规则的组成
 - 3.2.1 css注释
 - 3.2.2 长度单位
 - 3.2.3 颜色单位和url值
 - 3.3 在html文档中放置css的几种方式
 - 3.3.1 内联样式表
 - 3.3.2 嵌入一个样式表
 - 3.3.3 连接到一个外部的样式表
 - 3.4 css选择器
 - 3.4.1 html选择器
 - 3.4.2 类选择器
 - 3.4.3 id选择器
 - 3.4.4 关联选择器
 - 3.4.5 组合选择器
 - 3.4.6 伪元素选择器
 - 3.5 css常见的样式属性和值
 - 3.5.1 字体属性
 - 3.5.2 颜色属性
 - 3.5.3 背景属性
 - 3.5.4 文本属性
 - 3.5.5 边框属性
 - 3.5.6 鼠标光标属性
 - 3.5.7 列表属性
 - 3.5.8 综合示例
 - 3.6 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

本章的学习建议

第4章 div+css网页标准化布局

4.1 div+css对页面布局的优势

4.2 “无意义”的html元素div和span

4.3 w3c盒子模型

4.4 和页面布局有关的css属性

4.5 盒子区块框的定位

4.5.1 相对定位

4.5.2 绝对定位

4.6 使用盒子模型的浮动布局

4.6.1 设置浮动

4.6.2 行框和清理

4.7 div+css的兼容性问题

4.7.1 不同浏览器解释盒子模型的差异

4.7.2 设置浏览器去遵循w3c标准

4.8 使用盒子模型设计页面布局

4.8.1 居中设计

4.8.2 设置两列浮动的布局

4.8.3 设置三列浮动的布局

4.8.4 设置多列浮动的布局

4.9 div+css网站首页面布局实例

4.9.1 html文件的设计

4.9.2 css文件设计

4.10 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

本章的学习建议

第2部分 php基础篇

第5章 从搭建你的php开发环境开始

5.1 几种常见的php环境安装方式

5.1.1 linux系统下源代码包方式安装环境

5.1.2 在windows系统上安装web工作环境

5.1.3 搭建学习型的php工作环境

5.2 环境安装对操作系统的选择

5.2.1 选择网站运营的操作系统

5.2.2 选择网站开发的操作系统

5.3 安装集成php开发环境

5.3.1 安装前准备

5.3.2 安装步骤

5.3.3 环境测试

5.4 phpmyadmin的配置与应用

5.4.1 http身份验证模式

5.4.2 cookie身份验证模式

5.4.3 config身份验证模式

5.5 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第6章 php的基本语法

6.1 php在web开发中的应用

6.1.1 就从认识php开始吧

6.1.2 php都能做什么

6.2 第一个php脚本程序

6.3 php语言标记

6.3.1 将php代码嵌入html中的位置

6.3.2 解读开始和结束标记

6.4 指令分隔符“分号”

6.5 程序注释

6.6 在程序中使用空白的处理

6.7 变量

6.7.1 变量的声明

6.7.2 变量的命名

6.7.3 可变变量

6.7.4 变量的引用赋值

6.8 变量的类型

6.8.1 类型介绍

6.8.2 布尔型 (boolean)

6.8.3 整型 (integer)

6.8.4 浮点型 (float或double)

6.8.5 字符串 (string)

6.8.6 数组 (array)

6.8.7 对象 (object)

6.8.8 资源类型 (resource)

6.8.9 null类型

6.8.10 伪类型介绍

6.9 数据类型之间相互转换

6.9.1 自动类型转换

6.9.2 强制类型转换

6.9.3 类型转换细节

6.9.4 变量类型的测试函数

6.10 常量

6.10.1 常量的定义和使用

6.10.2 常量和变量

6.10.3 系统中的预定义常量

6.10.4 php中的魔术常量

6.11 php中的运算符

6.11.1 算术运算符

6.11.2 字符串运算符

6.11.3 赋值运算符

6.11.4 比较运算符

6.11.5 逻辑运算符

6.11.6 位运算符

6.11.7 其他运算符

6.11.8 运算符的优先级

6.12 表达式

6.13 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第7章 php的流程控制结构

7.1 分支结构

7.1.1 单一条件分支结构（if）

7.1.2 双向条件分支结构（else从句）

7.1.3 多向条件分支结构（elseif子句）

7.1.4 多向条件分支结构（switch语句）

7.1.5 巢状条件分支结构

7.1.6 条件分支结构实例应用（简单计算器）

7.2 循环结构

7.2.1 while语句

7.2.2 do...while循环

7.2.3 for语句

7.3 特殊的流程控制语句

7.3.1 break语句

7.3.2 continue语句

7.3.3 exit语句

7.4 小结

本章必须掌握的知识点

本章需要了解的内容

第8章 php的函数应用

8.1 函数的定义

8.2 自定义函数

8.2.1 函数的声明

8.2.2 函数的调用

8.2.3 函数的参数

8.2.4 函数的返回值

8.3 函数的工作原理和结构化编程

8.4 php变量的范围

8.4.1 局部变量

8.4.2 全局变量

8.4.3 静态变量

8.5 声明及应用各种形式的php函数

8.5.1 常规参数的函数

8.5.2 伪类型参数的函数

8.5.3 引用参数的函数

8.5.4 默认参数的函数

8.5.5 可变个数参数的函数

8.5.6 回调函数

8.6 递归函数

8.7 使用自定义函数库

8.8 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第9章 php中的数组与数据结构

9.1 数组的分类

9.2 数组的定义

9.2.1 直接赋值的方式声明数组

9.2.2 使用array () 语言结构新建数组

9.2.3 多维数组的声明

9.3 数组的遍历

9.3.1 使用for语句循环遍历数组

9.3.2 使用foreach语句遍历数组

9.3.3 联合使用list ()、each () 和while循环遍历数组

9.3.4 使用数组的内部指针控制函数遍历数组

9.4 预定义数组

9.4.1 服务器变量：\$_server

9.4.2 环境变量：\$_env

9.4.3 url get变量：\$_get

9.4.4 http post变量：\$_post

9.4.5 request变量：\$_request

9.4.6 http文件上传变量：\$_files

9.4.7 http cookies：\$_cookie

9.4.8 session变量：\$_session

9.4.9 global变量：\$globals

9.5 数组的相关处理函数

9.5.1 数组的键/值操作函数

9.5.2 统计数组元素的个数和唯一性

9.5.3 使用回调函数处理数组的函数

9.5.4 数组的排序函数

9.5.5 拆分、合并、分解和接合数组

9.5.6 数组与数据结构

9.5.7 其他有用的数组处理函数

9.6 操作php数组需要注意的一些细节

9.6.1 数组运算符

9.6.2 删除数组中的元素操作

9.6.3 关于数组下标的注意事项

9.7 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第10章 php面向对象的程序设计

10.1 面向对象的介绍

10.1.1 类和对象之间的关系

10.1.2 面向对象的程序设计

10.2 如何抽象一个类

10.2.1 类的声明

10.2.2 成员属性

10.2.3 成员方法

10.3 通过类实例化对象

10.3.1 实例化对象

10.3.2 对象类型在内存中的分配

10.3.3 对象中成员的访问

- 10.3.4 特殊的对象引用“\$this”
- 10.3.5 构造方法与析构方法
- 10.4 封装性
 - 10.4.1 设置私有成员
 - 10.4.2 私有成员的访问
 - 10.4.3 __set ()、__get ()、__isset () 和__unset () 四个方法
- 10.5 继承性
 - 10.5.1 类继承的应用
 - 10.5.2 访问类型控制
 - 10.5.3 子类中重载父类的方法
- 10.6 常见的关键字和魔术方法
 - 10.6.1 final关键字的应用
 - 10.6.2 static关键字的使用
 - 10.6.3 单态设计模式
 - 10.6.4 const关键字
 - 10.6.5 instanceof关键字
 - 10.6.6 克隆对象
 - 10.6.7 类中通用的方法__toString ()
 - 10.6.8 __call () 方法的应用
 - 10.6.9 自动加载类
 - 10.6.10 对象串行化
- 10.7 抽象类与接口
 - 10.7.1 抽象类
 - 10.7.2 接口技术
- 10.8 多态性的应用
- 10.9 面向对象版图形计算器
 - 10.9.1 需求分析
 - 10.9.2 功能设计及实现
 - 10.9.3 类的组织架构
- 10.10 小结
- 本章必须掌握的知识点
- 本章需要了解的内容
- 本章需要拓展的内容
- 第11章 字符串处理
 - 11.1 字符串的处理介绍
 - 11.1.1 字符串的处理方式
 - 11.1.2 字符串类型的特点
 - 11.1.3 双引号中变量解析总结
 - 11.2 常用的字符串输出函数
 - 11.3 常用的字符串格式化函数
 - 11.3.1 去除空格和字符串填补函数
 - 11.3.2 字符串大小写的转换
 - 11.3.3 和html标签相关的字符串格式化
 - 11.3.4 其他字符串格式化函数
 - 11.4 字符串比较函数
 - 11.4.1 按字节顺序进行字符串比较
 - 11.4.2 按自然排序进行字符串比较
 - 11.5 小结
- 本章必须掌握的知识点

