

Раздаточный материал к шестой лекции

Строки

Пример: Heredoc синтаксис определения строки

```
<?php
$str = <<<EOD
Пример строки,
охватывающей несколько строчек,
с использованием heredoc-синтаксиса.
EOD;
?>
```

Особенности строк

- К строкам не применим оператор “+” для конкатенации, для слияния нескольких строк используется оператор точка “.”.
- Массивы всегда преобразуются в строку **"Array"**.
- Объекты всегда преобразуются в строку **"Object"**. Если вы хотите получить имя класса требуемого объекта, используйте функцию `get_class()`.
- Ресурсы всегда преобразуются в строки со структурой **"Resource id #1"**, где **1** - это уникальный номер ресурса (resource), присвоенный ему PHP во время выполнения. Если вы хотите получить тип ресурса, используйте `get_resource_type()`.
- Интересной особенностью PHP является возможность сериализации экземпляра класса или массива в строку и десериализации из строки (Функции `serialize()` и `unserialize()`!).
- Не ожидайте получить код символа, преобразовав его в целое (как вы могли бы сделать, например, в C). Для преобразования символов в их коды и обратно используйте функции `ord()` и `chr()`.

Основные функции для работы со строками

addslashes	Экранирует спецсимволы в стиле языка C
addslashes	Экранирует спецсимволы в строке
bin2hex	Преобразует бинарные данные в шестнадцатиричное представление
crc32	Вычисляет CRC32 для строки
crypt	Необратимое шифрование (хэширование)
echo	Выводит одну или более строк
explode	Разбивает строку на подстроки
fprintf	Записывает отформатированную строку в поток
html_entity_decode	Преобразует HTML сущности в соответствующие символы
htmlentities	Преобразует символы в соответствующие HTML сущности.

htmlspecialchars_decode	Преобразует специальные HTML-сущности обратно в соответствующие символы
htmlspecialchars	Преобразует специальные символы в HTML сущности
implode	Объединяет элементы массива в строку
ltrim	Удаляет пробелы из начала строки
md5	Возвращает MD5 хэш строки
money_format	Форматирует число как денежную величину
nl2br	Вставляет HTML код разрыва строки перед каждым переводом строки
parse_str	Разбирает строку в переменные
similar_text	Вычисляет степень похожести двух строк
str_replace	Заменяет строку поиска на строку замены
str_shuffle	Переставляет символы в строке
str_split	Преобразует строку в массив
strip_tags	Удаляет HTML и PHP тэги из строки
strlen	Возвращает длину строки
strpbrk	Ищет в строке любой символ из заданного набора
substr_replace	Заменяет часть строки
substr	Возвращает подстроку

Работа с файлами и директориями

Основные функции для работы с файлами

chmod	Изменяет режим доступа к файлу или каталогу
copy	Копирует файл
dirname	Возвращает имя каталога из указанного пути
disk_free_space	Получить размер доступного пространства в каталоге
fclose	Закрывает дескриптор файла
feof	Проверяет, достигнут ли конец файла
fflush	Сброс буфера вывода в файл
fgetc	Считывает символ из файла
fgetcsv	Читает строку из файла и производит разбор данных CSV
fgets	Читает строку из файла
fgetss	Прочитать строку из файла и отбросить HTML-теги
file_exists	Проверить наличие указанного файла или каталога
file_get_contents	Получить содержимое файла в виде одной строки
file_put_contents	Записать строку в файл
file	Читает содержимое файла и помещает его в массив
fileatime	Получить время последнего доступа к файлу
filesize	Получить размер файла
filetype	Получить тип файла
flock	Портируемое рекомендательное запираание файлов
fopen	Открывает файл или URL
fseek	Устанавливает смещение в файловом указателе
is_dir	Определяет, является ли файл директорией
is_file	Определяет, является ли файл обычным файлом
rename	Переименовывает файл или директорию
rewind	Сбрасывает курсор у файлового указателя
rmdir	Удаляет директорию
tempnam	Создаёт файл с уникальным именем
touch	Устанавливает время доступа и модификации файла
unlink	Удаляет файл

Пример: Простая запись в файл:

```
$outStr      = «Some string to write»;
$outFile     = fopen("data.js","w+");
fwrite($outFile,$outStr);
fclose($outFile);
```

is_dir()

Функция определяет, является ли файл директорией.

```
<?
var_dump(is_dir('a_file.txt')) . "\n";
var_dump(is_dir('bogus_dir/abc')) . "\n";

var_dump(is_dir('..')); //на одну директорию выше
?>
```

Основные используемые при работе с каталогами функции

chdir	Сменить каталог
chroot	Сменить корневой каталог
dir	Класс directory
closedir	Освободить дескриптор каталога
getcwd	Получить имя текущего рабочего каталога
opendir	Открыть каталог
readdir	Получить элемент каталога по его дескриптору
rewinddir	Сбросить дескриптор каталога
scandir	Получить список файлов и каталогов, расположенных по указанному пути

Пример: Получение списка всех файлов в директории

```
<?php
// Обратите внимание, что оператор !== не существовал до версии 4.0.0-RC2

if ($handle = opendir('/path/to/files')) {
    echo "Дескриптор каталога: $handle\n";
    echo "Файлы:\n";

    /* Именно этот способ чтения элементов каталога является правильным. */
    while (false !== ($file = readdir($handle))) {
        echo "$file\n";
    }

    /* Этот способ НЕВЕРЕН. */
    while ($file = readdir($handle)) {
        echo "$file\n";
    }

    closedir($handle);
}
?>
```

Замечание: readdir() также возвращает элементы с именами . и ... Если они не нужны, то их нужно отбрасывать.

Работа с заголовком HTML документа

Согласно [RFC 2616](#) заголовок гипертекстового документа может содержать множество разнообразной информации.

Примеры:

Простой редирект, однако это не простое перемещение по новому адресу, но также и возвращается статусный код 302 (Redirect).

```
<?php
  header("Location: http://www.tsi.lv/");
  exit();
?>
```

Пример вызова требования аутентификации:

```
<?php
header('WWW-Authenticate: Negotiate');
header('WWW-Authenticate: NTLM', false);
?>
```

Также модификация заголовка используется для того, чтобы указать, что происходит вывод документа в отличном от HTML формате.

К примеру, если скрипт динамически генерирует PDF документ, то его скорее всего нужно выдать пользователю вместе с диалогом сохранения файла, для этого можно воспользоваться следующим кодом:

```
<?php
//Указание типа выводимого документа
header('Content-type: application/pdf');

//Имя документа по умолчанию
header('Content-Disposition: attachment; filename="downloaded.pdf");

//Выдача содержимого документа
readfile('original.pdf');
?>
```

Пример управления кэшированием документа

```
<?php
header("Cache-Control: no-cache, must-revalidate"); // HTTP/1.1
header("Expires: Mon, 26 Jul 1997 05:00:00 GMT"); // Date in the past
?>
```

Важно понимать, что заголовки всегда отсылаются клиентскому браузеру, а функция **header()** позволяет переписать некоторые их значения.