

??????

Двойные кавычки разименовывают переменную. Если нет пробелов со следующим текстом, то нужно в фигурных скобках:

```
$n = 12;  
echo "You are {$n}th number";
```

Одинарные - как есть.

Формат heredoc

```
$cleriheW = <<EndOI0uote  
Съешь ещё этих мягких  
французских булок,  
да выпей чаю.  
EndOI0uote;  
echo $cleriheW;
```

Вывод строк:

```
echo "one", "two";  
print("some");
```

Есть еще printf, print_r, var_dump

Функции:

strlen(\$string);	длина строки
trim(), ltrim(), rtrim()	обрезание лишних пробелов слева и справа, второй необ. параметр - список символов, удаляемых вместе с пробелами
strtolower() strtoupper()	всю строку
ucfirst()	одну первую букву всей строки
ucwords()	первую букву каждого слова
htmlentities() htmlspecialchars{}	Преобразование сырой строки в html-безопасную строку
strip_tags()	Удаление html тегов

<code>get_meta_tags()</code>	возвращает массив МЕТА-тегов HTML страницы, указанный в виде URL или локального файла
<code>rawurlencode()</code> <code>rawurldecode()</code>	кодирование/декодирование строки согласно URL-соглашению RFC 3986
<code>addslashes()</code> <code>stripslashes()</code>	экранировать/убрать обратный слеш перед: одинарные кавычки, двойные кавычки, NUL-байты и обратные слешы
<code>strrev()</code>	порядок символов на обратный
<code>str_repeat()</code>	Повтор строки n раз
<code>str_pad()</code>	заполняет одну строку другой строкой
<code>explode(separator, string [,limit]);</code> <code>implode(separator, array);</code>	разбитие / сборка

Сравнение и поиск строк:

<code>soundex()</code>	Сравнение степени похожести звучания (не написания) двух строк
<code>metaphone()</code>	Более мягкое сравнение степени похожести звучания (не написания) двух строк
<code>similar_text()</code>	возвращает число одинаковых символов, которые есть в двух переданных строках. В третий параметр, если он задан, заносится степень похожести двух строк, выраженная в процентах
<code>levenshtein()</code>	Вычисляет расстояние Левенштейна между двумя строками
<code>substr(string, start [, length]);</code>	
<code>substr_count(большая_строка, маленькая_строка)</code>	сколько раз меньшая строка встречается в большей строке
<code>substr_replace(original, new, start [, length]);</code>	<p>заменяет часть строки <code>original</code> начинающуюся с символа с порядковым номером <code>start</code> и длиной <code>length</code> строкой <code>new</code> и возвращает результат.</p> <p>Если не задан четвертый аргумент, <code>substr_replace()</code> удаляет текст, начиная с позиции <code>start</code> до конца строки.</p> <p><code>length</code> в 0 для вставки без удаления "<code>''</code>" для удаления без вставки</p> <p>Если <code>start</code> - отрицательное число, замена начинается с символа с порядковым номером <code>start</code>, считая от конца строки</p> <p>Отрицательное значение аргумента <code>length</code> определяет количество символов от конца строки, на котором заканчивается замена</p>
<code>strpos()</code> <code>strrpos()</code>	первая / последняя позиция

strspn(), strcspn()	длина участка. в начале строки, полностью соответствующего маске
parse_url()	массив составляющих URL parse_url("http://me:secret@example.com/cgi-bin/board?user=fred"); ([scheme] => http [host] => example.com [user] => me [pass] => secret [path] => /cgi-bin/board [query] => user=fred)

Revision #3

Created 3 July 2024 16:12:26 by Admin

Updated 4 July 2024 06:27:08 by Admin