
Разработка многостраничного сайта на PHP

ТЕМА 4.2 РАСШИРЕННЫЕ МЕТОДЫ
ЗАНЯТИЕ № 5 - ПРАКТИКА

Тема занятия – Расширенные методы

Цель занятия –

Продолжить изучение расширенных методов на практике

Актуализация

На прошлом занятии мы решали задачи по РНР, используя расширенные методы.

Сегодня мы продолжим решать новые задачи, узнавать новые расширенные методы, используемые в РНР.

Содержание занятия

- 1) Задача № 1
- 2) Задача № 2
- 3) Задача № 3
- 4) Задача № 4
- 5) Задача № 5
- 6) Задача № 6
- 7) Задача № 7
- 8) Задача № 8
- 9) Задача № 9
- 10) Задача № 10

Введение

PHP является одним из самых популярных языков в мире, который используется для создания веб-приложений, включая множество онлайн-магазинов, социальных сетей и блогов. Одним из основных преимуществ PHP является его гибкость и возможность расширения функциональности.

В этой практической части мы рассмотрим несколько расширенных методов работы с PHP, которые обеспечивают высокую производительность и функциональность веб-приложений.

Задача № 1

Напишите программу, которая будет принимать имя входа в каталоги и выводить на экран список всех файлов в этом каталоге, отсортированных по дате последней версии.

Решение

```
<?php
// Получаем имя входного каталога из аргумента командной строки
$dir = $argv[1];

// Получаем список файлов в каталоге
$files = scandir($dir);

// Создаем пустой массив для хранения метаданных файлов
$fileMetadata = [];

// Получаем метаданные для каждого файла и сохраняем их в массиве $fileMetadata
foreach ($files as $file) {
    if ($file !== '.' && $file !== '..') {
        $path = $dir . '/' . $file;
        $fileMetadata[$path] = [
            'name' => $file,
            'last_modified' => (new DateTime())->setTimestamp(filemtime($path))
        ];
    }
}

// Сортируем файлы по дате последней модификации
usort($fileMetadata, function($a, $b) {
    return $a['last_modified'] <=> $b['last_modified'];
});

// Выводим список файлов на экран
foreach ($fileMetadata as $metadata) {
    echo $metadata['name'] . PHP_EOL;
}
?>
```

Задача № 2

Написана программа, которая будет получать на два входа и выводить пропорции между ними в днях, часах, минутах и секундах. Даты нужно получать в качестве аргументов командной строки в формате "ГГГГ-ММ-ДД ЧЧ:ММ:СС".

Решение

```
<?php
if (count($argv) != 3) {
    echo "Usage: php time_diff.php <date1> <date2>\n";
    exit(1);
}

$date1 = new DateTime($argv[1]);
$date2 = new DateTime($argv[2]);

$interval = $date1->diff($date2);

echo "Difference between dates:\n";
echo $interval->format("%a days, %h hours, %i minutes, %s seconds\n");
?>
```

Задача № 3

Напишите программу, которая будет получать входной текстовый файл и выводить на экран список всех возможных слов в этом файле в алфавитном порядке.

Решение

```
<?php
if (count($argv) != 2) {
    echo "Usage: php word_list.php <filename>\n";
    exit(1);
}

$filename = $argv[1];

if (!file_exists($filename)) {
    echo "File not found: $filename\n";
    exit(1);
}

$text = file_get_contents($filename);
$words = preg_split('/\W+/', $text, -1, PREG_SPLIT_NO_EMPTY);
$unique_words = array_unique($words);
sort($unique_words);

echo "Words in file:\n";
echo implode("\n", $unique_words) . "\n";
?>
```

Задача № 4

Написана программа, которая будет получать на два входных массива чисел и находить совокупность элементов массива, умноженных на их индекс. Массивы нужно получать в качестве аргументов командной строки, каждый элемент массива разделяет запятой.

Решение

```
<?php
if (count($argv) != 3) {
    echo "Usage: php array_index_multiply.php <array1> <array2>\n";
    exit(1);
}

$array1 = explode(",", $argv[1]);
$array2 = explode(",", $argv[2]);

if (count($array1) != count($array2)) {
    echo "Error: arrays must have the same length\n";
    exit(1);
}

$result = array_map(function($value, $index) {
    return $value * $index;
}, $array1, array_keys($array2));

echo "Result: [" . implode(", ", $result) . "]\n";
?>
```

Задача № 5

Написать программу, которая будет получена на входе в строку и выводить на экран самое длинное слово в этой строке. Строку нужно получать в качестве аргумента командной строки.

Решение

```
<?php
if (count($argv) != 2) {
    echo "Usage: php longest_word.php <string>\n";
    exit(1);
}

$string = $argv[1];
$words = preg_split('/^[A-Za-z]+/', $string); // Разделяем строку на слова
$longest_word = '';

foreach ($words as $word) {
    if (strlen($word) > strlen($longest_word)) { // Если текущее слово длиннее,
                                                //чем предыдущее найденное слово
        $longest_word = $word;
    }
}

echo "Longest word in the string is: $longest_word\n";

?>
```

Задача № 6

Написать программу, которая будет получать на входе строку и выводить ее на экран, заменяя все цифры соответствующей буквой латинского алфавита (например, цифра 1 - буква A, цифра 2 - буква B и т.д.). Строку нужно получать в качестве аргумента командной строки.

