



PHP基础教程

极客学院出版

前言

PHP 最为目前流行的服务器端开发语言，深受各大互联网工程师的喜爱。本教程是 PHP 语言的基础教程，根据作者学习 PHP 语言，遇到的知识点总结而成。是一本不错的具有手册性质的教程。能够让读者快速掌握 PHP 的要点。

适用人群

学习本教程前，你需要对 PHP 的背景和开发工具有一定的了解，如果你有其他面向对象语言基础的读者，那么本教程不失为一本不错的参考手册。

学习前提

准备基础开发环境，对电脑的基础常识要具备，如果有一定的开发能力，那么上手会很快。

鸣谢：<http://daige.me/tag/php/>

目录

前言	1
第 1 章 PHP 概述	3
第 2 章 变量	7
第 3 章 HTML 表单和 PHP	10
第 4 章 使用数值	13
第 5 章 使用字符串	15
第 6 章 控制结构	19
第 7 章 使用数组	21
第 8 章 创建 Web 应用程序	25
第 9 章 cookie 和 session	28
第 10 章 创建函数	32
第 11 章 文件和目录	34
第 12 章 数据库介绍	38
第 13 章 总结	40



PHP 概述



HTML 语法基础

本书使用 XHTML，XHTML 和 HTML 的区别有：

- 所有的标签都使用小写字母
- 嵌套标签必须有恰当的格式(必须就近匹配)
- 所有的标签属性值必须使用引号引起来
- 所有的标签必须关闭 有些HTML标签并不强制使用关闭标签，但是有效的 XHTML 标签，必须在末尾处添加一个空格和斜线以关闭它们 比如: ``

CSS 基础

CSS 是格式化 HTML 和 XHTML 内容的外观和行为

在 web 页面加入 CSS，有两种方法：

- style 标签

```
<style type="text/css">
```

CSS 规则

```
</style>
```

- 引用外部文件

```
<link href="style.css" rel="stylesheet" type="text/css" />
```

CSS 的规则可以应用于一般的页面元素，CSS 类和特定元素

- `img { border: 0px;}` 应用所有图片标签
- `.error { color: red;}` class 为 error 的元素
- `#about {background-color: #ccc;}` id 为 about 的特定元素

PHP 语法基础

PHP 文档和 HTML 文档的区别：

- 后缀不一样 PHP 文档为 .php 后缀
- 在 PHP 文档中，必须使用 PHP，而且只有在 `<?php ?>` 中才能使用 PHP

- PHP 脚本必须运行在启动了 PHP 的 web 服务器上 所以 PHP 必须通过 URL 运行

向浏览器发送文本

PHP 最频繁的操作就是以文本和 HTML 标签的方式向浏览器发送信息

```
print "something";
```

print 是一个语言结构，并不是函数，作用是输出数据

```
<html>
<body>
<?php
    echo "hello world";
?>
</body>
</html>
```

类似 print 功能的有 echo 结构和 printf() 函数

向浏览器发送 HTML

使用 PHP 向浏览器发送 HTML 标签和其他数据：

- `print "hello\nworld!";` \n换行符 注意：只影响HTML源代码
- `print "link";` 对引号进行转义
- `print "hello";` 输出CSS代码

注意，任何在 PHP 标签外的 HTML 都将自动发送到浏览器

调试 PHP 脚本

- 确认总是通过 URL 来运行 PHP 脚本
- 熟知运行的 PHP 版本
- 确保 display_errors 是开启状态
- 检查 HTML 源代码
- 相信错误消息

- 休息一下



T



变量



预定义的变量

PHP 中的变量是临时的，只存在于脚本执行期间

预定义的变量是不需创建就可以使用，因为 PHP 已经完成创建工作

`print_r()`函数可以显示变量值

打印 PHP 预定义的变量：

```
<?php
    print_r($_SERVER);
?>
```

`$_SERVER` 这个预定义的变量包含了服务器的所有数据

例如：

```
[HTTP_ACCEPT] => text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
[HTTP_USER_AGENT] => Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/38.0.2125.
[SERVER_SOFTWARE] => Apache/2.4.9 (Win32) OpenSSL/1.0.1g PHP/5.5.11
```

