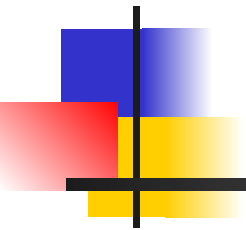


PHP. Типы данных.

Строки



Тип строка: string

- Смысл: наборы символов, представленных байтами
- В PHP нет встроенной поддержки Unicode (и UTF8)
- Длина строк (string) – до 2Гб
- Простейший способ определить строковое значение - заключить строку в одинарные кавычки 'так мы задаем строку'
- В строке (string) может быть много строк (line)
Для перевода строки используйте управляющие последовательности \n – новая строка и \r – перевод каретки
'этот текст \n\r записан в двух строках ';



Способы задания значения типа "строка"

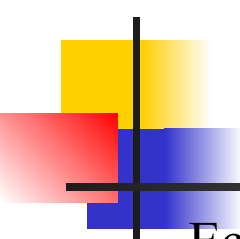
- Задание строковых значений – одна из самых используемых возможностей языка.
- Строковое значение может быть определено несколькими способами, предоставляющими различный круг возможностей
- Рассмотрим следующие:
 - с помощью одинарных кавычек
 - с помощью двойных кавычек
 - heredoc - синтаксисом

Задание строковых значений с помощью одинарных кавычек

- Заключение набора символов в одинарные кавычки (символ ') – простейший способ задать значение строки
- Для включения в строку кавычки, её надо экранировать обратной косой чертой, для включения обратной косой – продублируйте её
- Никакие другие управляющие последовательности в строках , заданных с помощью одинарных кавычек не используются
- В результате выполнения следующих инструкций:

```
$str1='text'; $str2='\text\'; $str3=\\text\\;  
echo $str1,'+', $str2,'+', $str3;
```
- В поток вывода будет отправлено:

```
text+'text'+\text\
```



Задание строковых значений с помощью двойных кавычек

- Если строка заключена в двойные кавычки ("), РНР распознает большее количество управляющих последовательностей для специальных СИМВОЛОВ
-

Управляющие последовательности при задании строк

- Если строка заключена в двойные кавычки ("), PHP распознает большее количество управляющих последовательностей для специальных символов, в том числе:
 - \n - новая строка (LF или 0x0A (10) в ASCII)
 - \r - возврат каретки (CR или 0x0D (13) в ASCII)
 - \t - горизонтальная табуляция (HT или 0x09 (9) в ASCII)
 - \\ - обратная косая черта
 - \\$ - знак доллара
 - \" - двойная кавычка

Обработка переменных

- Задание значений строк с помощью двойных кавычек позволяет обрабатывать переменные.

- Пример, демонстрирующий смысл обработки переменных:
код

```
$good='мяч'; $color='белый'; $size=40;  
echo "товар $good имеет $color цвет и размер $size см.";
```

выведет:

```
товар мяч имеет белый цвет и размер 40 см.
```

- Помните, что интерпретатор старается взять максимальное число символов для разрешения имени переменной. Следующий код:
"товар \$goods имеет \$colors цвет и размер \$sizes см.";
будет искать переменные `$goods`, `$colors`, `$sizes` и если не найдет их
выведет:
товар \$goods имеет \$colors цвет и размер \$sizes см.

Задание строковых значений

HEREDOC - синтаксисом

- Heredoc-синтаксис задает целые блоки
- Начинает блок комбинация трех угловых скобок и идентификатора блока:
`<<<HERE_NAME`
Далее, непосредственно после имени блока должен следовать перевод строки
- Заканчивается идентификатором и символом (;), указывающим на конец инструкции:
`HERE_NAME;`
последняя строка конструкции не может содержать дополнительно никаких символов, включая пробелы.
- В строке могут быть использованы управляющие последовательности и обработка переменных
- Можно создавать многострочные конструкции и не экранировать двойные кавычки

Пример задания строковых значений HEREDOC - синтаксисом

- `$str0='!!!';`

`$str = <<<EOD`

Пример строки, охватывающей несколько строчек,
с использованием heredoc-синтаксиса.

Мы можем использовать 'одинарные' и "двойные" кавычки
без экранирования, а также обрабатывать переменные `$str0`
`EOD`;

- Пример строки, охватывающей несколько строчек,
с использованием heredoc-синтаксиса.

Мы можем использовать 'одинарные' и "двойные"
кавычки без экранирования, а также обрабатывать
переменные !!!