Решение

```
<?php
if (count($argv) != 2) {
    echo "Usage: php replace_numbers.php <string>\n";
    exit(1);
}
$input_string = $argv[1];
$alphabet = range('A', 'Z');

for ($i = 0; $i <= 9; $i++) {
    $letter = $alphabet[$i];
    $input_string = str_replace((string)$i, $letter, $input_string);
}
echo $input_string . "\n";
?>
```

//Пример работы программы

```
$ php replace_numbers.php "My 2 favorite numbers are 7 and 9"
My B favorite numbers are G and I
```

Задача № 7

Напишите программу, которая будет получать на два входа числа и выводить на экран наибольший общий делитель (НОД) и наименьшее общее кратное (НОК) этих чисел. Числа необходимо получить в качестве аргументов командной строки.

Решение

```
<?php
if (count($argv) != 3) {
    echo "Usage: php gcd_lcm.php <num1> <num2>\n";
    exit(1);
}

$num1 = gmp_init($argv[1]);
$num2 = gmp_init($argv[2]);

$gcd = gmp_strval(gmp_gcd($num1, $num2));
$lcm = gmp_strval(gmp_lcm($num1, $num2));

echo "GCD: $gcd\n";
echo "LCM: $lcm\n";
?>
```

Задача № 8

Написана программа, которая будет получать на два входа массива и находить все элементы, которые встречаются в сборе массивов.

Массивы необходимо получить в качестве аргументов командной строки, каждый элемент массива разделить запятой.

Решение

```
<?php
if (count($argv) != 3) {
    echo "Usage: php array_intersection.php <array1> <array2>\n";
    exit(1);
}

$array1 = explode(',', $argv[1]);
$array2 = explode(',', $argv[2]);

$intersection = array_intersect($array1, $array2);

echo "Intersection of arrays:\n";
print_r($intersection);

?>
```

//Пример использования

```
php array_intersection.php "1,2,3,4,5" "3,4,5,6,7"
```

//Результат

Intersection of arrays:

Array

(

[2] => 3

[3] => 4

[4] => 5

)

Задача № 9

Написать программу, которая будет получена по входу и проверке, является ли она панграммой (строка, содержащая все буквы букв). Строку нужно получать в качестве аргумента командной строки.

Решение

```
<?php
if (count($argv) != 2) {
    echo "Usage: php pangram_check.php <input_string>\n";
    exit(1);
}

$input_string = strtolower($argv[1]);

$alphabet = range('a', 'z');

foreach ($alphabet as $letter) {
    if (strpos($input_string, $letter) === false) {
        echo "Input string is not a pangram.\n";
        exit(0);
    }
}

echo "Input string is a pangram!\n";

?>
```

Задача № 10

Напишите программу, которая будет получать массив входных чисел и находить медиану этого массива (серединное число, если отсортировать массив). Массив необходимо получить в качестве аргумента командной строки, каждый элемент массива разделить запятой.

Решение

```
<?php
if (count($argv) != 2) {
    echo "Usage: php median.php <numbers>\n";
    exit(1);
}

// Получаем массив чисел из аргумента командной строки
$numbers = explode(' ', $argv[1]);

// Сортируем массив чисел
sort($numbers);

// Получаем количество элементов в массиве
$count = count($numbers);

// Находим медиану в зависимости от того, является ли число элементов четным или нечетным
if ($count % 2 == 0) {
    $median = ($numbers[$count / 2 - 1] + $numbers[$count / 2]) / 2;
} else {
    $median = $numbers[( $count - 1) / 2];
}
echo "Median: " . $median . "\n";

?>
```

```
//Пример использования
php median.php 4,2,1,3,5
```

```
//Результат
Median: 3
```

Заключение

В заключении хочу отметить, что изучение расширенных методов работы с РНР помогает разработчикам создавать более сложные и функциональные веб-приложения. В ходе практической части мы изучили множество полезных инструментов и методов, которые позволили повысить качество и производительность приложений.

Кроме того, использование современных фреймворков и создание классов позволяет ускорить процесс разработки и повысить ее эффективность. Я надеюсь, что практическая часть была полезной для вас и помогла расширить кругозор в области программирования на РНР.

Рефлексия

Сегодня были решены еще задачи по теме: «Расширенные методы в РНР»

Ответьте на несколько вопросов:

1. Какая задача была самая интересная?
2. Какая задача показалась наиболее сложной?
3. Какую задачу вы поняли?

A decorative rectangular frame with a light beige background. The frame is adorned with four corner ornaments, each featuring a pink flower, purple swirls, and green leaves. The text is centered within this frame.

**Спасибо
за
внимание**