变量命名

注意：PHP 的变量区分大小写

变量命名建议：

- 变量名全部用小写字母
- 让变量名具有意义
- 使用注释说明变量用途
- 保持一致的命名规则

变量类型

- 字符串
"1990" 字符串型的整型在进行数学运算时会被当成数值型

- 数组

PHP 的数组有两种类型：索引数组和关联数组

索引数组：使用数组作为键

关联数组：使用字符串作为键

print 不能打印非单值的类型，数组和对象需要使用print_r打印值

理解引号

- 单引号 单引号的内容将照字面意思进行处理
- 双引号 双引号引用的内容需要进行推断 双引号里面的变量将被值代替

```
$str = '1990'; //这里可以使用双引号
print '$str'; //$str 单引号 字面值
print "\n"; //转义字符 必须双引号
print "$str+10"; //1990+10 双引号 变量被值替换
```



3

HTML 表单和 PHP



简单的表单

HTML 中的创建表单使用 form 标签

form 标签有一个 action 属性，用来指明将表单数据提交到哪个页面

```
<form action="handle_form.php"> 提交的页面文件需要指定路径
```

form 中还需要执行 method 属性，它是告诉服务器如何将数据传送到处理的脚本

```
<form action="handle_form.php" method="get">
```

GET 和 POST 的区别

- GET 方式，传送的信息量有限，而且是以公开的方式在 URL 中传送，GET 方式的可以添加书签，因为 URL 中带有信息，一般从服务器请求信息用 GET，
- POST 方式，传送的信息可以很多，不公开，但是不能设书签一般用在触发基于服务器的行为，如提交数据等场景 在表单中，一般使用 POST

使用 PHP 接收表单数据

PHP 处理表单数据引用的特定变量可能是 `$_GET` 或者 `$_POST`，如果你使用 GET 方式传送数据那么数据就存储在 `$_GET` 中，POST 方式就存储在 `$_POST` 中

注意：`$_GET` 和 `$_POST` 是预定义变量，也是数组，也是一种特殊的变量类型

访问数组需要使用索引或者键

比如：

- `print $_POST['hello']` 但是需要注意的是，在下面情况中会出现解析错误：
- `print "$_POST['hello'],world"` 双引号中，无法解析 `$_POST` 数据,需要使用变量

如果想向 PHP 脚本传递预设值，可以在 HTML 表单中使用文本输入框的隐藏类型，例如：

```
<input type="hidden" name="page" value="feedback.html" />
```

关于 Magic Quote：Magic Quote 是一个过时的特性，如果你提交的表单数据对单引号双引号会被自动的转义处理，那么 Magic Quote 就是开启状态

显示错误

开启错误显示：

- 开启 `display_errors` 开发网站时，首选
- 为某个单独的脚本开始 `display_errors` 设置

在脚本中包含代码：`ini_set('display_errors',1);` 使用`ini_set`可以覆盖PHP配置 文件中的设置，1表示开启。注意：这种方法的不足之处是如果脚本发生特定类型错误，这行代码不执行，会显示空白页

注意：运行脚本时如果看到空白页，也需要检查HTML代码

错误报告

PHP 有 11 种不同的错误，常见的 4 种错误类型有：

- 通知 非致命性的错误
- 警告 非致命性错误 通常表明有问题存在
- 解析错误 由语法错误导致的致命性错误
- 错误 一般性的致命错误

两种设置 PHP 报告错误的方法：

- 在 PHP 配置文件中调整 `error_reporting()` 级别
 - 在脚本中使用 `error_reporting()` 函数来调整级别
- 通常设置 `error_reporting(E_ALL | E_STRICT);` 来开发和测试PHP脚本

