

!ДОМАШНЯЯ РАБОТА по теме «Основы синтаксиса PHP»

Вдумчиво прочитать учебный материал, отладить скрипты на понимание особенностей использования переменных, констант, управляющих конструкций, устно ответить на контрольные вопросы [1, 372, 353-375]

Лабораторная работа № 25

ТЕМА «ИСПОЛЬЗОВАНИЕ СТАНДАРТНЫХ ФУНКЦИЙ ДЛЯ РАБОТЫ С МАССИВАМИ И СТРОКАМИ»

Цель работы: ознакомиться со стандартными функциями обработки массивов и научиться использовать их при создании и обработке массивов; рассмотреть функции работы со строками, изучить два подхода к решению задач по обработке строк.

СОЗДАНИЕ И ОБРАБОТКА МАССИВОВ [1, 396-397]

ФУНКЦИИ ПОИСКА ЭЛЕМЕНТОВ [1, 397]

ФУНКЦИИ ДОБАВЛЕНИЯ И УДАЛЕНИЯ ЭЛЕМЕНТОВ [1, 398]

ФУНКЦИИ ПЕРЕБОРА ЭЛЕМЕНТОВ [1, 398-400]

ФУНКЦИИ РАЗМЕРА МАССИВА [1, 400]

ФУНКЦИИ СОРТИРОВКИ [1, 401]

ПОЛЕЗНЫЕ ФУНКЦИИ [1, 402-403]

1. **Изучить** теоретический материал по теме использования стандартных функций обработки массивов в php-скриптах [1,396-397]. **Обратить внимание** на особенности их использования.
2. **Отладить** примеры по использованию функций:
 - а) для работы с массивами [1, 403-404];
 - б) сортировка массива по длине [1. 404].

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

3. **Продумать скрипт** по обработке массива, отладить. **Записать в тетрадь условие задачи.**

```
<?php
$x = [3, 5, 6, 10, 1];
$max = $x[0];
for ($i = 1; $i < 5; $i++) {
    if ($x[$i] > $max) {
        echo $x[$i], '<br>';
        $max = $x[$i];
    }
}
echo $max;
```

4. **Проанализировать** сравнительные характеристики внутренних сортировок вставками, выбором и пузырьковой. **Быстродействие каждой из сортировок записать в тетрадь.**

```
<?php
$n = 10;
for ($i = 0; $i < $n; $i++) {
    $array[$i] = mt_rand(0, 100);
}
```

```

}
function printArray($array, $currentKey = -1)
{
    echo '<tr>';
    foreach ($array as $key => $value) {
        $style = '';
        if ($key == $currentKey) $style = ' style="background-
color:green"';
        echo "<td$style>$value</td>";
    }
    echo '</tr>';
}

echo 'Сортировка выборкой<table>';
$sortingArray = $array;
$count = count($sortingArray);
printArray($sortingArray);
$compare = 0; //Количество сравнений
$change = 0; //Количество обменов
for ($currentKey = 0; $currentKey < $count-1; $currentKey++) {
    $minKey = $currentKey;
    $minVal = $sortingArray[$currentKey];
    for ($nextKey = $minKey + 1; $nextKey < $count;
$nextKey++){
        $compare++;
        if ($sortingArray[$nextKey] < $minVal) {
            $minKey = $nextKey;
            $minVal = $sortingArray[$nextKey];
        }
    }
    $compare++;
    if ($minVal < $sortingArray[$currentKey]) {
        $sortingArray[$minKey] = $sortingArray[$currentKey];
        $sortingArray[$currentKey] = $minVal;
        $change++;
    }
    printArray($sortingArray, $currentKey);
}
echo "</table>Количество сравнений $compare, обменов
$change<br>";

echo 'Пузырьковая сортировка<table>';
$sortingArray = $array;
$count = count($sortingArray);
$compare = 0; //Количество сравнений
$change = 0; //Количество обменов
printArray($sortingArray);
for ($currentKey = 0; $currentKey < $count-1; $currentKey++) {
    $flag = false; //Для проверки, были ли замены
    for ($nextKey = $count - 1; $nextKey > $currentKey;
$nextKey--){
        $compare++;
        if ($sortingArray[$nextKey] < $sortingArray[$nextKey -
1]) {
            $temp = $sortingArray[$nextKey];

```

```

        $sortingArray[$nextKey] = $sortingArray[$nextKey -
1];
        $sortingArray[$nextKey - 1] = $temp;
        $flag = true;
        $change++;
    }
}
printArray($sortingArray, $currentKey);
if (!$flag) break;
}
echo "</table>Количество сравнений $compare, обменов
$change<br>";

echo 'Сортировка вставками<table>';
$sortingArray = $array;
$count = count($sortingArray);
printArray($sortingArray);
$compare = 0; //Количество сравнений
$change = 0; //Количество обменов
for ($currentKey = 1; $currentKey < $count; $currentKey++) {
    $temp = $sortingArray[$currentKey];
    for ($nextKey = $currentKey; $nextKey > 0; $nextKey--) {
        $compare++;
        if ($sortingArray[$nextKey - 1] > $temp) {
            $sortingArray[$nextKey] = $sortingArray[$nextKey -
1];
            $change++;
        } else {
            break;
        }
    }
    $sortingArray[$nextKey] = $temp;
    printArray($sortingArray, $currentKey);
}
echo "</table>Количество сравнений $compare, обменов
$change<br>";