本章需要拓展的内容

第12章 正则表达式

12.1 正则表达式简介

12.1.1 选择php正则表达式的处理函数库

12.2 正则表达式的语法规则

12.2.1 定界符

12.2.2 原子

12.2.3 元字符

12.2.4 模式修正符

12.3 与perl兼容的正则表达式函数

12.3.1 字符串的匹配与查找

12.3.2 字符串的替换

12.3.3 字符串的分割和连接

12.4 文章发布操作示例

12.5 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要扩展的内容

第3部分 php常用功能模块篇

第13章 php的错误和异常处理

13.1 错误处理

13.1.1 错误报告级别

13.1.2 调整错误报告级别

13.1.3 使用trigger_error () 函数来替代die ()

13.1.4 自定义错误处理

13.1.5 写错误日志

13.2 异常处理

13.2.1 异常处理实现

13.2.2 扩展php内置的异常处理类

13.2.3 捕获多个异常

13.3 小结

本章必须掌握的知识点

本章需要了解的内容

第14章 php的日期和时间

14.1 unix时间戳

14.1.1 将日期和时间转变成unix时间戳

14.1.2 日期的计算

14.2 在php中获取日期和时间

14.2.1 调用getdate () 函数取得日期/时间信息

14.2.2 日期和时间格式化输出

14.3 修改php的默认时区

14.4 使用微秒计算php脚本执行时间

14.5 日历类

14.6 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

本章的学习建议

第15章 文件系统处理

- 15.1 文件系统概述
 - 15.1.1 文件类型
 - 15.1.2 文件的属性
- 15.2 目录的基本操作
 - 15.2.1 解析目录路径
 - 15.2.2 遍历目录
 - 15.2.3 统计目录大小
 - 15.2.4 建立和删除目录
 - 15.2.5 复制目录
- 15.3 文件的基本操作
 - 15.3.1 文件的打开与关闭
 - 15.3.2 写入文件
 - 15.3.3 读取文件内容
 - 15.3.4 访问远程文件
 - 15.3.5 移动文件指针
 - 15.3.6 文件的锁定机制
 - 15.3.7 文件的一些基本操作函数
- 15.4 文件的上传与下载
 - 15.4.1 文件上传
 - 15.4.2 处理多个文件上传
 - 15.4.3 文件下载
- 15.5 设计经典的文件上传类
 - 15.5.1 需求分析
 - 15.5.2 程序设计
 - 15.5.3 文件上传类代码实现
 - 15.5.4 文件上传类的应用过程
- 15.6 小结
- 本章必须掌握的知识点
- 本章需要了解的内容
- 本章需要拓展的内容
- 本章的学习建议
- 第16章 php动态图像处理
 - 16.1 php中gd库的使用
 - 16.1.1 画布管理
 - 16.1.2 设置颜色
 - 16.1.3 生成图像
 - 16.1.4 绘制图像
 - 16.1.5 在图像中绘制文字
 - 16.2 设计经典验证码类
 - 16.2.1 设计验证码类
 - 16.2.2 应用验证码类的实例对象
 - 16.2.3 表单中应用验证码
 - 16.2.4 实例演示
 - 16.3 php图片处理
 - 16.3.1 图片背景管理
 - 16.3.2 图片缩放
 - 16.3.3 图片裁剪
 - 16.3.4 添加图片水印
 - 16.3.5 图片旋转和翻转

16.4 设计经典的图像处理类

16.4.1 需求分析

16.4.2 程序设计

16.4.3 图像处理类代码实现

16.4.4 图像处理类的应用过程

16.5 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第4部分 数据库开发篇

第17章 mysql数据库概述

17.1 数据库的应用

17.1.1 数据库在web开发中的重要地位

17.1.2 为什么php会选择mysql作为自己的黄金搭档

17.1.3 php和mysql的合作方式

17.1.4 结构化查询语言sql

17.2 mysql数据库的常见操作

17.2.1 mysql数据库的连接与关闭

17.2.2 创建新用户并授权

17.2.3 创建数据库

17.2.4 创建数据表

17.2.5 数据表内容的简单管理

17.3 小结

本章必须掌握的知识点

第18章 mysql数据表的设计

18.1 数据表 (table)

18.2 数据值和列类型

18.2.1 数值类的数据列类型

18.2.2 字符串类数据列类型

18.2.3 日期和时间型数据列类型

18.2.4 null值

18.2.5 类型转换

18.3 数据字段属性

18.4 数据表对象管理

18.4.1 创建表 (create table)

18.4.2 修改表 (alter table)

18.4.3 删除表 (drop table)

18.5 数据表的类型及存储位置

18.5.1 myisam数据表

18.5.2 innodb数据表

18.5.3 如何选择innodb还是myisam表类型

18.5.4 数据表的储存位置

18.6 数据表的默认字符集

18.6.1 字符集

18.6.2 字符集支持原理

18.6.3 创建数据对象时修改字符集

18.7 创建索引

18.7.1 主键索引 (primary key)

18.7.2 唯一索引 (unique)

18.7.3 常规索引 (index)

18.7.4 全文索引 (fulltext)

18.8 规范化

18.8.1 起点

18.8.2 第一范式

18.8.3 第二范式

18.8.4 第三范式

18.8.5 规范化理论

18.9 数据库的设计技巧

18.9.1 数据库设计要求

18.9.2 起名字的技巧

18.9.3 数据库具体设计工作中的技巧

18.10 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第19章 sql语句设计

19.1 操作数据表中的数据记录 (dml)

19.1.1 使用insert语句向数据表中添加数据

19.1.2 使用update语句更新数据表中已存在的数据

19.1.3 使用delete语句删除数据表中不需要的数据记录

19.2 通过dql命令查询数据表中的数据

19.2.1 选择特定的字段

19.2.2 使用as子句为字段取别名

19.2.3 distinct关键字的使用

19.2.4 在select语句中使用表达式的列

19.2.5 使用where子句按条件检索

19.2.6 根据空值 (null) 确定检索条件

19.2.7 使用between and进行范围比较

查询

19.2.8 使用in进行范围比对查询

19.2.9 使用like进行模糊查询

19.2.10 多表查询 (连接查询)

19.2.11 嵌套查询 (子查询)

19.2.12 使用order by对查询结果排序

19.2.13 使用limit限定结果行数

19.2.14 使用统计函数

19.2.15 使用group by对查询结果分组

19.3 查询优化

19.4 小结

本章必须掌握的知识点

本章需要拓展的内容

本章的学习建议

第20章 php访问mysql的扩展函数

20.1 php访问mysql数据库服务器的流程

20.2 在php脚本中连接mysql服务器

20.2.1 在php程序中选择已创建的数据库

20.2.2 执行sql命令

20.2.3 在php脚本中处理select查询结果集

20.3 设计完美分页类

20.3.1 需求分析

20.3.2 程序设计

20.3.3 完美分页类的代码实现

20.3.4 分页类的应用过程

20.4 管理books表实例

20.4.1 需求分析

20.4.2 程序设计

20.5 php的mysqli扩展介绍

20.5.1 启用mysqli扩展模块

20.5.2 mysqli扩展接口的应用概述

20.6 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

本章的学习建议

第21章 数据库抽象层pdo

21.1 pdo所支持的数据库

21.2 pdo的安装

21.3 创建pdo对象

21.3.1 以多种方式调用构造方法

21.3.2 pdo对象中的成员方法

21.4 使用pdo对象

21.4.1 调整pdo的行为属性

21.4.2 pdo处理php程序和数据库之间的数据类型转换

21.4.3 pdo的错误处理模式

21.4.4 使用pdo执行sql语句

21.5 pdo对预处理语句的支持

21.5.1 了解pdostatement对象

21.5.2 准备语句

21.5.3 绑定参数

21.5.4 执行准备好的查询

21.5.5 获取数据

21.5.6 大数据对象的存取

21.6 pdo的事务处理

21.6.1 mysql的事务处理

21.6.2 构建事务处理的应用程序

21.7 小结

本章必须掌握的知识点

本章需要了解的内容

本章需要拓展的内容

第5部分 php开发高级篇

第22章 memcache管理与应用

22.1 memcache概述

22.1.1 初识memcache

22.1.2 memcache在web中的应用

22.2 memcached的安装及管理

22.2.1 linux下安装memcache软件

- 22.2.2 windows下安装memcached软件
- 22.2.3 memcached服务器的管理
- 22.3 使用telnet作为memcached的客户端管理
 - 22.3.1 连接memcached服务器
 - 22.3.2 基本的memcached客户端命令
 - 22.3.3 查看当前memcached服务器的运行状态信息
 - 22.3.4 数据管理指令
- 22.4 php的memcached管理接口
 - 22.4.1 安装php中的memcache应用程序扩展接口
 - 22.4.2 memcache应用程序扩展接口
 - 22.4.3 memcache的实例应用
- 22.5 memcached服务器的安全防护
- 22.6 小结
- 本章必须掌握的知识点
- 本章需要了解的内容
- 本章需要拓展的内容
- 第23章 会话控制
 - 23.1 为什么要使用会话控制
 - 23.2 会话跟踪的方式
 - 23.3 cookie的应用
 - 23.3.1 cookie概述
 - 23.3.2 向客户端计算机中设置cookie
 - 23.3.3 在php脚本中读取cookie的资料内容
 - 23.3.4 数组形态的cookie应用
 - 23.3.5 删除cookie
 - 23.3.6 基于cookie的用户登录模块
 - 23.4 session的应用
 - 23.4.1 session概述
 - 23.4.2 配置session
 - 23.4.3 session的声明与使用
 - 23.4.4 注册一个会话变量和读取session
 - 23.4.5 注销变量与销毁session
 - 23.4.6 session的自动回收机制
 - 23.4.7 传递session id
 - 23.5 一个简单的邮件系统实例
 - 23.5.1 为邮件系统准备数据
 - 23.5.2 编码实现邮件系统
 - 23.5.3 邮件系统执行说明
 - 23.6 自定义session处理方式
 - 23.6.1 自定义session的存储机制
 - 23.6.2 使用数据库处理session信息
 - 23.6.3 使用memcached处理session信息
 - 23.7 小结
 - 本章必须掌握的知识点
 - 本章需要了解的内容
 - 本章需要拓展的内容
- 第24章 php的模板引擎smarty
 - 24.1 什么是模板引擎
 - 24.2 自定义模板引擎

