

Еріс РНР #2

Рома Лапин, Evercode Lab
@memphys, roma@evercode-lab.com

var_dump

Выводит информацию о переменных.

```
<?php
```

```
$values = array(14, 42, array("text", "test", "best"));
```

```
var_dump($values);
```

GET

<http://localhost/epicphp/test.php?name=Vazzzya>

```
<?php
```

```
$input = $_GET['name'];
```

```
printf('Hello %s', $input);
```

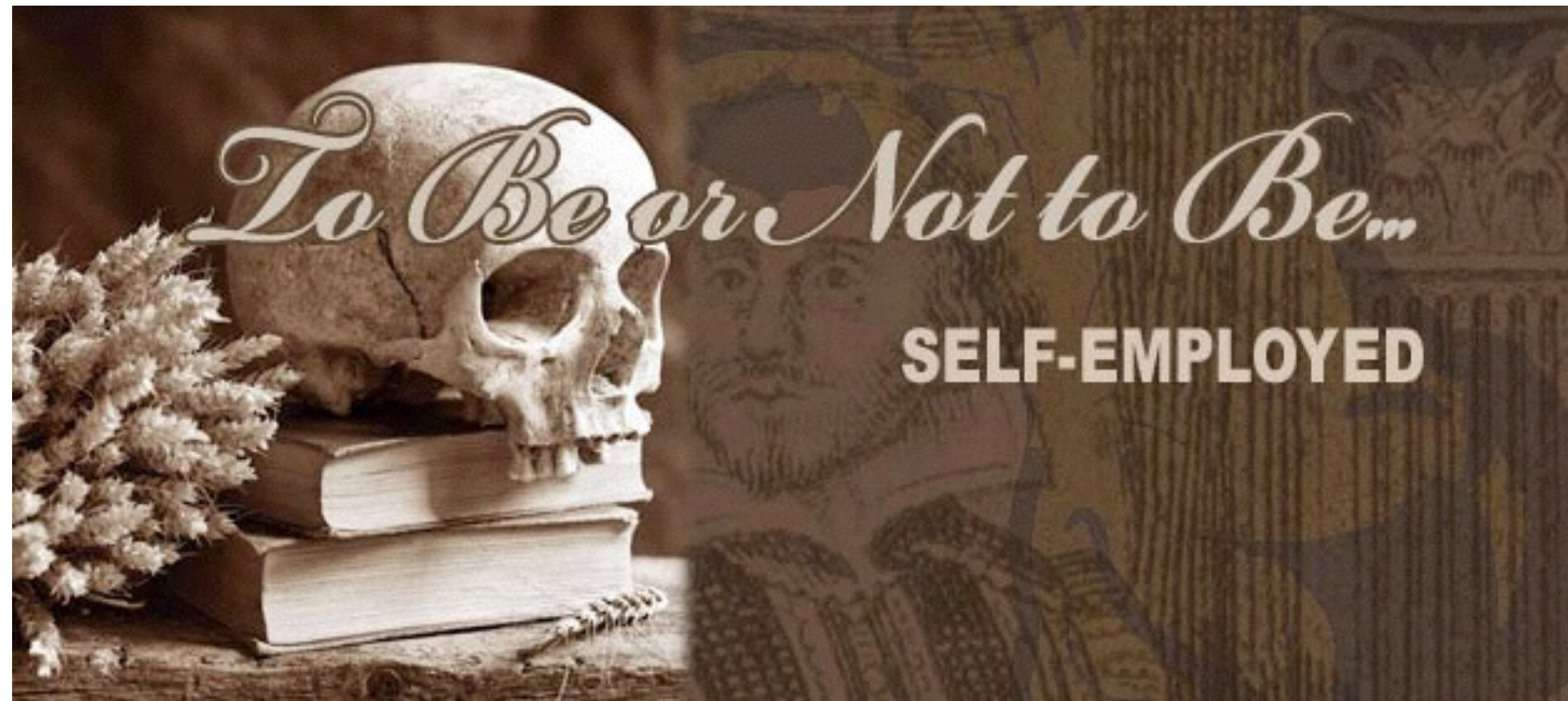
Почти фреймворк

```
<?php
```

```
$input = isset($_GET['name']) ? $_GET['name'] : 'World';
```

```
header('Content-Type: text/html; charset=utf-8');
```

```
printf('Hello %s', htmlspecialchars($input, ENT_QUOTES, 'UTF-8'));
```



to require or to include,
that is the question

to require or to include

require и include дают возможность включать в код одного файла код другого до этапа интерпретации