```

5. Продумать скрипты обработки массивов, отладить. В комментарии записать условие задачи.

```

<?php
$x = [[1, 2, 3],
      [4, 5, 6],
      [7, 8, 9]];
//комментарий к работе скрипта
$s = 0;
for ($i=0; $i < 3; $i++) {
    for ($j=0; $j < 3; $j++) {
        $s += $x[$i][$j];
    }
}
echo $s, '<br>---<br>';
//комментарий к работе скрипта
$s = 0;
for ($j=0; $j < 3; $j++) {

```

```

        for ($i=0; $i < 3; $i++) {
            $s += $x[$i][$j];
        }
    }
    echo $s, '<br>---<br>';
    //комментарий к работе скрипта
    for ($i=0; $i < 3; $i++) {
        $s = 0;
        for ($j=0; $j < 3; $j++) {
            $s += $x[$i][$j];
        }
        echo $s, '<br>';
    }
    echo '---<br>';
    //комментарий к работе скрипта
    for ($j=0; $j < 3; $j++) {
        $s = 0;
        for ($i=0; $i < 3; $i++) {
            $s += $x[$i][$j];
        }
        echo $s, '<br>';
    }
}

```

Во всех заданиях для самостоятельной работы записать и отладить программы с использованием стандартных функций обработки массивов.

ОБРАБОТКА СТРОК [1, 405-47]. ФУНКЦИИ ДЛЯ РАБОТЫ СО СТРОКАМИ [1, 408-416].

6. Изучить теоретический материал по теме использования стандартных функций обработки строк в php-скриптах [1, 405-417]. Обратить внимание на то, как кодировка документа может влиять на использование функций.
7. Отладить пример по использованию стандартных функций обработки строк [1. 416].

ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

8. Продумать и отладить скрипт по обработке строк для юникода. Изменить код программы для обработки строк однобайтовой кодировки и протестировать.

```

<?php
//аналог str_replace для юникода
function mb_str_replace($haystack, $search, $replace, &$count
= null, $encoding = '')
{
    if (empty($encoding)) $encoding = mb_internal_encoding();
    $len_sch = mb_strlen($search, $encoding);
    $count = 0;
    $result = '';
    while (($offset=mb_strpos($haystack, $search, 0,
$encoding)) !== false){
        $result .= mb_substr($haystack,0, $offset, $encoding)
. $replace;
        $haystack = mb_substr($haystack,$offset +
$len_sch,null, $encoding);
        $count++;
    }
}

```

```
    return $result . $haystack;
}
```

```
$text = "Для замены фрагмента текста в юникоде встроенные  
функции замены str_replace, str_ireplace могут дать неверные  
результаты";  
echo mb_str_replace($text, 'замены', '<b>замены</b>',  
$count);  
echo '<br>количество замен - ', $count;
```

9. **Оформить отчет** по лабораторной работе с рассуждениями о выполнении заданий. Отлаженные php-скрипты должны быть подключены к сайту.

10. **Защитить** лабораторную работу.

ДОМАШНЕЕ ЗАДАНИЕ

1. **Ответить на контрольные вопросы (письменно)** [1, 404-405, 417].
2. **Задания по обработке массивов**
 - a) Определить произведение элементов массива, расположенных между максимальным и минимальным элементами.
 - b) Преобразовать массив таким образом, чтобы в первой его половине располагались элементы, стоявшие в исходном массиве на нечетных позициях (1, 3, 5, ...), а во второй половине — элементы, стоявшие на четных позициях (0, 2, 4, ...).
 - c) В двумерном массиве определить номера столбцов, не содержащих ни одного нулевого элемента, и вычислить произведения элементов каждого из этих столбцов.
3. **Задания по обработке строк**

Для текста, например,

\$text =
'What is Symfony. Symfony is a set of PHP Components, a Web Application framework, a Philosophy, and a Community - all working together in harmony.
Symfony Framework. The leading PHP framework to create websites and web applications.
Built on top of the Symfony Components.
Symfony Components. A set of decoupled and reusable components on which the best PHP applications are built, such as Drupal, phpBB, and eZ Publish.
Symfony Community. A passionate group of over 600,000 developers from more than 120 countries, all committed to helping PHP surpass the impossible.
Symfony Philosophy. Embracing and promoting professionalism, best practices, standardization and interoperability of applications.'

 - a) Определить, сколько раз встречается в тексте слово Symfony и вывести текст, выделив его в тексте цветом.
 - b) Вывести в браузер статистику файла - количество абзацев, предложений, слов, символов.
 - c) Найти самое длинное слово. Если таких несколько, вывести в браузер их все.
 - d) Для каждого символа, имеющегося в тексте подсчитать, сколько раз он там встречается, символы расположить в возрастающем порядке.

ЛИТЕРАТУРА

1. Брылёва, А. А. Программные средства создания интернет-приложений : учеб. Пособие / А. А. Брылёва. – Минск : РИПО, 2022. – 483 с, : ил.

*Лабораторная работа составлена
преподавателем спецдисциплин Еленой Иосифовной Шутько
март 2025 год*