向页面手动发送数据

手动发送数据，可以通过生成 URL 链接，靠点击来使用 `$_GET` 传递数据

比如：

```
<a href="handle_form.php?test=daige">daige</a>
```

这种方式不推荐使用



使用数值



格式化数值

- round 对数值截取特定位数的小数 `round(4.289,2);` 输出 4.29
- number_format 通过千位分组来格式化数字 `number_format(1234.118,2)` 输出 123,4.12
- abs 返回数值的绝对值

在 PHP 中，round()分为两种函数：

- ceil 最近的整数入运算 `ceil(11.2)` 输出12
- floor 最近的整数舍运算 `floor(11.2)` 输出11

创建随机数

rand()是一个创建随机数的生成工具

rand()可以接受最小值和最大值参数来产生特定范围的随机数

`rand(0,10)` 产生 0 到 10 之内的数

getrandmax()返回使用 rand()可能产生的随机数的最大数值

mt_rand 是一个更好的生成随机数的函数



使用字符串



连接字符串

在编程的时候，特定的字符串连接，使用运算符句点(.)

```
$greet="hello, " . "world!";
```

也可使用：

```
$greeting="hello, "; $greeting .= "world!"
```

PHP 在处理字符串变量的时候，使用变量替换他们的值：

```
$greeting="$s1$s2"
```

处理换行符

PHP 中的 nl2br() 函数能够将换行符自动换为 break 标签

比如：

```
<p>备注：
    <textarea name="commit" rows="3" cols="30"></textarea>
</p>
//把textare中输入的换行符转为break
print nl2br($_POST["commit"]);
```

HTML 和 PHP

用户在表单中可以输入 HTML 字符，这样将对页面的格式产生影响，更糟糕的情况是可能引发安全方面的问题，所以有必要处理字符串变量中的 HTML 标签

处理字符串中的HTML标签的函数有：

- htmlspecialchars() 将特定的 HTML 标签转换为实体版本
- htmlentities() 将所有的 HTML 标签转换为实体版本 相反功能的函数为 html_entity_decode
- strip_tags() 移除所有的 HTML 和 PHP 标签，如果还需要将换行符转为 break 标签，可以先使用 strip_tags 后再使用 nl2br

htmlspecialchars 和 htmlentities 会把 HTML 标签转义，比如：

输入 `<script>alert(1)</script>` 会得到 `<script>alert(1)</script>gt;`

字符串的编码和解码

在使用 GET 方法的时候，是把数据通过 URL 传递的，如果想在传递的数据中有多个词以及特殊的符号那就需要对数据进行编码

urlencode 函数可以对一个字符串进行编码，使它能够完全适合 URL 的传输，它会用+号替换空格，并把特殊字符替换

```
<?php
    $name = "hello,world!\ndaige ";
    $name = urlencode($name);
    $uid="123456";
    $uid=urlencode($uid);
    print "Click <a href=\"name.php?name=$name&uid=$uid \">here</a> to continue";
?>
```

编码之后的 URL 为：

```
name.php?name=hello%2Cworld%21%0Adaige+&uid=123456
```

当我们点击此链接的后，我们并不需要使用 urldecode 函数来解码才能获取数据，因为 PHP 会自动解码

字符串的加密和解密：

PHP 可以使用 crypt 函数来对数据进行加密，注意它是单向加密

另外 Mcrypt 扩展中提供了一对加密解密函数：mcrypt_encrypt()和 mcrypt_decrypt()

查找子字符串

PHP 提供一些函数用来拆解，搜索和比较字符串

- strtok
- substr
- strlen
- str_word_count 字符串的单词数量
- strcmp 不区分大小写 strcmp

- strnatcmp 不区分大小写 strnatcasecmp
- strstr 不区分大小写 stristr
- strpos 不区分大小写 stripos

替换局部字符串

- str_ireplace 不区分大小写
- str_replace 区分大小写
- trim 移除字符串首尾的所有空白 rtrim 移除右边 ltrim 移除左边
- ucfirst 第一个字母大写
- ucwords 每个单词第一个字母大写
- strtoupper 整个字符串大写
- strtolower 整个字符串小写



