

PHP基本语法

PHP基本语法

一、PHP简介与环境搭建

什么是PHP

PHP环境安装

代码编辑选择

二、基本语法

PHP基本语法

PHP变量与输出

PHP常量

PHP注释

PHP单引号双引号声明

三、登录页面

PHP表单

四、登录页面构造与mysql交互

PHP构造简单登录页面

一、PHP简介与环境搭建

什么是PHP

(外文名:PHP: Hypertext Preprocessor, 中文名:“超文本预处理器”)是一种通用开源脚本语言。语法吸收了C语言、Java和Perl的特点,利于学习,使用广泛,主要适用于Web开发领域。用PHP做出的动态页面与其他的编程语言相比,PHP是将程序嵌入到HTML(标准通用标记语言下的一个应用)文档中去执行,执行效率比完全生成HTML标记的CGI要高许多;PHP还可以执行编译后代码,编译可以达到加密和优化代码运行,使代码运行更快。

目前PHP在全球网页市场、手机网页市场还有位手机提供API(程序接口)排名第一

PHP入门简单,学习入门易入手

PHP环境安装

集成环境

PHPStudy

小皮面板

XAMPP

同时，WINDOWS/LINUX上同样可直接安装环境，在这里建议选用PHPStudy作为环境安装，官网为(<https://www.xp.cn/>)

代码编辑选择

选择PHP代码的编辑器时，这里给出两点建议

拥有代码高亮功能

不会将进行自动提示（建议新手提示）

推荐：

phpstorm:(<http://phpstorm.p2hp.com/>)

notepad++:(<https://notepad-plus-plus.org/>)

vscode:(<https://code.visualstudio.com/>)

二、基本语法

PHP基本语法

```
<!DOCTYPE html>
<html>    这段代码作为一个PHP文件示例，将实例内容放置服
    <body> 务器端可实现浏览器输出的操作
        <?php echo "hello word!"?>
    </body>
</html>
```

PHP变量与输出

```
<?php
$iphone13 = 5880; //变量首字母不能是数字，同时，变量区分大小写//
$iphone12 = 6088; //把变量命名的有意义一点，否则会给读者挖坑//
$shen = $iphone13 + $iphone12
echo $shen; //echo是在PHP里最常用的一个输出、显示功能的命令//
?>
```

\$叫做美元符，PHP中的变量必须以美元符开始

PHP常量

啥是变量？

顾名思义，就是长久不变的量，在PHP中，用define来定义一个常量，如下：

```
<?php
define('my_name','何源');

echo my_name;
?>
```

系统内置常量

LINE 当前所在的行

FINE 当前文件所在服务器的路径

FUNCTION 当前函数名

CLASS 当前类名

METHOD 当前成员方法

PHP注释

// 表示当行注释

#号也表示单行注释，用的比较少

多行注释(注释多行) /*注释多行/

PHP单引号双引号声明

```
?>
<?php
$byball = '大哥';
$sellball = "你好";
?>
```

在PHP中，单引号双引号都可适用于字符串

更推荐适用单引号，双引号会解析其中的 变量

1.对变量解析的不同

PHP 会解析双引号中的变量，而不会解析单引号中的变量。也就是说，如果使用单引号定义的字符串中出现变量，在输出时变量会被原样输出，不会解析成变量的值。而如果使用双引号定义的字符串中存在变量，在输出时变量会被解析为具体的值。

需要注意的是，虽然双引号定义的字符串能够解析变量，但是如果变量后边还有字符串的话，就需要将变量与后面的字符串使用空格分开，或者使用大括号{ }将变量包裹起来。如果不这么做的话，很可能会造成意想不到的结果。

2.转义字符的不同

单引号和双引号中都可以使用转义字符\，但是，在单引号定义的字符串中只能转义单引号和转义符本身，而在双引号定义的字符串中，PHP 可以转义更多的特殊字符。

转义字符 说明

\n 换行

\r 回车

\t 水平制表符，一个缩进

\v 垂直制表符

\$ 代表\$本身

\ 代表\本身

' 代表单引号本身

\“ 双引号本身

3.解析速度不同

因为单引号不需要考虑变量的解析，所以处理速度比双引号要快，我们在定义字符串时应该尽量遵循能用单引号尽量用单引号的原则。

如果在字符串中不需要加入变量或者单引号'和反斜杠\时，尽量使用单引号来定义字符串。因为省去了双引号检查处理转义字符和解析变量的时间，所以可以提高程序的运行效率。

三、登录页面

该html文档能够建立一个提交页面，我们利用post提交方式向get_name.php提交了名字以及性别资料，点击提交按钮进行提交

```
<html>
  <head>
    <meta charset="utf-8">
    <title>问卷调查</title>
  </head>
  <body>
    <form action="whomain.php" method="post">
      你的名字: <input type="text" name="fname">
      <br>
      多大: <input type="text" name="age">
      <input type="submit" value="提交">
    </form>
  </body>
</html>
```

