|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |

Институт Информационных технологий

Кафедра Инструментального и прикладного программного обеспечения

**ПРАКТИЧЕСКАЯ РАБОТА №2**

по дисциплине «Разработка серверных частей интернет-ресурсов»

**Студент группы** ИКБО-02-20 Окороков Н.Д.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись студента)

**Руководитель практической работы** преподаватель Волков М.Ю.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Москва 2022

СОДЕРЖАНИЕ

[1. Цель работы 3](#_Toc114533758)

[2. Ход работы 3](#_Toc114533759)

[3. Ответы на вопросы к практической работе 6](#_Toc114533760)

[3.1. Конфигурационный файл php.ini 6](#_Toc114533761)

[3.2. Как написать простой скрипт на php 6](#_Toc114533762)

[3.3. Основные правила, связанные с переменными в php 6](#_Toc114533763)

[3.4. Основные типы данных в php 7](#_Toc114533764)

[3.5. Какие существуют функции для работы с переменными в php вне зависимости от типа данных 7](#_Toc114533765)

[3.6. Предопределенные переменные в php 8](#_Toc114533766)

[3.7. Переменные переменных в php 9](#_Toc114533767)

[3.8. Выражения в php 9](#_Toc114533768)

[3.9. Арифметические операторы в php 10](#_Toc114533769)

[3.10. Битовые операции в php 10](#_Toc114533770)

[3.11. Оператор присваивания в php 10](#_Toc114533771)

[3.12. Операторы сравнения в php 10](#_Toc114533772)

[3.13. Логические операторы в php 11](#_Toc114533773)

[3.14. Условная конструкция в php 11](#_Toc114533774)

[3.15. Циклы в php 12](#_Toc114533775)

[3.16. Конструкции switch и match в php 13](#_Toc114533776)

[3.17. Include и require в php 14](#_Toc114533777)

[3.18. Функции в php 14](#_Toc114533778)

[4. Ссылка на удаленный репозиторий проекта 15](#_Toc114533779)

[ЗАКЛЮЧЕНИЕ 16](#_Toc114533780)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 16](#_Toc114533781)

1. Цель работы

* Предлагается создать веб-сервис Drawer для рисования svg объектов. Ему передается один параметр - целое число, представляющее закодированная фигура для рисования.
* Реализовать одну из сортировок на языке программирования PHP по варианту. Массив передается скрипту как параметр строка: состоящий из значений элементов массива, разделенных запятыми.
* Реализовать информационно-административную веб-страницу о сервере с помощью таких команд Unix как: ls, ps, whoami, id и так далее.

1. Ход работы

Для создания фигуры используется числовой параметр num, заданный в URL строке. Это число преобразовывается в двоичный вид для определения свойств фигуры: тип фигуры (2 бита: круг, прямоугольник, квадрат, треугольник) – цвет (3 бита: RGB) – размер (2 бита: маленький, средний, большой, очень большой).

Рисунок 2.1 – Веб-сервис Drawer

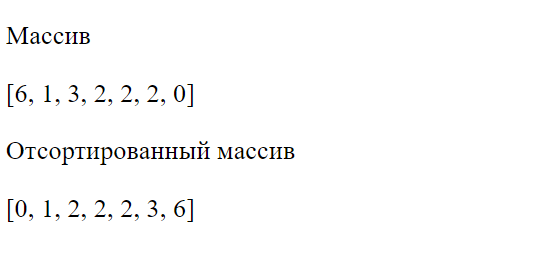
Сортировка массива была реализована с помощью «Быстрой сортировки».

Рисунок 2.2 – Веб-сервис Sort

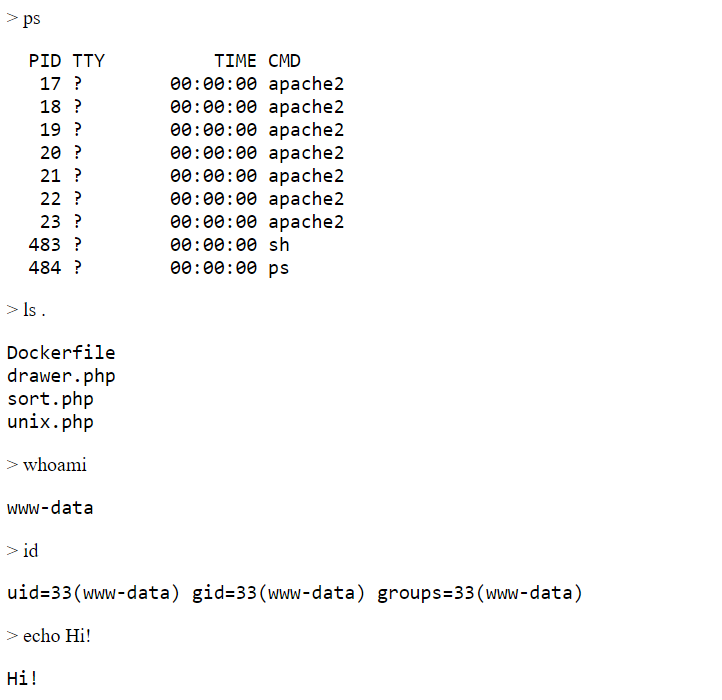
Для запуска Unix команд, все команды были помещены в массив для дальнейшего их вызова по порядку.