- 24.2.1 自定义模板引擎类
 - 24.2.2 使用自己的模板引擎
 - 24.2.3 应用自定义模板引擎的示例分析
 - 24.3 选择smarty模板引擎
 - 24.4 安装smarty及初始化配置
 - 24.4.1 安装smarty
 - 24.4.2 初始化smarty类库的默认设置
 - 24.4.3 第一个smarty的简单示例
 - 24.5 smarty的基本应用
 - 24.5.1 php程序员常用和smarty相关的操作
 - 24.5.2 模板设计时美工的常用操作
 - 24.6 smarty模板设计的基本语法
 - 24.6.1 模板中的注释
 - 24.6.2 模板中的变量应用
 - 24.6.3 模板中的函数应用
 - 24.6.4 忽略smarty解析
 - 24.7 在smarty模板中的变量应用
 - 24.7.1 从配置文件中读取变量
 - 24.7.2 在模板中使用保留变量
 - 24.8 在smarty模板中的变量调解器
 - 24.8.1 变量调解器函数的使用方式
 - 24.8.2 smarty默认提供的变量调解器
 - 24.8.3 自定义变量调解器插件
 - 24.9 smarty模板中自定义函数
 - 24.9.1 为smarty模板扩充函数插件
 - 24.9.2 为smarty模板扩充块函数插件
 - 24.10 smarty模板中的内置函数
 - 24.10.1 变量声明
 - 24.10.2 流程控制
 - 24.10.3 声明和调用模板函数
 - 24.10.4 数组遍历
 - 24.10.5 smarty提供的其他内置函数
 - 24.11 smarty的模板继承特性
 - 24.11.1 使用{extends}函数实现模板继承
 - 24.11.2 在子模板中覆盖父模板中的部分内容区域
 - 24.11.3 合并子模板和父模板的{block}标签内容
 - 24.12 smarty的缓存控制
 - 24.12.1 在smarty中控制缓存
 - 24.12.2 每个模板多个缓存
 - 24.12.3 为缓存实例消除处理开销
 - 24.12.4 清除缓存
 - 24.12.5 关闭局部缓存
 - 24.13 小结
 - 本章必须掌握的知识点
 - 本章需要了解的内容
 - 本章需要拓展的内容
- ## 第25章 mvc模式与php框架
- 25.1 mvc模式在web中的应用
 - 25.1.1 mvc模式的工作原理

- 25.1.2 mvc模式的优缺点
- 25.2 php开发框架
 - 25.2.1 什么是框架
 - 25.2.2 为什么要用框架
 - 25.2.3 框架和mvc设计模式的关系
 - 25.2.4 比较流行的php框架
- 25.3 划分模块和操作
 - 25.3.1 为项目划分模块
 - 25.3.2 为模块设置操作
- 第26章 超轻量级php框架brophp
 - 26.1 brophp框架概述
 - 26.1.1 系统特点
 - 26.1.2 环境要求
 - 26.1.3 brophp框架源码的目录结构
 - 26.2 单一入口
 - 26.2.1 基于brophp框架的单一入口编写规则
 - 26.3 部署项目应用目录
 - 26.3.1 项目部署方式
 - 26.3.2 url访问
 - 26.4 brophp框架的基本设置
 - 26.4.1 默认开启
 - 26.4.2 配置文件
 - 26.4.3 内置函数
 - 26.5 声明控制器 (control)
 - 26.5.1 控制器的声明 (模块)
 - 26.5.2 操作的声明
 - 26.5.3 页面跳转
 - 26.5.4 重定向
 - 26.6 设计视图 (view)
 - 26.6.1 视图与控制器之间的交互
 - 26.6.2 切换模板风格
 - 26.6.3 模板文件的声明规则
 - 26.6.4 display () 用新用法
 - 26.6.5 在模板中的几个常用变量应用
 - 26.6.6 在php程序中定义资源位置
 - 26.7 应用模型 (model)
 - 26.7.1 brophp数据库操作接口的特性
 - 26.7.2 切换数据库驱动
 - 26.7.3 声明和实例化model
 - 26.7.4 数据库的统一操作接口
 - 26.8 自动验证
 - 26.9 缓存设置
 - 26.9.1 基于memcached缓存设置
 - 26.9.2 基于smarty的缓存机制
 - 26.10 调试模式
 - 26.11 内置扩展类库
 - 26.11.1 分页类page
 - 26.11.2 验证码类vcode
 - 26.11.3 图像处理类image

- 26.11.4 文件上传类fileupload
- 26.12 自定义功能扩展
 - 26.12.1 自定义扩展类库
 - 26.12.2 自定义扩展函数库
- 26.13 小结
- 本章必须掌握的知识点
- 本章需要了解的内容
- 第6部分 项目开发篇
- 第27章 b/s结构软件开发流程
 - 27.1 软件开发过程的划分
 - 27.2 需求开发
 - 27.2.1 需求分析流程
 - 27.2.2 需求分析说明
 - 27.2.3 输出
 - 27.3 系统设计
 - 27.3.1 系统设计流程
 - 27.3.2 系统设计说明
 - 27.4 编码测试
 - 27.4.1 编码与测试流程
 - 27.4.2 编码说明
 - 27.4.3 结果测试说明
 - 27.5 试运行
 - 27.5.1 软件试运行流程
 - 27.5.2 软件试运行说明
 - 27.6 实施
 - 27.6.1 软件实施流程
 - 27.6.2 软件实施说明
 - 27.7 验收
 - 27.7.1 软件验收流程
 - 27.7.2 软件验收说明
 - 27.7.3 验收标准
 - 27.8 服务与维护
 - 27.8.1 责任人
 - 27.8.2 收集信息
 - 27.8.3 维护分析
 - 27.8.4 软件维护
 - 27.8.5 改进
 - 27.8.6 输出
 - 27.9 项目管理
 - 27.9.1 软件项目的计划
 - 27.9.2 软件项目的组织
 - 27.9.3 项目小组组织形式
 - 27.10 项目参考
- 第28章 需求分析说明书
 - 28.1 文档介绍
 - 28.1.1 编写说明
 - 28.1.2 项目背景
 - 28.1.3 读者对象
 - 28.1.4 参考资料

- 28.1.5 术语与缩写解释
- 28.2 任务概述
 - 28.2.1 产品的描述
 - 28.2.2 系统目标
 - 28.2.3 系统功能结构
 - 28.2.4 系统流程图
- 28.3 业务描述
 - 28.3.1 后台登录管理
 - 28.3.2 后台操作界面管理
 - 28.3.3 常规管理
 - 28.3.4 公告管理
 - 28.3.5 友情链接管理
 - 28.3.6 相册管理
 - 28.3.7 图片管理
 - 28.3.8 栏目管理
 - 28.3.9 文章管理
 - 28.3.10 幻灯片管理
 - 28.3.11 用户组管理
 - 28.3.12 用户管理
 - 28.3.13 前台首页管理
 - 28.3.14 栏目列表管理
 - 28.3.15 文章内容管理
 - 28.3.16 文章搜索管理
 - 28.3.17 登录注册管理
 - 28.3.18 个人空间管理
 - 28.3.19 消息管理
 - 28.3.20 动态管理
- 28.4 系统运行环境
 - 28.4.1 硬件环境
 - 28.4.2 软件环境
- 28.5 需求设计评审
- 第29章 数据库设计说明书
 - 29.1 引言
 - 29.1.1 编写目的
 - 29.1.2 背景
 - 29.1.3 定义
 - 29.1.4 参考资料
 - 29.2 外部设计
 - 29.2.1 标识符和状态
 - 29.2.2 使用它的程序
 - 29.2.3 约定
 - 29.2.4 支持软件
 - 29.3 结构设计
 - 29.4 逻辑结构设计
 - 29.4.1 er图向关系模型的转化
 - 29.4.2 确定关系模式
 - 29.4.3 消除冗余
 - 29.5 物理结构设计
 - 29.5.1 设计数据表结构