В случае ошибки

- require выдаст fatal error (E_COMPILE_ERROR) и прекратит дальнейшую работу скрипта
- include только выдаст warning (E_WARNING) и работа скрипта продолжится

to require or to include

Подключаемый файл увидит переменные из области видимости родительского файла. А функции и классы из подключаемых файлов сразу получают глобальную видимость.

to require or to include

```
include 'footer.php';  
require 'content.php';
```

to require or to include

```
// works  
if ((include 'vars.php') == 'OK') {  
    echo 'OK';  
}
```

to require or to include

```
<?php
// won't work
// evaluated as include(('vars.php') == 'OK'),
// i.e. include("")
if (include('vars.php') == 'OK') {
    echo 'OK';
}
```

once or not once

- Как следует из названия `require_once` и `include_once` исполняются лишь один раз независимо от количества вызовов
- Оптимизация
- Избегаем ошибок с переопределением

who the hell cares

Прямое подключение файлов сходит на нет под
напором праведного автолоадинга.

Пробуем

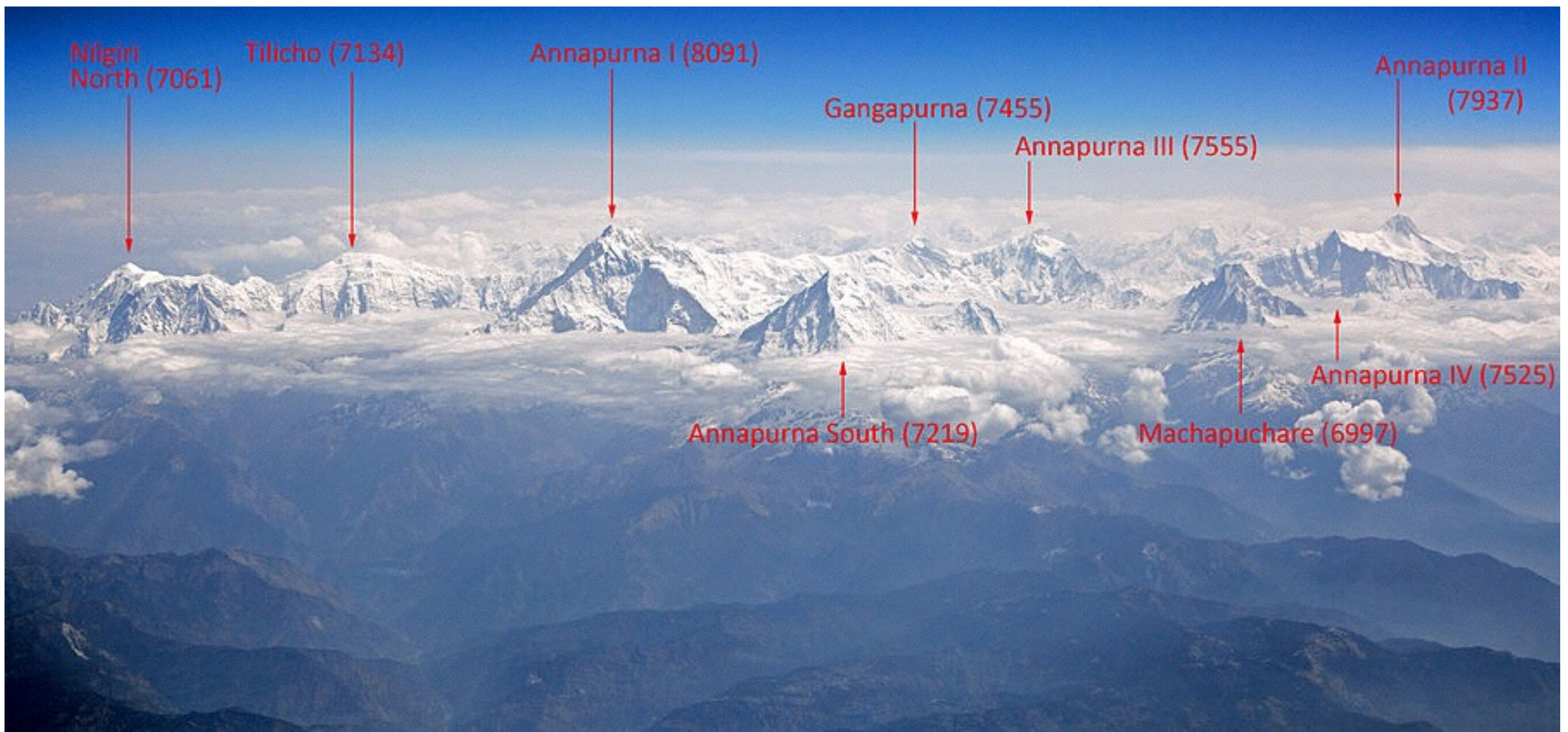
- Создайте файлы: index.php, header.php, footer.php
- В header.php и footer.php сделайте простую верстку, соответствующую названию
- Подключите файлы header и footer из index.php

