# Web开发

Html：

Form的用法：表单，通常是客户端跟服务器之间进行交互数据的html元素，表单用于搜集不同类型的用户输入

重要属性：action、method

Cation:表示当前表中的内容提交给哪个页面进行处理

Method:表示当前表单提交的方式，常见的有get和post方式，默认是get提交

<form action="login.php" method="POST">

表设计示提交给"login.php"页面进行处理，提交方式为post

“<br/>” 规范的换行写法，必须要带双引号。

Input元素

表示输入框，有多种形式，可根据type属性来确定显示那种输入框

Type = ‘test’ 单行文本输入框

Type = ‘password’ 密码输入框

Type = ‘checkbox’ 复选框

Type = ‘radio’ 单选框

Type = ‘file’ 文件域

Type = ‘submit’ 将表单里的信息提交给表单属性action所指向的文件

Type = ‘reset’ 将表单里的信息清空，重新填写

Type = ‘image’ 图片提交按钮

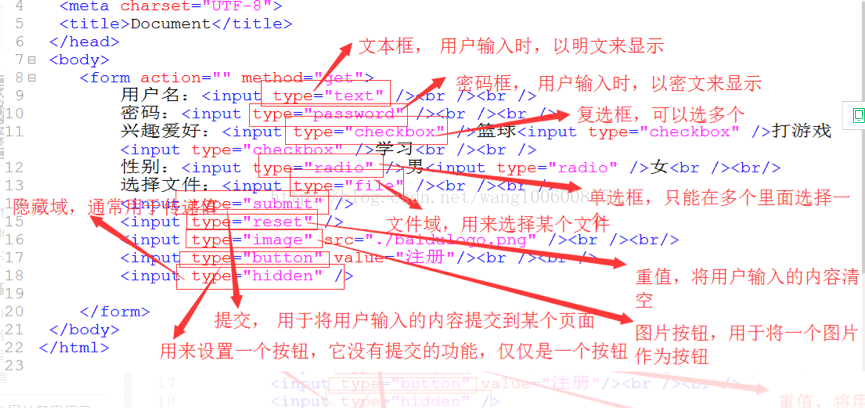
Type = ‘button’ 按钮

Type = ‘hidden’ 隐藏域

Username:<input type=’text’ style=’width:150px’ name=’cuihua

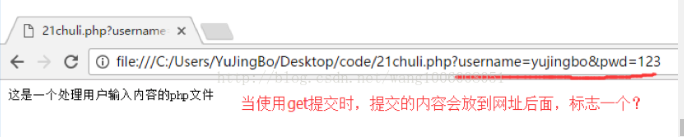
‘></input>

单行文本输入框、单元格的总体宽度 设置一个属性name属性规定input元素的名称，name属性用于对提交到服务器后的表单数据进行标识，只有设置了name属性的表单元素才能在提交表单时传递他们的值



提交方式：get、post

Get会将用户输入的内容放在地址栏里面，使用get请求不安全，而且如果没有写上name属性，那么用户输入的内容是提交不了的



Post当用户以post方式提交，用户输入的内容不会放在地址栏里面，比较安全：



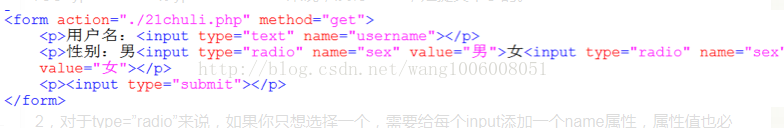
细节：

1. 每一个input标签都有一个value属性，不同类型的input，value的作用也不同

<input type=”button”>这就是一个按钮，如果不写javascript的话，按下去什么也不会发生

<input type=”submit”>这样的按钮用户点击之后会自动提交form，除非你写了javascript来阻止它，<input type=”submit” value=”login”>定义按钮的属性，按钮的名字叫”login”

1. 对于type=’text’来说、values可有可无，因为用户输入的内容都可以提交过去；
2. 对于type=’radio’或者type=’checkbox’来说，没有value是提交不了的