- 29.5.2 创建数据表
- 29.5.3 数据表记录的输入
- 29.6 安全保密设计
 - 29.6.1 完整性
 - 29.6.2 数据库设计的其他问题
- 第30章 程序设计说明书
 - 30.1 引言
 - 30.1.1 编写目的
 - 30.1.2 背景
 - 30.1.3 定义
 - 30.1.4 使用技术
 - 30.1.5 参考资料
 - 30.2 系统的结构
 - 30.2.1 项目的目录结构
 - 30.2.2 模块结构
 - 30.2.3 程序结构
 - 30.3 用户管理模块设计说明
 - 30.3.1 功能
 - 30.3.2 流程逻辑
 - 30.3.3 接口
 - 30.3.4 存储分配
 - 30.3.5 注释设计
 - 30.3.6 限制条件
 - 30.3.7 测试计划
 - 30.3.8 尚未解决的问题
 - 30.3.9 获取添加用户的界面操作add ()
 - 30.3.10 用户数据入库的操作insert ()
 - 30.3.11 查询用户列表操作index ()
 - 30.3.12 获取修改用户的界面操作mod ()
 - 30.3.13 用户数据修改的操作update ()
 - 30.3.14 删除用户操作del ()
- 附录
 - 附录a 编码规范
 - a.1 绪论
 - a.1.1 适用范围
 - a.1.2 目标
 - a.1.3 开发工具
 - a.2 php的文件格式
 - a.2.1 php开始和结束标记
 - a.2.2 注释规范
 - a.2.3 空行和空白
 - a.2.4 字符串的使用
 - a.2.5 命名原则
 - a.2.6 语言结构
 - a.2.7 其他规范细节
 - a.3 mysql设计规范
 - a.3.1 数据表的设计
 - a.3.2 索引设计原则
 - a.3.3 sql语句设计

a.4 模板设计

附录b php的安全和优化

b.1 网站安全security

b.1.1 安全配置php

b.1.2 隐藏配置细节

b.1.3 隐藏敏感数据

b.1.4 清理用户数据

b.1.5 数据加密

b.2 网站优化optimize

b.2.1 php脚本级优化

b.2.2 使用代码优化工具

b.2.3 缓存加速

b.2.4 http加速

b.2.5 启用gzip内容压缩

附录c、附录d见本书光盘

章节摘录

版权页：插图：2.生成动态网页 PHP脚本程序和客户端的JavaScript脚本程序不同的是，PHP代码是运行在服务端的。PHP脚本程序可以根据用户在客户端的不同输入请求，在服务端运行该脚本后，动态输出用户请求的内容。客户端就能接收到想得到的结果，但无法得知其背后的代码是如何运作的。甚至可以将Web服务器设置成让PHP来处理所有的HTML文件，这么一来，用户就无法得知服务端到底做了什么。

3.字符串处理 在编写程序代码或是文本处理时，经常在操作字符串，所以字符串处理一直是程序员使用最多的技术之一，PHP是把字符串作为一种基本的数据类型来处理的。在PHP中提供了丰富的字符串处理函数，并使用强大的正则表达式来对字符串或文本进行搜索、查找、匹配、替换等操作。

4.动态输出图像 使用PHP并不局限于输出HTML文本。PHP通过使用GD扩展库还能用来动态输出图像，例如文字按钮、验证码、数据统计图等，还可以轻松地编辑图像，例如处理缩略图和为图片添加水印等，具有强大的图像处理功能。

5.处理服务器端文件系统 要想让数据可以长期保留，可以使用数据库或是文件系统来存取信息。在某些存取数据相对简单的应用中，或是一些特定的应用中，没有必要使用数据库，就可以采用文件操作。PHP可以利用文件系统函数任意操作服务器中的目录或文件。包括目录或文件的打开、编辑、复制、创建、删除，以及文件属性等操作。

6.编写数据库支持的网页 PHP最强大、最显著的特性之一，是它支持很大范围的数据库。用户会发现利用PHP编写数据库支持的网页简单得难以置信。目前，PHP可以连接任何支持世界标准的数据库。

7.会话跟踪控制 我们访问Web服务器通常是使用HTTP协议完成的，但它是一个无状态的协议，没有一个内建机制来维护两个事务之间的状态。也就是当一个用户在请求一个页面后再请求另外一个页面时，HTTP无法告诉我们这两个请求是来自N——+NP。所以可以在PHP中使用会话控制思想在网站中跟踪一个用户，这样就可以很容易地做到用户登录的支持，并根据某个用户的授权级别和个人喜好显示相应的内容，也可以根据会话控制记录该用户的行为。

《细说PHP》

媒体关注与评论

本书内容充实，是作者多年丰富PHP授课经验的总结，为初学者首选教材。——腾讯开放平台副总经理/腾讯云负责人 陈磊
敏捷高效开发是PHP的一大特性，从书中掌握设计思想，学而实习，必将获益匪浅。——盛大云北京研发部经理/PHPCMS负责人 王参加
此书对于PHP爱好者，可谓一本好书，内容详尽实用，结构组织符合实战需求，我诚挚推荐。——北京4399轻电商团队技术总监 张宴
内容详尽，由浅入深，覆盖全面，藏经之作。——Think PHP框架创始人 刘晨

《细说PHP》

编辑推荐

《细说PHP(第2版)》获得51CTO “最受读者喜爱的原创IT技术图书”奖。

《细说PHP》

名人推荐

本书内容充实，是作者多年丰富PHP授课经验的总结，为初学者首选教材。——腾讯开放平台副总经理，腾讯云平台负责人 陈磊 敏捷高效开发是PHP的一大特性，从书中掌握设计思想，学而时习之，必将获益匪浅。——盛大云北京研发部经理，PHPcMs负责人 王参加 此书对于PHP爱好者，可谓一本好书，内容详尽实用，结构组织符合实战需求，我诚挚推荐。——北京4399轻电商团队技术总监 张宴 内容详尽，由浅入深，覆盖全面，藏经之作。——ThinkPHP框架创始人 刘晨

精彩短评

- 1、初学php，听说这本书不错，买下之后希望最大化利用他~
- 2、有许多内容在光盘里，书里不太详细。适合有一定基础的学习者使用。
- 3、学习PHP必须的
- 4、这本书正如51cto所评价的一样，内容详实，站在初学者的立场上讲解问题，这很好。初学者往往这也学那也学，学到最后很迷茫，为什么？就是因为不系统，零散的知识点形成不了知识体系，这很不好，永远觉得思想上提不到一定的高度，悬在半空，往下下不来，往上上不去，这种感觉很是不爽。现在好了，有了这本书，不仅知识细节详实，而且对于web开发的体系有个整体的把握。是本不可多得的好书。拿到手后，真是爱不释手！在此我也向所有进军web开发的有志之士推荐此书。当然好的也有不完美的，就是光盘内容视频只是一部分，其他的需要有网络，从网上下载，着实有些不方便。
- 5、还行吧，容易理解内容充实。有些小错误，花了几个月读的
- 6、以前过看过细说PHP第一版，感觉第二版与第一版差别不大，不过对于初学PHP的人很实用。
- 7、兄弟连的教材，很不错，值得推荐。
- 8、细说PHP（第2版）（含DVD光盘1张）（？
- 9、作者多年经验总结，PHP入门的一本好书
- 10、现在正在读大三，决定要学PHP于是果断买了这本书。书就用当当网的袋子装着，袋子很薄，没有其他的包装，有点失望。但是只要书好就行。这么厚的一本书真要好好下功夫学习啊。我是学通信的，又要学习专业课，还要抽出时间学习这本书，真的要勤奋了。。
- 11、10月5日就预定了，23日发货，25日收到，这3天实在是太漫长，每天不停的刷新看快递的状态。书拿到手后第一时间撸了一管庆祝一下，内容丰富，涵盖了PHP的所有重要知识点，810页只要81块钱，已经很超值了。支持高老师，支持兄弟连！
- 12、细说php2 让你成为屌丝程序员
- 13、适合有html基础的新手学习php
- 14、很好的php入门书籍，很适合初学者用。书很厚，里面的内容也很全，总之非常喜欢。还有我觉得高老师讲课讲的真心不错，简单易懂。
- 15、细说php，细细的研究。
- 16、资料很详细，也很全面，可以做个PHP的百科全书了，不过偶尔会遇到错别字，亲们仔细学习吧！
- 17、打开一看书的一部分都被压褶了，一看就是发货的时候没有仔细包装造成的，在当当买就是为了买新书买正版，结果由于你们的失误整的还不如买二手书呢，书的内容确实很好，内容很详细，很值得买啊，努力学习中···
- 18、前端也介绍了一点，作为零基础入门非常合适，包括了面向过程和面向对象，感觉非常好，思路也很清楚。
- 19、php绝对的好书，值得一看
- 20、差不多是一本完全无基础学的人学习PHP的好书，毕竟还是需要多理解HTML DIV+CSS之后才进入PHP
- 21、之前一直看兄弟连的教程，不错，书也不错
- 22、还可以吧
- 23、还好吧，比较适合初学者
- 24、拿到这本书大致的看了一下，书很大很厚很重，含金量不错，前4章还包含了html\css\div基础知识，挺照顾刚接触web开发新手的，光盘内容也充实
- 25、很好，比前面买的几本书详细，一些细致的地方都有讲解
- 26、正在学习中，涵盖的内容算比较完整。。。
- 27、泛泛之谈
- 28、质量不错 就是初级和高级都在一起 好厚的
- 29、很全面，很多有用的PHP重点知识,PHP初学者必备！
- 30、研究了好几天终于决定买下了这本书，拿到书，比我想象中的还要厚，可以挡子弹了，哈哈，不