Усложняем

- Создайте файлы: `main.php`, `contacts.php`, `about.php`
- В зависимости от `get`-параметра `page` в строке браузера подключайте соответствующий файл из `index.php`

Массивы

Горные массивы?



Нет, просто массивы

Массивы

Упорядоченное отображение, которое устанавливает соответствие между значением и ключом. Может быть использован как собственно массив, список (вектор), хэш-таблицу (являющуюся реализацией карты), словарь, коллекцию, стэк, очередь и, возможно, что-то еще. Так как значением массива может быть другой массив PHP, можно также создавать деревья и многомерные массивы.

Массивы

```
$values = array(  
    'key' => 'value',  
    'key2' => 'value2',  
    'key3' => 'value3',  
);
```

```
// Начиная с PHP 5.4  
$array = [  
    "foo" => "bar",  
    "bar" => "foo",  
];
```

Массивы

Пустой массив

```
$values = array();
```

// Начиная с PHP 5.4

```
$array = [];
```

Массивы

Добавляем элементы в массив

```
$values = [];
```

```
$values[] = 'new element';
```

Массивы

Доступ к элементам массива может быть осуществлен с помощью синтаксиса `$array[key]`.

Ключи могут быть либо `integer`, либо `string`.

```
$array['somekey'];  
$array[1050];
```

Массивы

Всегда заключайте в кавычки строковый литерал в индексе ассоциативного массива.

`$array[string_key]` — плохо

`$array['string_key']` — хорошо

Массивы

В С-подобных языках программирования, индекс массива традиционно начинается с 0.

```
$values = array('hello', 15, 42, 'tomato');  
echo $values[0]; // выведет hello
```

Массивы

Числовые ключи — могут быть заданы явно, либо будут присвоены автоматически. Отлично подходят в тех случаях, когда вам не важно, как проиндексированы данные в массиве.

```
$values = array('hello', 15, 42, 'tomato');  
foreach($values as $value) {  
    echo $value;  
}
```

Массивы

Ассоциативные массивы — часто используются при работе с базами данных, либо в тех случаях, когда нужны явные ассоциации «ключ => значение»

```
$values = array(
    'name' => 'Neo', 'version' => 'wat?',
);
echo '<table>';
foreach($values as $key => $value) {
    echo '<tr><td>' . $key . '</td><td>' . $value . '</td></tr>';
}
echo '</table>';
```

Массивы

Многомерные массивы — массивы массивов (массивов массивов). Могут содержать множество данных, но в то же время могут быть сложны для операции с ними и доступа к ним.

```
$values = array(  
    'name' => 'Neo',  
    'friends' => array(  
        'Trinity',  
        'Morpheus',  
    ),  
);
```

Массивы

- Ключи, содержащие только цифры, будут приводиться к типу `integer`
- Ключи массивов регистрозависимые, но типонезависимые
- Если несколько элементов в объявлении массива используют одинаковый ключ, то только последний будет использоваться, а все другие будут перезаписаны.