对于type=’radio’来说，如果你只想选择一个，需要给每个input添加一个name属性，属性值也必须一致，如果不添加，就不能满足单选框的需求

下拉列表：

暂时不写先等待：

先来说说php：（超文本预处理器）

Php是一种通用开源脚本语言，主要适用于web开发领域，php是将程序嵌入到HTML文档中去执行。PHP是一种创建动态交互性站点的强有力的服务器端脚本语言。

PHP是动态页面，其运行结果就是静态的HTML页面

静态页面：

静态页面是不能随便改动的、静态web无法连接数据库；静态web资源开发技术：HTML，静态页面的交互性较差，在功能方面有较大的限制。是设计者把页面上所有的东西都设定好作死了，然后放上去的，在任何时候看到的内容都是一样的

动态页面：

可以连接数据库，从数据库中取出数据等一系列操作后动态拼凑页面的展示内容，动态页面的内容一般都是依靠服务器端的程序来生成的，显示的内容可能不同

PHP的基本语法：

PHP脚本语言以<?php开始，以?>结束。每个代码行都必须以分号结束。

PHP输出文本指令：echo和print

PHP中的注释：

在PHP中，我们使用//来编写单行注释，或者使用/\*和\*/来编写大的注释块。

/\*

This is

A comment

Block

\*/

PHP变量：变量用于存储值，比如数字、字符串或函数的结果，这样我们就可以在脚本中多次使用它们了。一旦设置了某个值，我们就可以在脚本中重复的使用它

PHP中的所有变量都是以$符号开始的

在PHP中设置变量的正确方法是：

$var\_name = value;

创建一个存有字符的变量和一个存有数值的变量：

$txt = “hello world”;

$number = 16;

在PHP中，不需要再设置变量之前声明该变量，不必向PHP声明该变量的数据类型，根据变量被设置的方式，PHP会自动的把变量转换成正确的数据类型

变量的命名规则：

1. 以字母或者下划线”\_”开头。
2. 能包含字母数字字符以及下划线
3. 不能包含空格，如果变量名由多个单词组成，那么用下划线进行分割（$my\_string）或者以大写字母开头（$myString）。

PHP字符串：

$txt=”hello world”; 字符串”hello world”赋值给名为$txt的字符串变量

并置运算符（concatenation operator）

并置运算符（.）用于把两个字符串值连接起来：

$txt1=”hello world”

$txt2=”1234”

Echo $txt1.” ”.$txt2;

输出结果：hello world 1234 这里由于我们需要插入第三个字符串为了分割这两个变量，我们在txt1与txt2之间插入了一个空格。

Strlen()函数：

Strlen()函数用于计算字符串长度：

echo strlen(“hello world!”); 输出：12

Strpos()函数：

strpos() 函数用于在字符串内检索一段字符串或一个字符。

如果在字符串中找到匹配，该函数会返回第一个匹配的位置。如果未找到匹配，则返回 FALSE。

让我们试一下，是不是能在字符串中找到子字符串 "world"：

<?php

echo strpos("Hello world!","world");

?>

输出：6 正如您看到的，在我们的字符串中，字符串 "world" 的位置是 6。返回 6 而不是 7，是由于字符串中的首个位置的 0，而不是 1

If...else、elseif 语句：

多得不说，跟之前python中差不多这里说如果需要在条件成立或不成立时执行多行代码，应该把这些代码行包括在花括号中：

<?php

$d=date("D");

if ($d=="Fri")

{

echo "Hello!<br />";

echo "Have a nice weekend!";

echo "See you on Monday!";

}

?>

如果希望在多个条件之一成立时执行代码，请使用 elseif 语句：

if (condition)

code to be executed if condition is true;

elseif (condition)

code to be executed if condition is true;

else

code to be executed if condition is false;

PHP函数：

PHP的真正威力源于它的函数。

创建PHP函数：

1)所有的函数都使用关键词 "function()" 来开始

2)命名函数 - 函数的名称应该提示出它的功能。函数名称以字母或下划线开头。

3)添加 "{" - 开口的花括号之后的部分是函数的代码。

4)插入函数代码

5)添加一个 "}" - 函数通过关闭花括号来结束。

function writeMyName()

{

echo "David Yang";

}

使用PHP函数：

我们来看看PHP函数的使用哈！！举例？

<?php

function writeMyName()

{

echo "David Yang";

}

echo "Hello world!<br />";

echo "My name is ";

writeMyName();

echo ".<br />That's right, ";

writeMyName();

echo " is my name.";

?>

猜猜看？？输出的结果是什么？？

这个怕不好猜哦？？

输出结果：

Hello world!

