

第5章 PHP对字符串的处理

- PHP对字符串的处理的完美支持，是PHP的强大功能之一。PHP提供了数十个用来处理字符串的内置函数，使用这些函数，可以在PHP程序中很方便地完成对字符串的各种操作。本章为读者介绍PHP开发中一些比较常用的字符串处理函数。

5.1 对字符串做分割和合并

- 在PHP程序中，可以使用函数explode用一个字符串分割另一个字符串，该函数的语法如下所示。
- `array explode(string $separator, string $str [,int $limit])`
- 函数explode使用字符串\$separator（该函数的第1个参数）作为边界点，来分割字符串\$str（该函数的第2个参数）。该函数的返回值是一个数组，字符串被分割后，分割的部分会存入该数组。该函数最后一个参数\$limit是可选参数。如果设置了该参数，则返回的数组中最多有limit个元素，最后一个元素包含字符串\$str的所有剩余部分。



5.2.1 比较两个字符串大小

- PHP提供函数strcmp对两个字符串比较大小。该函数语法如下所示。
- `int strcmp(string $str1, string $str2);`
- 函数strcmp接受两个参数，它们是待比较的字符串。该函数返回值是整数，如果字符串\$str1和字符串\$str2相等，则函数返回0，如果字符串\$str1小于字符串\$str2，则函数返回-1，如果字符串\$str1大于字符串\$str2，则函数返回值大于0。



5.2.2 字符串的替换

- 在实际开发中，有时需要将字符串中的某些字符串替换成其他字符串，这可以通过PHP的内置函数 `str_replace` 来完成。该函数的语法如下所示。
- `string str_replace(string $search, string $replace, string $str);`
- 该函数将字符串 `$str` 中的 `$search` 部分全部替换成字符串 `$replace`，并且返回替换后的字符串。



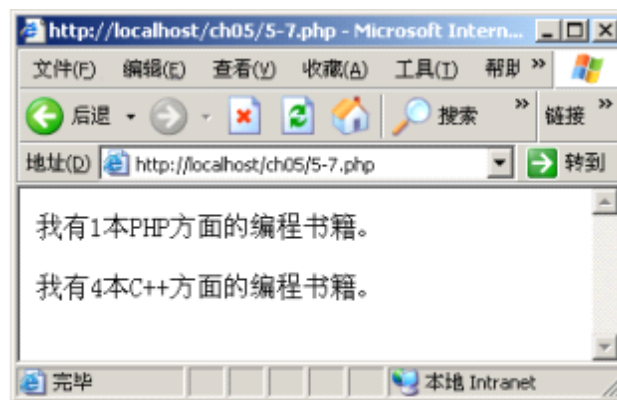
5.3 输出打印字符串

- 向Web页面输出字符串是PHP程序最常用的操作之一。事实上，在本节之前的几乎示例程序中都有输出字符串的代码。在PHP输出字符串的方式有如下两种。
- （1）使用print。它的用法如下所示。
- `int print(string $str);`
- print将字符串\$`str`输出，并且总是返回1。严格意义上讲，print不是函数，它和echo一样，只是PHP的语言结构，因此，可以不使用括号将其参数括住。
- （2）使用echo。使用echo输出字符串，在此之前已经大量使用，这里不再举例说明。



5.4 格式化字符串

- 将字符串转换成某种特定的格式，叫做字符串的格式化。比如一本书的价格是39元，一般都会格式化成“39.00”的形式。对字符串进行格式化，是程序开发中经常遇到处理操作之一。PHP语言提供函数sprintf，将字符串格式化，该函数的语法如下所示。
- `string sprintf(string $format, mixed $args);`
- 参数\$format是要转换的格式，该参数通常以百分号%开始，到转换字符为止。例如，%1.2f就是一个转换格式，其中f就是转换字符，它表示将字符串格式化成浮点数。这种用法，对于有C语言基础的读者来说，一定不会陌生。该函数返回格式化后的字符串。



sprintf 的简单应用



5.5 获取字符串中的一个子串

- 获取字符串中的一个子串是指，获取这个字符串的某一部分。在PHP中，可以使用函数substr来获取字符串的某一部分。该函数的语法如下所示。
- `string substr(string $str, int $start, int $length);`
- 函数substr将字符串\$str中，从位置\$start的字符开始，截取长度为\$length的字符串，然后返回该字符串。需要注意的是，字符串的第1个字符的位置是0，第2个字符的位置是1，以此类推。



5.6 删除字符串末尾部分的空白字符

- PHP程序有时需要去掉字符串中的空白字符，比如用户从网页提交数据时，在文本框输入内容后，多敲入了一些空格，而PHP处理程序很可能要将这些数据存入数据库，这时，不希望将这些没有实际意义的空白字符存入数据库，因此要将这些空白字符处理掉。
- PHP提供了一些函数，分别可以处理字符串末尾、开头的空白字符，也提供了同时处理字符串两端空白字符的函数。本节先介绍处理字符串末尾部分空白字符的内置函数。在PHP中，可以使用函数`rtrim`删除字符串末尾部分的空白字符。该函数的语法如下所示。
- `string rtrim(string $str);`