Массивы

Функция `unset()` позволяет удалять ключи массива. Обратите внимание, что массив НЕ будет переиндексирован.

```
$values = array('hello', 15, 42, 'tomato');  
unset($values[0]);  
echo $values[1]; // 15
```

Задание

Создайте массив, содержащий информацию по всем вашим одноклассникам: имя, фамилию, родной город, email, id в vk.com и небольшой список интересов (3-4).

Выведите его на страницу с помощью `var_dump()`.

Указатель

У массивов есть внутренний указатель (курсор).
Он показывает положение текущего выбранного
элемента.

Итерируем

- `foreach()` — создает копию массива, т.е. не будет менять оригинальный массив
- `foreach($array as $key => &$value) { ... }` — передает значения по ссылке

Задание

В массиве ваших одноклассников запишите для каждого в поле с именем фамилию, в поле с фамилией — родной город, в поле с городом — имя.

Результаты можно вывести с помощью `var_dump`.

Задание

Попробуйте выполнить предыдущее задание, используя передачу значения по ссылке.

```
foreach($array as $key => &$value) { ... }
```

Задание

Если осталось время, то напишите функцию, которая будет выводить результаты (произвольные массивы) в виде таблицы, где названия столбцов — ключи массива первой записи. Если среди значений есть вложенный массив, то его значения надо выводить в ячейку через запятую.

Итерируем

- `current()` — говорит за себя
- `next()` — перемещает указатель на один элемент вперед и возвращает значение нового элемента
- `prev()` — перемещает указатель на один элемент назад и возвращает значение нового элемента

Итерируем

- `reset()` — сбрасывает указатель на начало массива и возвращает значение первого элемента
- `end()` — перемещает указатель на последний элемент массива
- `key()` — возвращает значение ключа текущего элемента массива

Сортируем

- `bool sort (array &$array [, int $sort_flags = SORT_REGULAR])`
- `SORT_REGULAR`
- `SORT_NUMERIC`
- `SORT_STRING`

Сортируем

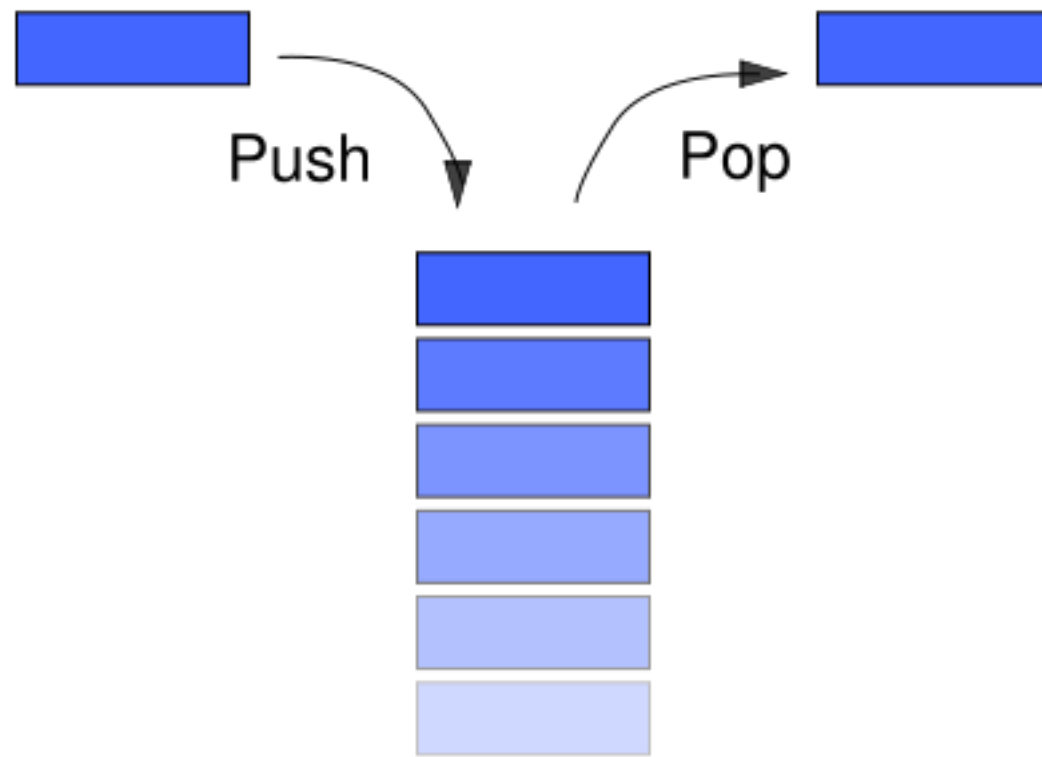
- `rsort()`
- `ksort()`
- `asort()`
- `usort()`
- `bool shuffle (array &$array)`