My name is David Yang.

That's right, David Yang is my name.

PHP函数--添加参数：

函数名称后面有一个括号，比如 writeMyName()。参数就是在括号中规定的。通过可以添加参数，我们向函数添加更多的功能。参数类似一个变量,像上面的函数它只能输出一个静态的字符串。

<?php

function writeMyName($fname)

{

echo $fname . " Yang.<br />"; #定义的变量

}

echo "My name is ";

writeMyName("David"); 输出：My name is David Yang.

echo "My name is ";

writeMyName("Mike");输出：My name is Mike Yang.

echo "My name is ";

writeMyName("John");输出：My name is John Yang.

?>

再来看看双参数呢？

<?php

function writeMyName($fname,$punctuation)

{

echo $fname . " Yang" . $punctuation . "<br />";

}

echo "My name is ";

writeMyName("David",".");输出：My name is David Yang.

echo "My name is ";

writeMyName("Mike","!");输出：My name is Mike Yang!

echo "My name is ";

writeMyName("John","...");输出：My name is John Yang...

?>

PHP函数—返回值

其实和python差不多都是用的return

<?php

function add($x,$y)

{

$total = $x + $y;

return $total;

}

echo "1 + 16 = " . add(1,16);输出：1 + 16 = 17

?>

PHP表单处理？？

<form action="welcome.php" method="post">

Name: <input type="text" name="name" />

Age: <input type="text" name="age" />

<input type="submit" />

</form>

上面的 HTML 页面实例包含了两个输入框和一个提交按钮。当用户填写该表单并单击提交按钮时，表单的数据会被送往 "welcome.php" 这个文件。

**$\_GET 变量用于收集来自 method="get" 的表单中的值。**

$\_GET 变量用于收集来自 method="get" 的表单中的值。从带有 GET 方法的表单发送的信息，对任何人都是可见的（会显示在浏览器的地址栏），并且对发送的信息量也有限制（最多 100 个字符）

当用户点击提交按钮时，发送的 URL 会类似这样：

http://www.w3school.com.cn/welcome.php?name=Peter&age=37

**PHP获取值**

<?php echo $\_GET["name"]; ?>.<br />

<?php echo $\_GET["age"]; ?>

## $\_REQUEST 变量

PHP 的 $\_REQUEST 变量可用来取得通过 GET 和 POST 方法发送的表单数据的结果

<?php echo $\_REQUEST["name"]; ?>.<br />

You are <?php echo $\_REQUEST["age"]; ?> years

**$\_POST 变量用于收集来自 method="post" 的表单中的值**

$\_POST 变量用于收集来自 method="post" 的表单中的值。从带有 POST 方法的表单发送的信息，对任何人都是不可见的（不会显示在浏览器的地址栏），并且对发送信息的量也没有限制

<form action="welcome.php" method="post">

Enter your name: <input type="text" name="name" />

Enter your age: <input type="text" name="age" />

<input type="submit" />

</form>

当用户点击提交按钮，URL 不会含有任何表单数据，看上去类似这样：

http://www.w3school.com.cn/welcome.php

"welcome.php" 文件现在可以通过 $\_POST 变量来获取表单数据了（请注意，表单域的名称会自动成为 $\_POST 数组中的 ID 键）：

Welcome <?php echo $\_POST["name"]; ?>.<br />

You are <?php echo $\_POST["age"]; ?> years old!

建立PHP代码与mysql之间的连接：在您能够访问并处理数据库中的数据之前，您必须创建到达数据库的连接

通过mysql\_connect() 函数完成。

$con = mysql\_connect(servername,username,password)

|  |  |
| --- | --- |
| **参数** | **描述** |
| servername | 可选。规定要连接的服务器。默认是 "localhost:3306"。 |
| username | 可选。规定登录所使用的用户名。默认值是拥有服务器进程的用户的名称。 |
| password | 可选。规定登录所用的密码。默认是 ""。 |

**注释：**虽然还存在其他的参数，但上面列出了最重要的参数

在下面的例子中，我们在一个变量中 ($con) 存放了在脚本中供稍后使用的连接。如果连接失败，将执行 "die" 部分：

<?php

$con = mysql\_connect("localhost","peter","abc123");

if (!$con) 判断是否有错误

{

die('Could not connect: ' . mysql\_error());