过书的内容很详细，很好，很值得看

- 31、早就想买这本书了，一直追随lamp，无兄弟，不编程。
- 32、这本书非常适合作为php手册使用，这样才能当字典查，太厚，略重
- 33、因为有c语言和asp的基础，看起来很快，一周看完了，尝试做了个小的留言板和带后台的新闻系统，php确实比asp好用多了，很强大，书太好了！强烈推荐看看
- 34、php专业书籍，看网上推荐所以选择这个，内容很详细
- 35、介绍的非常详细，很不错的宝贝，非常赞
- 36、php入门书籍，没有任何编程经验的人可以参看，有编程经验的人可能会觉得书上的内容可能太基本，不太适合。书总体来讲不错，810页，带一张光盘。
- 37、收到书了感觉不错！不管书怎样，主要还是自己要坚持学习，有耐心，看了书有很多不很懂，在网上也下载了兄弟连的视频，书和视频结合学习，相信慢慢会上手的！
- 38、一直期待高老师的这本书，拿到手立即翻看，知识点比一版更新全面很多，书价的确不低，但内容也够实在，共800多页，还送一张光盘，核算起来性价比非常合适，必须给好评！
- 39、非常详细的一本书，工作学习必备！
- 40、简单的总结：这只是一个工具书，一本包含PHP应该会的工具的基础框架的书。

如果你是有学过编程基础的，这本书不仅浪费钱而且浪费时间。但是，要说一句，对于自学的没受过学院训练的或者是作为了解的话还是可以的，因为不是讲的多到位，而是语言足够通俗。

- 41、正在学习中，感觉还不错，挺详细的，再配上免费的视频，自学足够了
- 42、嗯，很详细，最近我一直在用来当枕头，可以作为一本手册查询使用
- 43、买了一个月，看了一半了。非常好的入门书籍，讲得很细。就算没有Web编程经验，也能学懂。希望能出进阶的书籍。
- 44、书很厚，实用，很详细！
- 45、细说PHP第一版的时候就已经很不错了，现如今第二版更加出色。推荐。
- 46、我与我同学一共买了4本,等了半个多月,预定啊预定就是得预定~~~~~当书拿到手里的那一刹那,我擦,真tm厚,心里一想,书厚点好,值这个钱,毕竟知道高老师每次都是手敲代码挺辛苦的.反正有了这本书,我不泡妞了,我不玩游戏了,我不失眠了,我不害怕半路回家被人爆菊花了.....俩字安全!!!!!!!!!!!!!!
- 47、很不错，很详细。不过第一次买就发错光盘给我，不过换了。
- 48、送货速度快，书很实用，很不错！
- 49、第一版翻烂了，继续翻第二版，比第一版厚了很多，内容也比第一版多了，感觉更加精细，没白等，堪称熔学习、防身、健身于一体的PHP学习神器
- 50、很好，基本覆盖了PHP的内容，视频也还是不错的，兄弟连的网站有很多视频资源，很好
- 51、PHP入门的经典，内容详实易懂，全面涵盖了web开发的所需的知识内容。是一本从入门到精通PHP技术的首选教材。
- 52、本人在传智播客，四拨子那个地，8.14那个班韩老师不错，我不是托，不信你们来传智播客四拨子试听几节课吧！免费的哦！感谢小韩老师把我们带进门了 接下来修行靠个人了 这本书我一致在关注 几次去兄弟连总部了 也没买到，没想到当当网上早就开始卖了！接下来修炼开始了，祝广大php爱好者早日修的成果！
- 53、看了200多页了，初中级比较实用，讲解深入浅出，例子也切题容易理解。配书光盘还没看，内容暂不评价。
- 54、书到手了，粗略的翻了翻，图文并茂，讲的很详细。很好很强大，非常适合新手入门学习。准备接下来的时间慢慢品味其中那曲折动人的“故事”！
- 55、内容详实，结构合理，非常适合于PHP初学者学习参考。
- 56、全面覆盖PHP学习关键点,还不错~
- 57、还可以，同学推荐的，因为项目需要看了一点。附带有视频，讲解清晰，但是节奏太慢了，有时间的话配合着视频还是比较好理解的。后来忙起来就没时间看了，有机会继续读下去。
- 58、这书作者写了一年半，花了四天时间学完.....汗。几年前曾经读过别的php教材，后来没参与这类的项目就荒废了，这本算是复习。本书内容全面，很详细，虽然刚看过鸟哥的书，比较喜欢那种轻松近人的风格，不过这本书中规中矩读着也不错。

- 59、作用不是很大。
- 60、书对基础 和高级知识讲解的很详细感觉对我帮助很大
- 61、这本书太好啦，结合者兄弟连官方出的15G的教学视频，一起学习了。
- 62、现在作为我的php入门必备教程之选，实在是非常详细与实用
- 63、内容很全面 所有知识点都讲解透彻 值得一看~
- 64、很实用的一本书，讲的比较详细，适合我这种初学者学习。
- 65、内容照第一版充实了很多。只看了目录，还没详细看内容呢，打算当手册的补充来看。嗯，高胖子的表达能力有问题，第一版反正看说的东西基本是看不懂的。也不知道第二版都有没有改进？不过无所谓啦，我们主要看代码咯。
- 66、很早就想买了，趁着这次折上折，终于购入。这段时间一直在看它的电子书和视频。我想，中国的软件产业如果像兄弟连这样，不是利益至上，而是想要干一番自己的事业，那么早就腾飞了。
- 67、视频内容很详细，书也很好，推荐！！
- 68、好，内容实用详细。
- 69、讲的很通俗易懂。绝对适合新手入门
- 70、内容讲解详细，全面，简易程度适中
- 71、不错的php书籍不错的php书籍不错的php书籍不错的php书籍不错的php书籍不错的php书籍
- 72、想在闲暇时学学PHP，这本书相对来说比较适合新手。
- 73、php入门书籍，比市面上的各种从入门到精通或者24课时之类的好一些，不过也就是本工具书，恕我浅薄，看着看着就厌倦了。
- 74、很好的一本书，不管是入门还是提升都可以用这本书，很系统
- 75、嘟嘟，速度很快呀，昨天下的单今天下午就来了，书大致翻了下，很有质感，学习中。。。。。
- 76、当初啃过。祭奠下
- 77、系统有条理，值得一读
- 78、很好，很细致，很全面，学习php的利器。
- 79、书的内容很充实，实战性强。之前看过LAMP兄弟连的免费视频，书的内容很契合视频，整体而言，适合PHP程序猿，也适合初学者，及对WEB开发有兴趣的人员。
- 80、真心不错的书，看了视频来购买的 配套的视频很不错 有很多可以学习的代码 比如网上商城呀 bbs呀 cms呀。。还有，。。。。。
- 81、这本书真的不错，很系统，而且容易看。
- 82、内容丰富详实，专业性强，受益匪浅，不过代码的字体比较小，建议后续版本将字体放大一点好些
- 83、技术的书，还没看，希望别空洞
- 84、内容很详细，是学习PHP一本非常好用的工具书
- 85、书的质量很好，里面讲解的很详细，很容易就能看懂，附带的光盘内容也很丰富。
- 86、学习PHP就得看《细说PHP》
- 87、学习PHP比较全面和基础的一本书，跟着学习就可以做一些简单的系统了！！
- 88、这本书还是不错的，视频教程很给力，一边看高洛峰一边看张恩民学php就是好
- 89、书比较好，送货速度也可以。
- 90、适合初学者很多项目经验
- 91、书中自有黄金屋，书中自有颜如玉，每本书都有他值得评品读的地方。当当是我购书最多的地方，满分
- 92、靠这本书入了门，非计算机相关专业
- 93、通篇采用面向对象的思维和方法，感觉上很容易理解，他所讲的面向对象..特别是在构造行数和西沟函数所讲的那几句话，顿时明朗了，对OOP编程..现在，我都在慢慢的尝试使用面向对象编程的思维来设计程序了..值得购买，是一本工具类书记...
- 94、好厚，讲得好详细，后面还有软件工程的相关实践内容。
- 95、之前看兄弟连的视频，挺好的，论坛里面的朋友推荐的这本书，写的真细，大部分能看明白，一直关注他们的论坛，跟大家分享一下他们最近拍的一个短片，挺逗的<http://event.weibo.com/654632>

《细说PHP》

96、老实说书不错。但由于光盘一张，视频有大部分要官网网上下载。这本书本来就比较贵了，建议还是2张高容量DVD，把所有的视频放进来最好。

97、还在阅读中。每天读一章，帮助蛮大的

98、学习php有这一本足以！

99、细说PHP一直是我的案头书，第一版我当手册用着，我是看第一版的视频学会PHP的，高老师讲得真的不错，这次听说更新了我特地买来收藏的，也是对高老师的一个支持，虽然高老师从来没有见过我，但是我一直把高老师当作教会我PHP的恩师，不知道兄弟连离我远不远，有时间我一定登门拜访！

100、如果你是学习PHP的(自学)，不知道php兄弟连和PHP100的话，那么您就白学了。强烈推荐此书

101、看中书的作者才买的,比官网要便宜一点点,配有大量的视频资料,这点很重要.

章节试读

1、《细说PHP》的笔记-第一章至第八章、第十六章

2.1PHP基本语法

功能执行语句后面一定要加分号

结构定义语句后面一定不要加分号

2.2PHP中使用变量

如果在用到数据时，需要用到多次。就声明为变量使用

\$变量名=值

PHP是弱类型语言，变量的类型有存储的值决定

isset(); 判断变量是否存在

unset(); 删除并释放变量

变量的命名规则

1、变量前一定要使用\$，声明和使用都要有\$

2、不能以数字开头

3、不能使用PHP的运算符+ - * % / & .