PHP表单

接下来，我们写一个php文档，放在服务器端

```
<!DOCTYPE html>
<html>
  <body>
    你叫<?php echo $_POST["fname"]; ?>啊 <br>
    <?php echo $_POST["age"]; ?> 那有男朋友吗?
  </body>
</html>
```

四、登录页面构造与mysql交互

PHP构造简单登录页面

利用input表单就可以简单构造登陆页面

标签:框住标签内容

标签: 为fieldset标签添加标题

标签: 点击字体, 会自动聚焦后面控件

- 标签: 列表标签, 上方ul标签代表着下方li标签内容无序显示, 同时, ol标签则代表这有序显示

checkbox: 复选圆框表单元框

网页端代码

```
<!DOCTYPE html>
<html>
  <meta charset="utf-8">
  <body>
    <form action="login.php" method="post">
      <legend>用户登录</legend>
      <ul>
        <li>
          <label>用户名:</label>
          <input type="text" name="username">
        </li>
```

```

<li>
    <label>密码: </label>
    <input type="password" name="password">
</li>
<li>
    <label> </label>
    <input type="checkbox" name="remember" value="yes">7天自动登
录
</li>
<li>
    <label> </label>
    <input type="submit" name="login" value="登录">
</li>
</ul>
</fieldset>
</form>
</body>
</html>

```

服务器端并和**mysql**数据库交互代码**

```

<?php
header('Content-type:text/html; charset=utf-8');
//开启session
session_start();

//处理用户信息
if(isset($_POST['login'])){
    #接受用户的登录信息
    $username = trim($_POST['username']);
    $password = trim($_POST['password']);
    //判断提交的登录信息
    if(($username == '') || ($password == '')){
        //若为空,视为未填写,提示错误,并3秒后返回登录页面
        header('refresh:3; url=login.html');
        echo "用户名或密码不能为空,系统将在3秒后跳转到登陆页面,请重新填写登
录信息";
        exit;
    }
    //连接数据库
    $con = mysqli_connect('localhost','root','123');
    //验证数据库连接状态
    if(mysqli_errno($con)){

```

```

        echo "连接失败,请重试".mysqli_error($con);
        exit;
    }
    //设置解码方式
    mysqli_set_charset($con,'utf-8');
    //设置database
    mysqli_select_db($con,'test');
    //查看传入的用户名和传入的密码与数据库中的值是否相通
    $sql = "select * from login where username = '$username'
and password = '$password'";
    $result = mysqli_query($con,$sql);
    $num = mysqli_num_rows($result);
    if(!$num){
        #用户名或密码错误, 用同处理方式
        header('refresh:3; url=login.html');
        echo "用户名或密码错误,系统将在3秒后跳转到登陆页面,请重新填写登录信
息";
        exit;
    }
    else{
        #用户名和密码都正确, 将用户信息存到session中
        $_SESSION['username'] = $username;
        $_SESSION['islogin'] = 1;
        //若勾选7天内自动登录, 则将其保存到Cookie中并保留七天
        if($_POST['remember'] == "yes"){
            setcookie('username', $username,
time()+7*24*60*60);
            setcookie('code', md5($username.md5($password)),
time()+7*24*60*60);
        }
        else{
            //没有勾选则删除Cookie
            setcookie('username', '', time()-999);
            setcookie('code', '', time()-999);
        }
        echo "登录成功";
        mysqli_close($con);
    }
}
?>


```

接着去创建数据库


```
create databases test;    创建数据库
use tables;    使用数据库
create tables login;    创建login表
insert into login values("username","password");    插入数据
select * from login;    查询是否存在
```

然后去连接就成功了

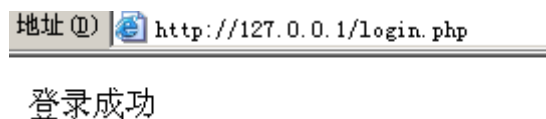



地址 (U)  http://127.0.0.1/login.html

用户登录

- 用户名:
- 密码:
- ☒ 7天自动登录
-

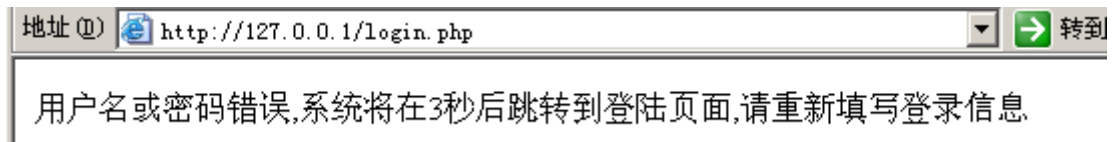
登陆成功






地址 (U)  http://127.0.0.1/login.php

登录成功

登录失败



地址 (U)  http://127.0.0.1/login.php   转到

用户名或密码错误,系统将在3秒后跳转到登陆页面,请重新填写登录信息.

成功了之后就完成了数据库交互