控制结构



验证函数

PHP 验证表单的常用函数有：

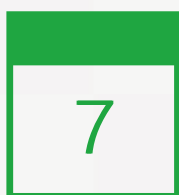
- `empty` 检验一个变量是否拥有 0 和空字符之外的其他值，没有返回 TRUE
- `isset` 当变量拥有值(包括 0, FALSE, 空字符串)返回 TRUE 避免引用不存在的变量
- `is_numeric` 检验变量是不是一个有效的数字类型 是返回 TRUE 数字字符串有效

switch 语句

switch 语句是替换很多的 if-elseif-else 的好办法

注意 switch 语句中的 case 可以为字符串和数值，而且字符串是区分大小写的

- `break` 退出当前结构 可以是 switch if-else 或者循环
- `continue` 终止当前的迭代 剩下的语句不执行 再次检查循环条件
- `exit` 终止 PHP 脚本 `exit` 后面的 php 脚本将不会被执行
- `die` `die` 与 `exit` 相同



使用数组



超全局变量

PHP 中有一些数组是超级全局变量

- `$_GET`
- `$_POST`
- `$_SERVER`
- `$_COOKIE` 存储在 cookie 中保存的数据
- `$_SESSION` 保存在 session 中的数据
- `$_ENV` 保存环境变量

创建数组

使用 `array()` 函数来创建数组

- `$list=array("1","2","3");` 索引从 0 开始
- `$list=array(1 => "1",2 => "2",3 =>"3");` 指定特定的索引，指定数值或者字符串

还可以使用 `range()` 函数创建：

- `$ten = range(1,10);`
- `$alphabet = range('a','z',1);`

显示数组可以使用 `var_dump` 代替 `print_r`，它更详细呈现数组的结构

向数组添加项

- 向数组添加项使用赋值操作符 `$list[3]='apple';` 注意要使用方括号
- `count` 获取数组中元素的数量 `sizeof` 函数具有相同的效果
- `unset` 删除变量并释放内存
- `$list=array()` 重置数组(清空数组的值)
- `$list[] = 'apple';` 如果 `$list` 不存在，将创建数组

数组合并：

`array_merge()` 函数可以合并两个数组，也可以使用 `+` 号

访问数组元素

如果数组被设定为以字符串作为键，那么用数值为键不指向任何值

遍历整个数组最快的方式为使用 `foreach` 循环：

```
$alphabet=range('a','z',1);
foreach ($alphabet as $key => $value)
{
    print $key . $value . "<br/>";
}
```

创建多维数组

多维数组是可以使用其他数组作为它的值的数组，访问的时候，需要多层键值索引，遍历时需要多层 `foreach`

使用 `print_r` 或者 `var_dump` 函数可以对多维数组查看

数组排序

- `sort` 排序值
- `rsort` 反向排序值
- `asort` 排序值
- `arsort` 反向排序值
- `ksort` 排序键
- `krsort` 反向排序键
- `shuffle` 随机重组值
- `natsort` 自然排序字符串
- `natcasesort` 自然排序字符串不区分大小写
- `usort` 对数组使用用户自定义比较函数方向进行排序
- `uasort` 这三个函数通常应用在多维数组中
- `ursort`

字符串和数组之间的转换

implode 将数组转换为字符串

```
$daige= array('1' => "hello" , '2' => "world!");  
hello=implode(", ", $daige); //连接符连接数组的值  
print $hello;
```

explode 将字符串转换为数组

```
$daige= "hello,world!";  
$hello=explode(", ", $daige); //分隔符分割字符串  
print_r ($hello);
```

list 函数 list 用来讲数组元素的值赋给单独的变量

```
$date = array('Thursday',23,'October');  
list($weekday,$day,$month) = $date;
```

