

學習要點:

1. 字符串格式化
2. 操作子字符串
3. 字符串比較
4. 查找替換字符串
5. 處理中文字符

三、字符串比較

到目前為止，我們已經用過“=”號來比較兩個字符串是否相等。使用 PHP 可以進行一些更複雜的比較。這些比較分為兩類：部分匹配和其他情況。

字符串的排序: `strcmp()`、`strcasemp()`和 `strnatcmp()`

該函數需要兩個進行比較的參數字符串。如果這兩個字符串相等，該函數返回 0，如果按字典排序 `str1` 和 `str2` 後面(大於 `str2`)就返回一個正數，如果 `str1` 小於 `str2` 就返回一個負數。這個函數是區分大小寫的。

函數 `strcasecmp()`除了不區分大小寫之外，其他和 `strcmp()`一樣。

函數 `strnatcmp()`及與之對應的不區分大小寫的 `strnatcasecmp()`函數是在 PHP4 中新增的。
這兩個函數按“自然排序”比較字符串，所謂自然排序是按人們習慣的順序進行排序。

```
<?php
    Echo strcmp('a','b');
?>
```

使用 `strspn()`函數返回一個字符串中包含有另一個字符串中字符的第一部分的長度。也就是求兩個字符串之間相同的部分。

```
<?php
    Echo strspn('gmail','yc60.com@gmail.com');
?>
```

使用 `strlen()`函數測試字符串的長度

可以使用函數 `strlen()`來檢查字符串的長度。如果傳給他一個字符串，這個函數將返回字符串的長度。例如: `strlen("hello")`將返回 5.

```
<?php
    Echo strlen('This is a Teacher!');
?>
```

確定字符串出現的頻率: `substr_count()` 返回一個字符串在另一個字符串中出現的次數。

```
<?php
    Echo substr_count('yc60.com@gmail.com','c');
?>
```

四、查找替換字符串

通常，我們需要檢查一個更長的字符串中是否含有一個特定的子字符串。這種部分匹配通常比測試字符串的完全等價更有用處。

在字符串中查找字符串: `strstr()`、`strchr()`、`strrchr()`和 `stristr()`

函數 `strstr()` 是最常見的，他可以用於在一個較長的字符串專供查找匹配的字符串或字符。請注意，函數 `strchr()` 和 `strstr()` 完全一樣。

```
<?php
    Echo strstr('yc60.com@gmail.com','@');
?>
```

函數 `strstr()` 有兩個變體。第一個變體是 `stristr()`，他幾乎和 `strstr()` 一樣，其區別在於不區分字符大小。對於我們的只能表單應用程序來說，這個函數非常有用，因為用戶可以輸入“delivery”、“Delivery”和“DELIVERY”。

第二個變體是 `strchr()`，他也幾乎和 `strstr()` 一樣，只不過是 `strstr()` 的別名。

查找字符串的位置: `strpos()`、`strrpos()`。

函數 `strpos()` 和 `strrpos()` 的操作和 `strstr()` 類似，但他不是返回一個子字符串，而返回子字符串 `needle` 在字符串 `haystack` 中的位置。更有趣的是，現在的 PHP 手冊建議使用 `strpos()` 函數代替 `strstr()` 函數來查看一個子字符串在一個字符串中出現的位置，因為前者的運行速度更快。

```
<?php
    Echo strrpos('yc60.com@gmail.com','c');
?>
```

替換字符串: `str_replace`、`str_ireplace`、`substr_replace()`

```
<?php
    Echo str_replace('@','#','yc60.com@gmail.com');
    Echo substr_replace('yc60.com@gmail.com',###,0,5);
```

?>