4、PHP可以使用系统关键字作为变量名

5、PHP变量区分大小写（只有变量和常量区分）

6、变量名称一定要有意义，驼峰命名法

可变变量：一个变量的变量名可以动态设置和使用

变量的引用赋值：使用一个&符号加到将要赋值的变量（只能是变量）前面，即可采用引用赋值

变量的类型

PHP中8中类型

4中标量:整型 int integer 布尔型 bool boolean 浮点型 float double real 字符串 string

2中复合类型 数组array 对象 object

2种特殊类型 资源类型 resource 空类型 null

var_dump(变量或值); //既可以查看变量或值的类型，也可以看到数据

各种类型变量如何声明

字符串声明中最好使用单引号，双引号会耗更多资源

定界符号声明大量字符串,hello自定义的，里面可以任意解析变量 反义字符

```
$str = &lt;&lt;&lt;hello
```

```
hello;
```

第五章

变量：

5.3.1

在php中变量的声明必须是使用一个\$符号，后面跟变量名来表示

unset()函数释放指定变量

isset()函数检测变量是否设置

empty()函数检查一个变量是否为空

5.3.2

变量命名严格区分大小写，但内置结构和关键字以及用户自定义的类名和函数名都是不区分大小写
变量名有字母或者下划线开头，后面跟上任意数量的字母，数字，或者下划线。

5.3.4

变量的引用赋值，意味着新的变量只是简单的引用了原始变量（并没有执行赋值操作），因而改动新

的变量会影响到原始变量。

使用引用赋值，简单的将&符号添加到将要赋值的变量前

```
$bar = &$foo;
```

只有有名字的变量才可以引用赋值

5.4.1

PHP支持8中原始数据类型

4种标量类型 boolean integer float string

两种复合类型 array object

两种特殊类型 resource null

如果想查看某个表达式的值和类型，可以使用函数var_dump()

八进制和十六进制的整型表达式的计算结果均以十进制数字输出

5.4.5 字符串

字符串有单引号 双引号 定界符三种方法定义

单引号中不能再包含单引号，可添加反斜线转义

单引号中的变量不会被变量的值替代，而是将变量名原样输出

双引号最重要的一点是其中的变量名会被变量值替代，即PHP可以解析双引号中包含的变量

```
定界符<&&&EOT*****
```

```
EOT;
```

5.4.6数组

使用print_r()函数查看数组中的全部内容

5.4.7对象

要初始化一个对象，用new语句将对象实例化到一个变量中

5.4.9 null

null值表示一个变量没有值，不区分大小写

将变量直接赋值位null

声明的变量尚未被赋值

被unset()函数销毁的变量

5.4.11数据类型转换

只有相同类型的数据才能彼此操作，自动转换要遵循按数据长度增加的方向进行

强制类型转换需要早转换的变量之前加上用括号括起来的目标类型，也可以使用转换函数

intval()获取变量的整数值

floatval() 获取变量的浮点值

strval() 获取变量的字符串值

以上强制类型转换都没有改变被转换变量本身的类型，而是通过转换将得到新类型数据赋值给新变量。如果需要将变量本身的类型改变成其他类型可使用

setType()函数

想得到一个易读懂的类型的表达方式用于调试可以使用gettype()函数

查看某个类型用is_type()函数

5.5常量的定义和使用

在PHP中是通过使用define()函数来定义的，其命名规则与变量相似，按照惯例常量标识符总是大写，

不要\$符号

常量和变量的不同点

- 1.常量前没有\$
- 2.常量只能用define()函数定义，而不能通过赋值语句
- 3.常量可以再任意地方定义和使用，一旦定义就不能被重新定义或者取消定义
- 4.常量只能是标量

5.5.3

预定义常量不区分大小写

5.6.1

在做求模运算时首先会将%运算符两边操作数转化为整型，然后返回第一个操作数除以第二个操作数后所得到的余数。在程序开发中一般用于计算闰年和一个范围内的随机数

5.6.7

三元运算符

(expr1)?(expr2):(expr3)

在expr1求值为true时执行expr2并获取其值，为false时执行expr3并获取其值

执行运算符反引号（` `）PHP尝试将反引号中的内容作为操作系统命令来执行，并将其输出信息返回。使用反引号的效果与函数shell_exec()相同

第六章PHP语言结构

6.1流程控制

顺序结构 语句按照出现的先后次序自上而下依次执行

分支结构 先做判断再做选择

6.3.4特殊的流程控制语句

break语句结束当前for、foreach、while、do-while、switch结构循环的执行

continue语句作用是跳过该次循环，继续执行下一次循环结构

exit语句作用是直接退出当前脚本

6.4PHP中的函数

函数的声明 可以没有参数列表 没有返回值

函数的调用 在需要使用函数的位置，使用函数名称和参数列表进行调用，执行完毕后返回调用位置继续向下执行

函数的返回值是函数执行后的结果返回给调用者

return语句可以向函数调用者返回函数体中任意确定的值，将程序控制权反回到调用者的作用域，即退出函数

如果要返回多个值，可以将多个值添加到一个数组中，然后返回这个数组

PHP变量的范围

局部变量 其作用域仅限于函数内部。不仅在函数中声明的变量是局部的，函数参数在函数名后面的括号内声明也是局部变量，当函数执行完毕后，函数内部的变量都被释放。如果在函数外部需要使用该变量值时，必须通过return语句

全局变量 作用域从变量定义开始，到本程序文件末尾

注意：在PHP函数中无法直接调用全局变量

若要使用全局变量，必须通过global关键字定义目标变量，以告诉函数此变量为全局变量。
还可以在PHP中使用预定义的全局变量数组\$GLOBALS

局部变量可分为动态存储类型（默认）和静态存储类型（函数执行完毕后，内部变量依旧保存在内存中）

6.4.5参数的传递

PHP中参数默认是按值传递，即使在函数内部改变参数的值，他并不会改变函数外部的值

按引用传递是把该数值或变量的内存存储区块相对地址导入函数之中，因此当该数值在函数中有任何变动时，会连带对父程序造成影响。

可以在函数定义中在参数前预先加上&

6.4.6变量函数

将函数名赋给变量，在程序中使用变量名并在后面加上圆括号时就调用那个函数执行

第七章PHP中的数组及数组结构

数组的分类

索引数组：下标是顺序的整数作为索引

关联数组：下标是字符串作为索引

PHP中下标只有整数 字符串两种类型

数组的多种声明方式

1.直接为数组元素赋值

print_r()或var_dump()函数可打印数组中所有元素的内容

如果索引下标不给出，默认会就从0开始索引

如果给出索引下标，下一个索引将会是最大索引数增1

如果后面出现前面的下标，如果是赋值则属于对前面的元素重新赋值

混合声明时，索引和关联不互相影响

2.使用array()函数

默认是索引数组

如果声明关联数组，必须使用“=>”运算符指定字符串下标

多个成员之间使用逗号分隔

3.使用其他的函数声明file()

7.3数组的遍历

1.使用for语句循环遍历数组（要求数组的下标必须是连续的数字索引）

其他语言（只有这一种方式）

PHP中这不是首选方式

2.使用foreach()语句循环遍历数组

```
foreach(数组变量 as 变量值){
```

```
    //循环体
```

```
}
```

循环次数由数组的元素个数决定

每一次循环都会将数组中的元素分别赋值给后面的变量

3.while()、list()、each()组合循环遍历数组

each()函数

需要一个数组作为参数

返回值也是一个数组，1 value 0 key 4个下标的形式排列
默认当前元素就是第一个元素
每执行一次后就会将当前元素后移

list()函数

list()函数需要将一个数组赋值给这个函数
数组中的元素个数，要和list()函数中参数个数相同
数组中每个元素值会赋值给list()函数的每个参数，list()将参数转为变量
list()只能接受索引数组，按索引下标的顺序排列

使用数组内部指针控制函数

next(数组) prev(数组) reset(数组) end(数组) current(数组) key(数组)

预定义数组 自动全局变量 超全局数组

\$_GET 由URL请求提交至脚本的变量
\$_POST 由http post提交至脚本的变量
\$_REQUEST 由get post cookie机制提交至脚本的变量 尽量不使用
\$_FILES 由http post文件上传而提交至脚本的变量
\$_COOKIE
\$_SESSION 会话控制
\$_ENV 执行环境提交至脚本的变量
\$_SERVER 变量由WEB服务器设定的或直接和当前脚本执行环境相关联
\$GLOBALS 只要是当前脚本有效地变量都包含在内

数组中相关处理函数

键值操作相关函数

array_values()

array_keys()

in_array()

array_flip 交换数组中的键和值

统计数组个数和唯一性函数

count() sizeof()

array_count_values() 统计数组中所有值出现的次数

使用回调函数处理数组中的函数

array_filter()

array_walk()

array_map()

数组的排序函数

简单的数组排序 根据元素字符串大小排序

sort() rsort()

根据键名对数组排序

krsort() ksort()

根据元素的值对数组排序

asort() arsort()

根据自然数排序法对数组排序

natsort() natcasesort()

根据用户自定义规则对数组排序

usort() uasort() uksort()

多维数组的排序

array_multisort()

拆分、合并、分解、接合

array_slice()

array_splice()

array_combine()

array_merge()

array_intersect()

array_diff()

数组与数据结构的函数

1使用数组实现堆栈功能

array_push() 压入

array_pop() 弹出

2实现队列功能

array_unshift() 入列

array_pop() 出列

array_shift() 删除数组第一个

unset()从中间删除

其他与数据操作相关的函数

array_rand()

shuffle()

array_sum()

range()