这样，三个变量分别得到数组的值，但是这种赋值有限制：

- list 函数只能对数值索引并从 0 开始的索引有效
- list 必须确认接受每一个数组元素，可以通过空值来忽略元素



T



8

创建 Web 应用程序



创建模板

在复杂的 Web 站点中，有一些特性会在站点的每个页面都用到，我们就可以将这些元素放在一个独立的页面。

创建模板可以将重复的内容和特定的页面的素材分开

比如：WP 中分为 header, sidebar footer 等模块

使用外部文件

- include 包含文件 如果 include 失败，PHP 会继续运行
- require 包含文件 如果 require 失败，PHP 会终止脚本执行

```
include ("test.html"); 或者 include "test.html";
```

使用常量

- define() 函数定义并赋值常量

```
define ('CONSTANT_NAME',value) 常量名需全部大写
```

常量的引用很简单，直接 `print CONSTANT_NAME` 但是在单引号和双引号里面的常量就无法打印

- defined() 函数 如果提交的常量已经定义，返回 TRUE

使用日期和时间

- date 返回格式化的日期和时间
- time 返回当前时刻的时间戳
- mktime 返回给定时间和日期的时间戳
- date_default_timezone_set 设置时区

输出缓冲

某些函数，只能在没有任何东西被发送到浏览器之前调用，比如 `header()`、`setcookie()` 和 `session_start()`，如果在 Web 浏览器已经收到了一些文本、HTML 或哪怕是一个空格之后，调用这些函数，就会得到一个恼人的 HTTP 头已发送错误消息，这个时候采用输出缓冲可以解决这个问题

利用输出缓冲，HTML 和打印的数据将被放到缓冲中。当脚本执行结束后，缓冲将被发送到 Web 浏览器，或者如果需要的话，缓冲可以清空而不发送到 Web 浏览器

启用输出缓冲：在页面顶端使用 `ob_start` 函数，在结尾调用 `ob_end_flush` 函数会将缓冲发送到 Web 服务器，调用 `ob_end_clean` 函数会删除缓冲的数据而不进行传输

缓冲区的大小可以在 `php.ini` 文件中设置，默认值为 4096 字节

- `ob_get_length` 返回缓冲内容长度
- `ob_get_contents` 返回缓冲区内容
- `ob_flush` 将当前缓冲区内容发送到 Web 浏览器并刷新缓冲区
- `ob_clean` 删除缓冲区的当前内容

处理 HTTP 头

Web 服务器通常需要用其他方式与客户端通信，这些通信需要用 HTTP 头实现

下面介绍用 `header` 函数重定向页面：

使用 PHP 重定向浏览器，需要发送一个 location 头：

```
header("Location: page.php") 通常后面会跟 exit(); 取消当前脚本执行
```



T



9

cookie 和 session



创建 cookie

cookie 必须在发送其他任何信息之前从服务器发送到客户端,如果使用缓冲可在任何位置

setcookie 函数发送 cookie 用法:

```
setcookie(name,value);
```

读取 cookie

函数 setcookie 将 cookie 数据存放在数组 \$_COOKIE 中

```
$_COOKIE['name']
```

添加 cookie 参数

setcookie 函数可以接受 5 个参数:

```
setcookie(name,value,expiration,path,domain,secure,httponly);
```

- name
- value
- expiration 指定 cookie 过期时间 没有设置则为 session(浏览器)关闭为止
- path 指定 web 站点特定文件夹特定域的 cookie
- domain 指定子域 cookie
- secure 指定 cookie 应当通过 HTTPS 连接传送 1 为必须使用安全连接
- httponly 限制对 cookie 的访问防止 xss 的利器

删除 cookie

通常 cookie 会在用户浏览关闭或者过期时间到了之后自动过期,但是我们还是需要手动删除它

删除 cookie 有两种方法:

- `setcookie(name,"");` or `setcookie(name,FALSE);` 把值设为空

- `setcookie(name,FALSE,time() - 600);` 把过期时间设为过去

什么是 session

session 是一种解决方案，解决的问题是它可以跟踪用户在一系列页面访问的数据

session 和 cookie 的区别：

- cookie 存储在客户端，session 存储在服务器端
- session 通常更安全，数据不会在客户端和服务器端来回传输
- session 能够存储比 cookie 更多的东西
- session 可以在用户不接受他们浏览器中的 cookie 时工作
- cookie 更加容易创建和检索
- cookie 对服务器造成的压力更少
- cookie 能够持续更长的时间

创建 session

使用 `session_start` 函数创建，访问或者删除 session

这个函数在首次启动时，会发送一个 cookie，所以必须在任何 HTML 发送给浏览器之前调用，在使用 session 的页面中，脚本必须起始行调用 `session_start` 函数

在第一次开启 session 时，会产生一个随机的 session ID，并向浏览器发送一个名为 PHPSESSID 的 cookie

启用 session，向数组 `$_SESSION` 赋值的形式为：

```
$_SESSION['username']= 'daige';
```

函数 `session_name` 函数可以修改 session 的名称来替代 PHPSESSI，而且应该在调用 `session_start` 函数之前使用

函数 `session_set_cookie_params` 函数用来修改 session cookie 的设置

访问 session 变量

创建或者访问一个存在的 session，都必须从 `session_start` 函数开始

访问变量和普通数组一样，如 `$_SESSION['mail'];`

- `isset($_SESSION['var'])` 可以查看特定的 session 值是否存在
- session 中的数据都是以纯文本的方式保存在一个可读可写的文本文件中，不要在 session 中存放敏感的数据
- 为了提高安全性，session 可以加密存储，读取 session 时解密

删除 session

session 的数据在两个地方存在，所以需要在两个地方进行删除操作

必须从 `session_start` 函数开始，然后设置 `$_SESSION` 数组来删除 session 值

```
$_SESSION = array(); 最后在服务器上删除 session 数据 session_destroy();
```




创建函数



global

global 声明就是：希望函数内的变量能够指向函数外具有相同名称的变量

global 就是把局部变量变成全局变量

```
function name($args)
{
    global $var; //转变为全局变量
    statement(s);
}
```

函数返回多个值

PHP 通过数组返回对个值

```
function some_function($a1,$a2)
{
    //something
    return array($v1,$v2);
}
```

调用的时候，使用 list 函数来接受变量：

```
list($var1,$var2) = some_function($p1,$p2)
```



11

文件和目录



文件权限

在 PHP 脚本得到通过 Web 服务器运行时，会视为一个不同的服务器用户，然后 PHP 和 web 服务器都能够读取上传的文件，但是 PHP 无法修改这些文件

Web 浏览器可以访问 Web 根目录及其子目录的所有文件，但是不能访问 Web 根目录外面的目录，所以，在创建可写和目录的时候，放在 Web 根目录之外更安全

写入文件

PHP5 中的函数：

```
file_put_contents($file, $data, FILE_APPEND)
//打开文件并写入数据，向文件追加数据

file_put_contents($file, $_POST['quote'] . PHP_EOL, FILE_APPEND);
//PHP_EOL 是表示当前操作系统的换行符
```

如果不是 PHP5，则需要使用旧的方法：

```
$fp = fopen($file, mode);
fwrite($fp, $data . PHP_EOL);
fclose($fp);
```

在向文件或者目录写入数据的时候，需要调用 `is_writable` 来避免权限错误

```
if( is_writable($file))
//返回布尔值 表示指定文件是否可以被写入
```

锁定文件

当多个 PHP 脚本试图同一时刻写入一个文件时，就会出现这个问题

在 PHP5.1 以上版本可以这样：

```
file_put_contents($file, $data, FILE_APPEND | LOCK_EX)
//LOCK_EX 是表示暂时锁定该文件
```