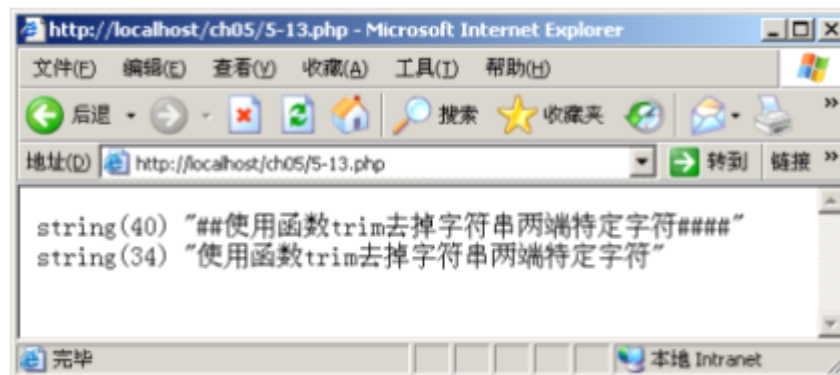
5.7 删除字符串开始部分的空白字符

- 和函数rtrim对应的是函数ltrim，该函数将字符串开始部分的空白字符删除掉。从这个函数的命名可以看出，ltrim中的字母l代表的是left，一个字符串的left就是它的开始部分，所以，ltrim表示去掉字符串开始部分的空白字符。同理，rtrim中的首字母r表示right。这样，初学者就容易记住这两个函数及其用法了。函数ltrim的语法如下所示。
- `string ltrim(string $str);`
- 该函数将字符串\$str开始部分的空白字符删除，返回开始部分没有空白的字符串。



5.8 删除字符串两头的空白字符

- 如果希望删除字符串两头的空白字符，可以使用函数trim，该函数的语法如下所示。
- `string trim(string $str);`



是用 trim 删除字符串两端特定字符



5.9 获取字符串的长度

- 字符串的长度一般是指，组成字符串的字符的数量。PHP提供函数strlen来获取一个字符串的长度。该函数的语法如下所示。
- `int strlen(string $str);`
- 该函数返回字符串\$str的长度。



使用 strlen 获取字符串长度



5.10 对字符串做大小写转换

- PHP提供两个函数分别完成对字符串中字母的大小写转换，它们是：
- `strtoupper($str)`，将字符串`$str`中的字母全部转换为大写字母。
- `strtolower($str)`，将字符串`$str`中的字母全部转换为小写字母。

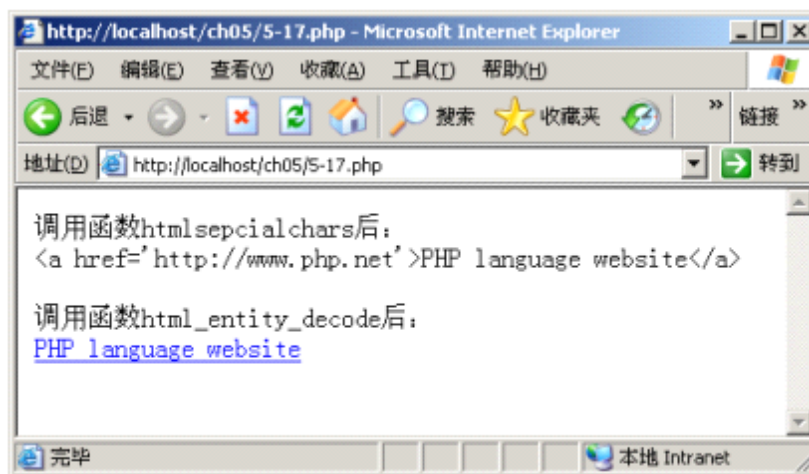


对字符串做大小写转换



5.11 处理含有HTML标记的字符串

- 使用PHP进行Web页面开发，经常会遇到处理HTML标记的问题。PHP提供的函数`htmlentities`完成对字符串中HTML标记的处理，该函数可将字符串中有关字符转换成HTML实体，即HTML字符编码。该函数的语法如下所示。
- `string htmlentities(string $str);`
- 函数`htmlentities`将字符串`$str`中的HTML标记（如`<`、`>`等）转换成相关编码的形式。