第八章PHP面向对象的程序设计

对象的三大特性：封装 继承 多态

面向对象和面向过程之间的区别

最小的单位：函数 面向过程

最小的单位：对象 面向对象

什么是对象

对象的抽象结果就是类

一个你能看到的实体，

包含成员属性==》变量==定义对象的外观和状态

成员方法==》函数==定义对象的功能的

什么是类

类描述了一组有相同特性（属性）和相同行为（方法）的对象的集合，类的实例化结果就是对象

在类中定义属性和方法

类和对象的关系

声明类==》实例化类==》用对象

8.2.1类的声明

```
[修饰词]class 类名{
```

```
    对象的成员属性：//对类的静态描述，可以是php中标量类型和复合类型
```

```
    对象的成员方法：//对类的动态描述
```

```
}
```

类名和变量名、函数名命名规则相似

在类中声明成员属性时一定要使用关键字如public private static var

一个文件只保存一个类，文件名中包含类名 一般是类名.class.php

变量和函数：驼峰式

常量：全大写

类名：首字符大写 一定要有意义

通过类实例化对象

只有对象才在存储空间

使用的是new新建一个对象，\$对象引用=new 类名；

创建一个对象就在内存中分配了空间

成员方法的声明和函数完全一样，只不过可以加一些关键字控制成员方法的权限

8.3通过类实例化对象

\$变量名 = new 类名称([参数列表]) //独享实例化格式

在php中只要使用一次new关键字就会实例化出一个对象，并在堆内存中开辟一块自己的空间，每个对象之间都是相互独立的。

内存结构 大体上分为4断

栈空间段 空间小但被CPU访问的速度快。存放程序中临时创建的变量 后进先出的特点 用来保存和恢复调用现场 int bool doub

堆空间段 存放程序运行中被动态分配的内存段，存储数据长度可变或占用内存较大 string array object

初始化数据段 存放程序静态分配的变量

代码段 用来存放可执行文件的操作指令 只准读取不准写入 function

\$引用名 = new 类名称([参数列表])

echo \$引用名 -> 成员属性; //获取属性值

\$引用名 -> 成员属性 = 值; //对属性赋值操作

\$引用名 -> 成员方法; //访问成员方法

对象一旦被创建，在对象的每个成员方法中都会存在一个特殊的对象引用\$this,专门用来发完成对象内部成员之间的访问

构造方法

对象创建完成后第一个被对象自动调用的方法，常用来执行一些初始化任务

构造方法名称必须是(PHP5) __construct()

同一个类只能声明一个构造方法

析构方法 __destruct()不能带有任何参数

对象被销毁前自动调用的方法，执行关闭文件，释放结果等操作，在垃圾回收程序回收对象之前调用

对象在内存中的分配

对象的使用

对象中的成员必须通过对象的引用来访问

在对象中的方法中都默认有一个\$this关键字，这个关键字代码调用这个方法的对象，相当于第一人称代词我

构造方法（构造器）

是对象创建完成以后“第一个自动调用”的方法

在PHP4中和类名相同的方法就是构造方法

在PHP5中构造方法选择使用魔术方法__construct()所有类中声明构造方法都使用这个，优点：在改变类名时，构造方法不用改变

作用：就是为成员属性初始化

魔术方法：方法名称都是系统提供的，加上这个就启用

__construct() __clone()克隆对象使用clone关键字，自动调用，\$this代表副本对象，\$that代表原本对象

`__destruct()` `__call()`在调用对象中不存在的方法时自动调用，处理一些不存在方法的错误调用
`__set()` `__sleep()`
`__get()` `__wakeup()`
`__isset()` `__toString()`直接输出对象引用时自动调用，用来快速获取对象的字符串最快捷方式
`__unset()` `__autoload()`这是唯一一个不再类中添加的方法，只要在页面中使用类，就会自动将类名传递个这个参数

析构方法

当对象被释放之前“最后一个自动调用”的方法
作用是关闭资源，做清理工作

封装性：面向对象的三大特性之一

封装性就是把对象的成员属性和成员方法组合成一个独立的相同单位，并尽可能隐藏内部细节

`private` 私有的 只能在对象内部访问（`$this`）

作用：使用`private`修饰，其只能在内部使用

成员属性的封装

只要一个变量需要在多个方法中使用，就将这个方法声明为成员属性

使用`private`关键字将成员属性和方法封装成私有属性，只能在内部使用

封装后经过公有方法可以对属性进行取值、赋值控制

在对象内部声明一些操作私有属性的公有方法，这样在对象外部就可以通过公有的方法作为访问接口，间接的访问对象内部私有成员

和封装有关的魔术方法：

`__set()`:直接设置成员属性值时，自动调用的方法

`__get()`：直接获取成员属性值时

`__isset()`：直接`isset()`查看对象中私有属性是否存在时自动调用的方法

`__unset()`:直接`unset()`删除对象中私有属性时自动调用的方法

继承性：

父类--基类 子类--派生类

开放性 可扩充性 增加代码的重用性

提高了软件的可维护性

1.类继承的应用

声明一个子类 使用`extends`关键字去继承（扩展）一个父类

2.访问类型的控制

虽然子类可以从父类中继承所有方法，但`private`的成员只能在本类中使用，子类也不行

如何封装既可以让自己的内部访问，也让子类访问，但类外部不能访问 `private`改为`protected`

3.子类中重载父类的方法

子类可以声明和父类相同的方法名，即子类覆盖了父类中同名的方法，子类还可以扩展。

在子类中调用父类中被覆盖的方法

`parent::方法名()`

在子类中编写构造方法，如果父类中也有构造方法一定要去调用一次覆盖那个构造方法

`parent::__construct()`

注意：子类重载的方法不能低于父类中的访问权限

1.如果两个或以上的类中有公共部分就可以共享，共享就可以作为父类

2.如果多各类都需要添加同一个功能，只要在父类中添加，所有子类都有这个功能

PHP java属于单继承，同一个类只能有一个父类，可以有多个子类

子类重载父类的方法

parent::方法名

在子类中重写父类的方法时，一定要高于或等于父类被覆盖的方法的访问权限

8.6常见的关键字和魔术方法

final关键字作用：

- 1.使用final标识的类不能被继承
 - 2.使用final标识的成员方法，在子类中不能被覆盖
- 用来限制类不被继承，方法不能被覆盖
final不能修饰成员属性只能修饰类和方法

static关键字

static可以修饰成员属性和方法，不能修饰类

static可以将类中成员属性标识为静态,这个static成员属于类的，总是唯一的，可以被同一个类的所有对象共享，与对象实例和其他的类无关

静态的数据是在类第一次加载时分配到内存中的，以后用时直接获取

注意：静态成员都要用类名去访问，不用创建对象，不用对象去访问

类名::静态成员

不能用\$this(代表本对象)来引用，而用self(代表本类) 关键字self::\$count++;

静态方法(static修饰的方法)不能访问非静态的成员

const关键字

- 1.只能修饰成员属性
- 2.类中声明常量属性使用const
- 3.命名方式和define一样
- 4.访问方式和static静态成员属性一样
- 5.常量一定要在声明时给初值。

第12章MySQL数据库设计

12.1数据库的连接与关闭

cmd mysql -h 服务器主机地址 -u 用户名 -p

Enter password:

输入好后即可连接

创建、选择、查看数据库

```
CREATE DATABASE [IF NOT EXISTS] bookstore;
```

//创建bookstore数据库

```
DROP DATABASE [IF EXISTS] bookstore;
```

//删除bookstore数据库

```
SHOW DATABASES;
```

```
USE bookstore;
```

数据字段属性

UNSIGNED 只能用于设置数值类型，不允许出现负数

ZEROFILL 只能用于设置数值类型，在数值之前自动用0补齐不足的位数

AUTO_INCREMENT 设置字段的自动增量属性，每增加一条记录，该字段的值就加1

NULL NOT NULL

DEFAULT 指定默认值

12.2.4创建、修改及删除表

创建数据表：

```
CREATE TABLE [IF NOT EXISTS] 表名称(
```

```
    字段名1 列类型 [属性] [索引],
```

```
    字段名2 列类型 [属性] [索引],
```

```
    字段名3 列类型 [属性] [索引],
```

```
    字段名n 列类型 [属性] [索引]
```

```
)[表类型] [表字符集];
```

```
CREATE TABLE IF NOT EXISTS users(
```

```
    userId INT(10) UNSIGNED NOT NULL AUTO_INCREMENT,
```

```
    userName VARCHAR(50) NOT NULL,
```

```
    userPass VARCHAR(50) NOT NULL,
```

```
    telNo VARCHAR(20) NOT NULL UNIQUE,
```

```
    sex ENUM('男','女') NOT NULL DEFAULT '男',
```

```
    brithday date not null default '0000-00-00',
```

```
    PRIMARY KEY(userId),
```

```
    INDEX username(userName,userPass)
```

```
)type=MyISAM DEFAULT CHARACTER SET utf8 COLLATE utf8_general_ci;
```

修改表

```
alter table 表名 add 字段名 &lt;建表语句&gt;[first|after 列名]
```

如果没有指定可选的first或after,则在列尾添加一列，否则在指定列添加新列

删除表

```
drop table 表名
```

12.2.7创建索引

主键索引|primary key

create table categories(
categoryId INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
categoryName VARCHAR(15) NOT NULL
);