Задание

Перемешайте массив ваших одноклассников случайным образом.

Затем отсортируйте массив в алфавитном порядке по фамилиям.

Stack



Stack

- LIFO – Last Input First Output
- `int array_push (array &$array , mixed $var [, mixed $...])`
- `mixed array_pop (array &$array)`

Очередь

- FIFO – First Input First Output
- `int array_unshift (array &$array , mixed $var [, mixed $...])`
- `mixed array_shift (array &$array)`



ОЧЕРЕДЬ

Это место, где люди действительно боятся потерять друг друга.

Функции работы с массивами

- `array_keys`
- `array_values`
- `array_merge`
- `array_key_exists`
- `array_combine`

Функции работы с массивами

- list
- count
- array_diff
- array_slice
- in_array

Задание

Разделите массив с информацией по вашим одноклассникам на два так, чтобы в одном было на 1 человека больше, чем во втором, и выведите два отдельных списка с именами и фамилиями.

Задание

Простой постраничный вывод.

На странице выводим только по два человека. В зависимости от GET параметра page мы показываем следующую или предыдущую пары. Если page такой, что значений уже не хватает, надо вывести сообщение, что значений больше нет.

Дополнение: вывести список ссылок на страницы, состоящий автоматически из количества записей в массиве.

Работа со строкама

Факты о строках

\$firstFact = 'Строки в одиночных кавычках интерпретируются как есть, и переменные, например, \$thisOne, и специальные escape последовательности типа \n не будут интерпретироваться';

Факты о строках

\$secondFact = “А в двойных кавычках
переменные интерпретируются. Вот тут
выведется предыдущий текст:\n \$firstFact”;

Факты о строках

```
$thirdFact = <<<GREETING
```

```
А можно еще и так выпендриться, $username!  
GREETING;
```

Факты о строках

PHP воспринимает строки как последовательность байтов. По умолчанию он знает только про те кодировки, в которых один байт — одна буква. Что для русского языка плохо.

Факты о строках

```
<?php
$a    = 'car'; // $a is a string
$a[0] = 'b';   // $a is still a string
echo $a;      // bar

// не прокатит с русским текстом
```

Конкатенация



Интерактив

- Создайте контактную форму для сайта: имя, email, возраст, комментарий
- `method="get"`
- создайте два PHP файла:
 - оба подключаются из `index.php`
 - один — выводит форму
 - второй — выводит данные формы


Интерактив

Соедините все полученные поля в одну переменную `$message` в формате:

Имя_поля: значение
Имя_поля2: значение2

Выведите в скрипте обработки эту переменную.


Функции

- php.net/manual/en/ref.strings.php
- `int strlen ($string)`
- `int strpos ($haystack, $needle, $offset)` 
- `substr`
- <http://php-guide.evercode.com/pages/strings-and-patterns.html>

Функции

- trim
- nlbr
- htmlspecialchars
- htmlentities
- strtolower
- strtoupper



Функции

- explode 
- implode

Задание

- Напишите функцию получающую домен из email-адреса
- Напишите функцию, принимающую массив, которая убирает у каждого элемента массива окружающие пробелы, приводит все к нижнему регистру, соединяет все элементы в строку (формат как для контактной формы).