}

关闭连接，请使用 mysql\_close() 函数。

在上面连接了数据库之后，通过 mysql\_select\_db() 函数选取数据库。

## **定义和用法**

mysqli\_select\_db() 函数用于更改连接的默认数据库。

## **语法**

mysqli\_select\_db(*connection,dbname*)*;*

|  |  |
| --- | --- |
| **参数** | **描述** |
| *connection* | 必需。规定要使用的 MySQL 连接。 |
| *dbname* | 必需，规定要使用的默认数据库。 |

两个参数必须要有，

## **例子**

<?php

$con = mysql\_connect("localhost", "hello", "321");

if (!$con) 与数据库建立连接

{

die('Could not connect: ' . mysql\_error());

}

$db\_selected = mysql\_select\_db("test\_db", $con);

选择的表的名称 mysql连接 $con

if (!$db\_selected)

{

die ("Can\'t use test\_db : " . mysql\_error());

}

mysql\_close($con);

?>

选择了数据库表后我们就需要查询选择的表的内容哦

## **定义和用法**

mysql\_query() 函数执行一条 MySQL 查询

### **语法**

mysql\_query(query,connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| query | 必需。规定要发送的 SQL 查询。注释：查询字符串不应以分号结束。 |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

<?php

$con = mysql\_connect("localhost","mysql\_user","mysql\_pwd");

if (!$con)

{

die('Could not connect: ' . mysql\_error());

}

$sql = "SELECT \* FROM Person"; 定义的变量 选择查询的表的内容

mysql\_query($sql,$con);

定义一个变量：$result=mysql\_query($sql,$con);

# mysql\_num\_rows() 函数

## **定义和用法**

mysql\_num\_rows() 函数返回结果集中行的数目。

### **语法**

mysql\_num\_rows(data)

|  |  |
| --- | --- |
| **参数** | **描述** |
| data | 必需。结果集。该结果集从 mysql\_query() 的调用中得到。 |

### **说明**

mysql\_num\_rows() 返回结果集中行的数目。此命令仅对 SELECT 语句有效

比如上面的例子：

$result=mysql\_query($sql,$con);

我们要显示这个查询的结果有多少行？？

mysql\_num\_rows($result)

Mysql\_error()函数

返回上一个mysql操作产生的文本错误信息，本函数返回上一个 MySQL 函数的错误文本，如果没有出错则返回 ''（空字符串）。

语法：

Mysql\_error(connection)

|  |  |
| --- | --- |
| **参数** | **描述** |
| connection | 可选。规定 SQL 连接标识符。如果未规定，则使用上一个打开的连接。 |

来看一个整体的连接，

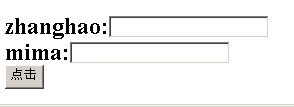
HTML表：

<form action="my.php" method="POST">

zhanghao:<input type="text" style="width:160px" name="shengshuai"></input><br>

mima:<input type="password" style="width:160px" name="pot"></input><br>

<input type="submit" value="点击"></input><br>



My.PHP

<?php

$a = $\_POST['shengshuai'];

echo $a."<br>";

$b = $\_POST['pot'];

$b =md5($b);

echo $b."<br>";

1：获取来自前段的用户名和密码（定义变量a获取zhanghao值，定义变量b获取mima值，并输出值）；

$con = mysql\_connect("localhost","hsc","");

if (!$con)

{

die('could not connect:'.mysql\_error());

}

else{

echo "connect mysql success.<br>";

}

2：建立PHP代码与mysql之间的连接（用到函数mysql\_connect()）

$shujuku=mysql\_select\_db("class88",$con);

if(!$shujuku)

{

die("can\'t use class88:".mysql\_error());

}

else{

echo "use class88 csuccess.<br>";

}

$chaxun="select \* from jianli where username='$a' and password='$b'";（数据库的属性值）

$fanhui=mysql\_query($chaxun,$con);

if(mysql\_num\_rows($fanhui)==1)

{

echo "welcome to my site,".$fanhui."<br>";

}

else{

echo "you can't login my site,".$fanhui."<br>";

}

3：建立PHP代码与数据库里面的某个表连接

mysql\_close();

?>

其实我这个写的不是很规范，变量名是随意取的看起来就是个新手啊啊啊 认识一点单词 database:数据库、error:错误、connect:连接、query:疑问，询问/’kuiweri/。