在早期的 PHP 版本中，需要使用：

```
flock($fp,locktype)
//LOCK_SH 读取共享锁
LOCK_EX 写入独享锁
LOCK_UN 释放一个锁
LOCK_NB 非阻塞锁
```

在 fclose 的时候，文件会自动解锁，最好在写入完成的时候加上解锁的语句

读取文件

- file_get_contents 所有内容读到一个字符串中
- file 每行一些数据 读取之后放在一个数组中
- fgets 返回具有指定长度的字符串

fgets 和 C 语法一样：

```
$fp = fopen($file, 'rb');
while (!feof($fp))
{
    $string = fgets($fp, 1024); //返回1023字节数据
}
fclose ($fp);
```

处理文件上传

处理文件上传必须：

- 初始 form 标签时必须包含代码 `enctype="multipart/form-data"`，表单必须使用 POST
- 必须添加一个特殊的隐藏输入框 用来建议浏览器最大能够上传的文件大小
- 使用 file 元素创建所需要的表单字段

在 PHP 中，\$_FILES 变量包含 5 个元素：

- name 文件名
- type MIME类型
- size 文件大小（字节）
- tmp_name 文件存放在服务器上的临时文件名
- error 如果发生错误，保存的错误代码

从安全角度，最好重命名上传的文件，要完成这一任务，需要设计一个系统，能够生成一个新的唯一的文件名，并在一个文本文件或数据库中同时保存原始文件名和新文件名。



12

数据库介绍



安全查询数据

PHP 安全涉及很广，而其中的 sql 注入也需要很深入了解。具体的知识点，另外一本书《SQL注入攻击与防御》值得参考，这本书的读书笔记也在进行中，可以关注一下

这里列举几个 PHP 关于安全查询的函数：

- `mysql_real_escape_string` 对任何可能危险的字符进行转义（在前面加一个反斜线）
- `addslashes`
- `htmlentities`
- `htmlspecialchars`
- `strip_tags` 除去字符串中所有的HTML，JS和PHP标签
- `md5`
- `sha1`
- `intval`

PHP 操作 mysql 函数

- `mysql_connect`
- `mysql_error`
- `mysql_select_db`
- `mysql_query`
- `mysql_fetch_array`
- `mysql_num_rows`
- `mysql_affected_rows`
- `mysql_close`



13

总结



功能

- 登录
- 登出
- 显示所有名人名言
- 添加名人名言
- 编辑名人名言
- 删除名人名言
- 主页显示名人名言

缺陷

- 数据库
只有一个表，表的设计也非常的简单，功能不足
- 登录
登录采用验证用户名和密码，这种方式非常不好。需要设计用户表，理清权限
验证是不是管理员时才用 COOKIE 验证，不安全，COOKIE 可以伪造，需要采用用户表中的权限验证，用户表中的密码需要加密
登录之后并没有跳转到主页，体验非常不好
- 登出
登出时设置 COOKIE 为过期，但是并没有跳转到主页
- 显示所有名人名言
显示所有的时候，样式不好看，没设计分页
- 添加名人名言
可以对添加的名人名言和数据库中存在的去重，添加时也可以选择类别等
- 编辑名人名言
编辑时，如果并没有任何的修改，也会更新一次数据库
- 删除名人名言
可以提示确定用户是否删除
- 主页显示名人名言
主页需要提供搜索功能，按照日期，来源和关键字等

后来的话

PHP 基础教程，到这里算是看完了，这前后经历了几个月的时间，跨度非常大。我有段时间觉得太简单了，就搁浅了一段时间，最后还是决定把它看完。

这本书的知识非常的基础，对于入门 PHP 也很有作用，我准备在此项目后面更新以后 PHP 的博文链接，希望有兴趣的人可以关注一下

至此，我的 PHP 旅程正式开始，后面将学习更多 PHP 知识，写出自己想要的功能

极客学院

jikexueyuan.com

中国最大的IT职业在线教育平台



更多信息请访问 

<http://wiki.jikexueyuan.com/project/php-basics-notes/>