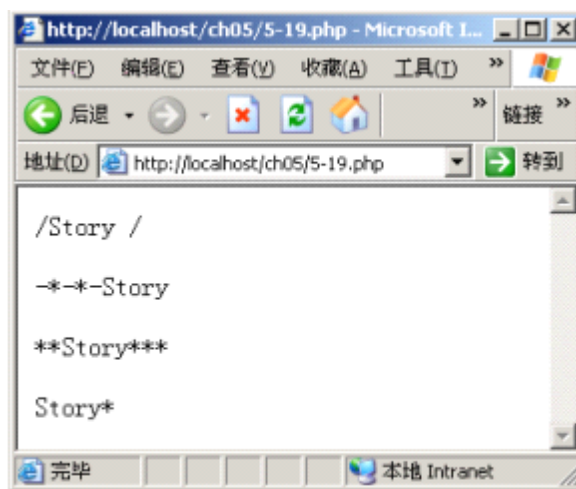


使用 `html_entity_decode` 转换 HTML 实体



5.12 多次生成一个字符串

- 函数`str_repeat()`可以用来重复生成一个字符串，该函数的语法如下所示。
- `string str_repeat (string $input, int $num)`
- 函数`str_repeat()`有两个参数，第1个参数`$input`是要重复生成的字符串，第2个参数`$num`是要重复的次数。参数`$num`的取值必须大于等于0，如果`$num`设置为0，该函数会返回一个空字符串。



使用字符串对另一字符串进行补全



5.13 将字符串分析到变量

- 这小节所说的分析字符串，是指分析 URL 中的查询字符串。
例如一个 URL 是：

`http://www.somesite.com/index.php?i=000&key=1a2b3c&cnt=10`，那么该 URL 的查询字符串就是

`i=000&key=1a2b3c&cnt=10`。PHP 提供函数 `parse_str()` 可以分析类似的字符，并将分析结果赋值给变量。函数 `parse_str()` 的语法如下所示。

- `void parse_str(string $str [, array $arr])`
- 本函数可将 URL 的查询字符串参数 `$str` 解析，返回的变量名及值就和查询字符串中的名称及值相对应。该函数第 2 个参数一个指定的数组，是可选的。如果指定这个数组参数，那么函数 `parse_str()` 将分析字符串得到变量作为数组索引、值作为数组元素存入该数组。



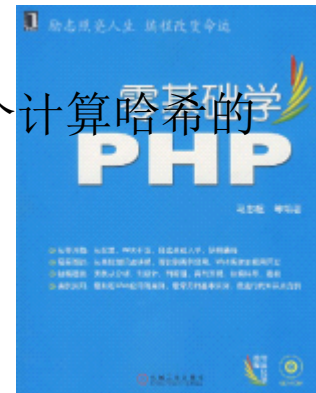
5.14 转换字符串到数组

- PHP提供的函数`str_split()`，可将字符串转换成一个数组，即将数组分割成几个部分，每个部分的值作为数组的元素。该函数的语法如下所示。
- `array str_split(string $str [, int $split_length])`
- 该函数将字符串`$str`做分割，将分割后的字符串各部分存入数组，并将数组返回。可选参数`$split_length`用来指定分割字符串长度。如果没有指定该参数，函数将把字符串`$str`按一个字符的长度分割。



5.15 计算字符串的Hash（哈希）

- 所谓的哈希（Hash）也叫散列，它就是指把任意长度的输入（又叫做预映射），通过散列算法，变换成固定长度的输出，该输出就是散列值。
- Hash主要用于信息安全领域的加密算法，它把一些不同长度的信息转化成杂乱的128位的编码，这个编码,叫做Hash值。也可以说，Hash就是找到一种数据内容和数据存放地址之间的映射关系。常见的Hash算法有MD5，SHA1等。
- PHP中提供了对应的函数，完成对字符串做MD5哈希计算的函数，也提供函数完成SHA1哈希计算的函数，这两个函数分别是md5()和sha1()，下面分别加以简要说明。函数md5()的语法如下所示。
- `string md5(string $str)`
- 该函数用来计算参数字符串\$str的MD5哈希，另外一个计算哈希的函数sha1()的语法如下所示。
- `string sha1(string $str)`



5.16 小结

- 本章通过介绍一些主要的字符串处理函数，来学习在PHP程序中如何完成对字符串的操作。这些函数都是比较基本的，读者务必掌握。本章讲到的函数包括：
- 将字符串分割后存入数组的函数`explode()`。
- 将数组中的元素合并成字符串的函数`implode()`。
- 比较两个字符串大小的函数`strcmp()`。
- 字符串替换函数`str_replace()`。
- 输出字符串的语言结构`echo`和`print`。
- 字符串格式化函数`sprintf()`。



5.16 小结

- 截取字符串的函数 `substr()`。
- 删除字符串开头和末尾空白字符的函数 `ltrim()`、`rtrim()`、`trim()`。
- 获取字符串长度的函数 `strlen()`。
- 对字符串做大小写转换的函数 `strtolower()` 和 `strtoupper()`。
- 处理含有HTML标记的函数 `htmlentities()`、`htmlspecialchars()` 和 `html_entity_decode()`。
- 重复生成字符串的函数 `str_repeat()`。
- 分析字符串到变量的函数 `parse_str()`。
- 转换字符串到数组的函数 `str_split()`。
- 计算字符串Hash的函数 `md5()`。和 `sha1()`。