Escape Sequences

-  `\n` linefeed (LF or 0x0A (10) in ASCII)
-  `\r` carriage return (CR or 0x0D (13) in ASCII)
- `\t` horizontal tab (HT or 0x09 (9) in ASCII)
- `\\` backslash
- `\$` dollar sign

ПРИНТ-ШМИНТ

- `int print (string $arg)`
- `int printf (string $format [, mixed $args [, mixed $...]])`
- `string sprintf (string $format [, mixed $args [, mixed $...]])`
- `int vprintf (string $format , array $args)` -
ВЫВОДИТ МАССИВ, КАК ФОРМАТИРОВАННУЮ СТРОКУ

Спецификации преобразования

- % [дополняющий_символ] [-] [ширина] [.точность]
тип
- d – интерпретируется как целое число и
выводится как десятичное число
- f – интерпретируется как число двойной точности
и выводится как число с плавающей точкой
- s – интерпретируется как строка и выводится как
строка



ARE YOU GANGSTERS?

no, we are russians!

Русский язык

- Multibyte String
- <http://php.net/manual/en/book.mbstring.php>
- `int mb_strlen (string $str [, string $encoding])`
- Те же функции, но с возможностью указать кодировку
- `mb_internal_encoding("UTF-8");`

Задания

- Дана строка в которой записаны слова через пробел. Необходимо посчитать количество слов с четным количеством символов.

Задания

- Дана строка в которой записаны слова через пробел. Необходимо посчитать количество слов.

Задания

- Функция, которая каждый второй символ строки приводит к верхнему регистру

Задания

- Дана строка в которой записаны слова через пробел. Необходимо перемешать в каждом слове все символы в случайном порядке кроме первого и последнего.

Задания

- Дана строка в которой записан путь к файлу.
Необходимо найти имя файла без расширения.

Задания

- <http://lurkmore.to/PHP>

ДЗ

- Составьте описание проекта, который вы бы хотели сами построить. Сайт, сервис, игра, что угодно. На следующих занятиях лучшая идея будет примером при разборе работы с базами данных.

ДЗ

- Зарегистрируйтесь на github.com. На следующем занятии познакомимся с git, а все задания будем хранить на этом сайте.

ДЗ

- Внимательно изучите работу функций `htmlspecialchars` и `htmlentities`. В чем между ними разница?

ДЗ

- Сделайте простой сайт, объединяющий все упражнения со строками. Есть форма для ввода текста, есть select с кратким описанием функции. В зависимости от выбора в select отправленный текст должен преобразовываться и выводиться на страницу сайта.

php.ini



Как и где искать

- `phpinfo()`;
- `php --ini` или `php.exe --ini`
- может быть разным для сервера и командной строки
- может называться не `php.ini` (редко)
- файлов может быть больше одного

Как и где искать

- `/etc/php5/apache2/php.ini`
- `/etc/php5/cli/php.ini`
- `/etc/php5/fpm/php.ini`
- <http://php.net/manual/en/ini.core.php>

Суровая правда

Настройки могут быть разными или работать по-разному для сервера и командной строки, а также для локальной разработки и для рабочего сервера (вывод ошибок, логирование, лимиты памяти).

Меняем настройки налету

- `ini_set();`
- `ini_get();`



Настройки ошибок

- `error_reporting` — что показываем
- `display_errors` — а показываем ли?
- `log_errors` — если не показываем, то хотя бы записываем?

error_reporting

- <http://www.php.net/manual/en/errorfunc.constants.php>
- Задается константами или целым числом
- локально используйте E_ALL для > 5.4 и E_ALL & E_STRICT для < 5.4

error_reporting

- E_ERROR – Fatal run-time errors
- E_WARNING – Run-time warnings (non-fatal errors)
- E_PARSE – Compile-time parse errors.
- E_NOTICE – Run-time notices

Поэкспериментируем

- Создайте папку errors
- На каждый тип ошибки создайте отдельный скрипт. Ошибка должна выводиться.
- Попробуйте поменять в каждом скрипте `display_errors` и `error_reporting`