唯一索引|unique

防止创建重复的值

常规索引|INDEX

全文索引|fulltext

第16章 php的模板技术smarty

16.4.1安装smarty

在自己的项目中包含smarty类库，这样在移动项目时会带着smarty，不需要改变web服务器配置

直接将libs文件夹复制到项目的主文件夹下

在执行的PHP脚本中通过require()将libs目录中的Smarty.class.php类文件加载进来就可以了

16.5 Smarty在应用程序逻辑层的使用步骤

1、加载Smarty模板引擎 require("Smarty.class.php");

2、建立Smarty对象 \$smarty=new Smarty();

3、修改Smarty默认行为 开启缓存机制，修改模板默认存放目录；

- 4、将程序中动态获取的变量用assign()方法置入模板
- 5、用display()方法将模板输出

前三步是Smarty的核心，可放置在main.inc.php中

assign()方法

在PHP脚本中调用该方法可以为Smarty模板文件中的变量赋值

- 1、指定“名称/数值”的赋值方式

```
$smarty->assign("name","jesse");
```

- 2、指定包含“名称/数组”的联合数组的使用方式

```
$smarty->assign(array("city"=>"lincoln","state"=>"nabraska"));
```

display()方法

smarty脚本中必须用到的方法，而且一个脚本中只能使用一次，他负责获取和显示smarty引擎引用的模板

```
Void display(string template[,string cache_id[,string compile_id]])
```

template 必选 需要指定一个合法的模板资源类型和路径

cache_id 可选 指定一个缓存标识符的名称

compile_id 可选 在维护一个页面的多个缓存时使用

//获取和显示\$template_dir属性所指定目录下的模板文件

```
$smarty->display("index.tpl");
```

//在win平台下绝对路径必须用file:前缀，用来使用不在\$template_dir模板目录下的文件

```
$smarty->display("C:/www/pub/templates/header/tpl");
```

16.6 Smarty模板中的程序逻辑

smarty引擎不仅在php逻辑层使用，在表现层也会偶尔使用（很简单）

模板中的注释

包围在定界符“{*”和“*}”之间的内容都是注释内容，是模板的内在注释

模板中变量的声明

smarty有几种不同类型的变量，变量的类型取决于他的前缀是什么符号。

以下几个变量可以再smarty模板中直接输出

```
{$Name}           {*常规类型变量*}
```

```
{$Contacts[row].Phone}   {*数组类型变量*}
```

```
&lt;body bgcolor="{#bgcolor#}"&gt;{*从配置文件中读取并输出的变量*}
```

在模板中输出从PHP中分配的变量

模板中只能输出从PHP中分配的变量，不能在模板中为这些变量重新赋值

在Smarty模板中变量的预设是全域的。只要分配一次就可以，分配两次以上以最后以此为主

通常在模板中通过遍历输出数组中的每个元素，通过foreach或section语句

在模板中单独输出数组中的某个元素

索引数组 和在PHP脚本的引用方式一样

```
{$contact2[1][2]}
```

关联数组 使用"."访问

```
{$contact.email}
```

索引关联混合数组 {\$contact2[1].name}

在PHP脚本中创建的对象类型变量也可以分配给模板，并访问对象中的每个成员，都是通过“->”运算符完成

```
{$person-&gt;name}
{$person-&gt;work()}
```

16.6.4模板中变量的数学运算

在模板中的变量不能重新赋值，但是可以参与数学运算

```
{$foo+1}
{$foo*$bar}
{$foo-&gt;bar-$bar[1]*$baz-&gt;foo-&gt;bar()-3*7}
{*复合类型变量也可参与运算*}
{if($foo+$bar.test%$baz*134232+10+$b+10)}
{*可以将数学运算在程序逻辑中运用*}
```

在Smarty模板中可以识别嵌入在""中的变量，只要此变量名称只包含字母数字下划线或[]。若有其他符号，此变量必须用两个反引号`包住

16.6.5在模板中使用{\$smarty}保留变量

{\$smarty}保留变量是可以在模板中直接访问的数组类型变量，通常用于访问一些特殊模板变量

1.在模板中访问页面请求变量

```
* 类似在PHP脚本中访问$_GET["page"] *}
{$smarty.get.page}
* 类似在PHP脚本中访问 $_POST['page'] *}
{$smarty.post.page}
* 类似在PHP脚本中访问$_COOKIE['username']*}
{$smarty.cookies.username}
* 类似在PHP脚本中访问$_SERVER['SERVER_NAME']*}
{$smarty.server.SERVER_NAME}
* 类似在PHP脚本中访问$_ENV["PATH"] *}
{$smarty.env.PATH}
* 类似在PHP脚本中访问$_REQUEST["username"] *}
{$smarty.request.username}
```

2.在模板中访问PHP中的变量

```
*在模板中输出php脚本中用户自定义的常量*}
{$smarty.const._MY_CONST_VAL}
*在模板中通过保留变量数组直接输出系统常量*
```

16.6.6变量调节器

变量在模板中输出以前如果需要调解，可以在该变量后面跟一个“|”，在后面使用调解命令。可以使用多个修改器按设定好的顺序依次组合使用

```
{$articleTitle|lower|spacify|truncate}
```

16.6.7模板的控制结构

1.条件选择结构 if-elseif-else

{if}/{/if}必须成对出现，中间的修饰词eq等等必须和变量或常量用空格隔开

2.重复的区域 foreach

{foreach}/{/foreach}必须成对出现

属性 类型 是否必须 缺省值 描述

from array Yes n/a 待循环数组的名称

item string Yes n/a 当前处理元素的变量名称

key string No n/a 当前处理元素的键名
name string No n/a 该循环的名称，用于访问该循环
foreachelse标记一定要与foreach一起使用，与ifelse相似
关联数组和索引数组都可以用foreach遍历

3.重复的区块 section 改进版的foreach

属性	类型	是否必须	缺省值	描述
name	string	Yes	n/a	该循环的名称
loop	[\$variable_name]	Yes	n/a	决定循环次数的数组变量名称
start	integer	No	0	循环执行的初始位置. 如果该值为负数，开始位置从数组的尾部算起. 例如：如果数组中有7个元素，指定start为-2，那么指向当前数组的索引为5. 非法值(超过了循环数组的下限)将被自动调整为最接近的合法值.
step	integer	No	1	该值决定循环的步长. 例如指定step=2将只遍历下标为0、2、4等的元素. 如果step为负值，那么遍历数组的时候从后向前遍历.
max	integer	No	1	设定循环最大执行次数.
show	boolean	No	true	决定是否显示该循环.

section遍历的数组必须是下表从0开始的顺序索引数组
section标记也可以使用可选的sectionelse子标记，当loop属性指定的数组为空时，输出sectionelse区域中的内容，必须与section一起使用

4.重复区块中可供调用的变量

在section和foreach循环中都有可供调用的变量，都必须通过\$smartyy进行访问
{\$smarty.foreach.foreachname.varname}
\$smartyy.foreach是Smarty保留变量
foreachname在foreach标记指定的name属性值
varname在循环中被调用的特定变量名称
foreach循环区域中可调用的变量

变量名	描述
iteration	用于显示当前循环的执行次数，总是从1开始，每执行一次增加1
first	当前 foreach 循环第一次执行时 first 被设置成 true.
last	当前 foreach 循环执行到最后一遍时 last 被设置成 true.
show	是 foreach 的一个参数. 取值为布尔值 true 或 false. 如果指定为 false 该循环不显示，如果循环指定了 foreachelse 子句，该子句显示与否也取决于 show 的取值.
total	用于显示循环执行的次数，可以在循环中或循环执行后调用.

section循环中调用可供变量的方式与foreach相似
section循环区域中可调用的变量

index
index_prev
index_next
iteration
first
last
rownum
loop
show
total

16.6.8在模板中包含子模板

如果在多个模板中有相同的输出内容，这些相同的部分可在独立的模板中定义吗，在需要的模板中将其用include标记导入，但必须指定模板资源的位置。

一般头部文件和尾部文件都可作为独立的模板导入

```
{include file="header.tpl"}
```

```
{include file="footer.tpl"}
```

在include标记中还有两个比较实用的特性

1、 assign属性

将导入的子模板内容不再当前模板中输出，而是赋给由assign指定的变量

```
{include file="header.tpl" assign="header"}
```

{*模板header.tpl中的全部内容以字符串的形式赋值给变量header，再通过{ \$header}

2、 可以在导入子模板的同时向其传递各种属性

```
{ include file="header.tpl" title="menu" table_bgcolor="#ccc"}
```

以此方式传递的各种属性只能在被导入的文件中使用，不能用于模板的其他位置。如果模板中有同名变量，则会被传递的属性替代

16.7.2加载配置文件

```
{ config_load file="foo.conf" section="customer"}
```

```
{ *加载foo.conf中第一节customer中的变量*}
```

16.7.3引用配置文件中的变量

配置文件中的变量需要通过两个##或者是\$smarty.config来调用

16.8.3缓存

is_cached() 判断指定模板的缓存是否存在

clear_all_cache()清除所有缓存

clear_cache()清除单个缓存

16.8.5关闭局部缓存

1、 使用{ insert}使模板的一部分不被缓存

2、 使用\$smarty->register_function(\$params.&,\$smarty)阻止插件从缓存中输出

3、 使用\$smarty->register_block(\$params,&,\$samrty)是整篇页面的某一块不被缓存

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:www.tushu000.com