Рисунок 2.3 – Веб-сервис Unix

1. Ответы на вопросы к практической работе
   1. Конфигурационный файл php.ini

Файл конфигурации (php.ini) считывается при запуске PHP. Для версий серверных модулей PHP это происходит только один раз при запуске вебсервера. Для CGI и CLI версий это происходит при каждом вызове. Php.iniсодержит набор директив, каждая из которых пишется с новой строки и представляет собой пару ключ-значение, разделенные знаком равно. Строки начинающиеся с «;» не выполняются, это комментарии. Директивы определяют множество настроек интерпретатора PHP, а также отвечают за загрузку модулей.

* 1. Как написать простой скрипт на php

Код на PHP вставляется в HTML код страницы и выполняется один раз перед отправкой данных клиенту. В документ можно добавлять конструкции при помощи ключевого слова echo. Блок кода на php начинается со следующей конструкции: <?php. Заканчивается конструкцией ?>. Для примера, следующий скрипт, при запросе к серверу, вернет на клиент тег <h1> с текстом “Hello, world!”:

<?php

echo “Hello World”;

?>

* 1. Основные правила, связанные с переменными в php
* Переменная начинается с знака $, за которым следует имя переменной;
* Имя переменной должно начинаться с буквы или символа подчеркивания;
* Имя переменной не может начинаться с числа;
* Имя переменной может содержать только буквенно-цифровые символы и знаки подчеркивания;
* Имена переменных чувствительны к регистру
  1. Основные типы данных в php
* integer: целое число со знаком, размер зависит от разрядности системы;
* double: вещественное число, число с плавающей точкой, отводится 8 байт. Для обозначения бесконечности используется INF, а для обозначения несуществующего числа NAN;
* boolean: логическая переменная с двумя возможными состояниями true и false;
* string: набор символов;
* array: ассоциативный массив. В PHP нет обычного классического массива, как в других языках программирования, здесь массив представляет набор элементов, представляющих пары ключ => значение. Ключами могут быть как целые числа, так и строки;
* object: ссылка на объект, который реализует несколько принципов ООП;
* resource: некоторый ресурс, обрабатываемый языком особым образом;
* null: специальное значение;
* callable: нововведение версии 5.4. Этим типом является функция обратного вызова. Данный подход используется, когда в функцию нужно передать другую функцию, которая в этом случае и называется функцией обратного вызова;
* iterable: псевдотип, введённый в PHP 7.1. Он принимает любой массив или объект, реализующий интерфейс Traversable. Используется как тип параметра для указания, что функция принимает набор значений, но ей не важна форма этого набора, пока он будет использоваться с foreach.
  1. Какие существуют функции для работы с переменными в php вне зависимости от типа данных
* boolval — Возвращает логическое значение переменной
* debug\_zval\_dump — Выводит строковое представление внутренней структуры zval
* empty — Проверяет, пуста ли переменная
* floatval — Возвращает значение переменной в виде числа с плавающей точкой
* get\_debug\_type — Возвращает имя типа переменной в виде, подходящем для отладки
* get\_defined\_vars — Возвращает массив всех определённых переменных
* get\_resource\_id — Возвращает целочисленный идентификатор для данного ресурса
* get\_resource\_type — Возвращает тип ресурса
* gettype — Возвращает тип переменной
* intval — Возвращает целое значение переменной
* is\_типданных — Определяет, является ли переменная типданных (is\_array, is\_bool, is\_int…)
* isset — Определяет, была ли установлена переменная значением, отличным от null
* print\_r — Выводит удобочитаемую информацию о переменной
* serialize — Генерирует пригодное для хранения представление переменной
* settype — Задаёт тип переменной
* strval — Возвращает строковое значение переменной
* unserialize — Создаёт PHP-значение из хранимого представления
* unset — Удаляет переменную
* var\_dump — Выводит информацию о переменной
* var\_export — Выводит или возвращает интерпретируемое строковое представление переменной
  1. Предопределенные переменные в php
* $GLOBALS — Ссылки на все переменные глобальной области видимости
* $\_SERVER — Информация о сервере и среде исполнения
* $\_GET — Переменные HTTP GET
* $\_POST — Переменные HTTP POST
* $\_FILES — Переменные файлов, загруженных по HTTP
* $\_REQUEST — Переменные HTTP-запроса
* $\_SESSION — Переменные сессии
* $\_ENV — Переменные окружения
* $\_COOKIE — HTTP Cookies
* $php\_errormsg — Предыдущее сообщение об ошибке
* $http\_response\_header — Заголовки ответов HTTP
* $argc — Количество аргументов, переданных скрипту
* $argv — Массив переданных скрипту аргументов
  1. Переменные переменных в php

Любопытным инструментом разработки PHP являются переменные переменных. То есть имя переменной, которое может быть определено и изменено динамически.

<?php

// определим обычную переменную

$a = 'hello';

// переменная переменной берет значение переменной и рассматривает его как имя переменной.

$$a = 'world'; // значение переменной $hello теперь такое

echo "$a ${$a}"; // hello world

echo "$a $hello"; // hello world

?>

* 1. Выражения в php

Выражения - это самые важные строительные элементы PHP. Почти всё, что вы пишете в PHP, является выражением. Самое простое и точное определение выражения - "все что угодно, имеющее значение". Самым простым примером выражения является переменная или константа в операторе присваивания.

Очень распространённым типом выражений являются выражения сравнения. Результатом вычислений являются false (ложь) или true (истина).

* 1. Арифметические операторы в php

Унарные операторы: идентичность +, отрицание -, префиксный инкремент (++$a), постфиксный инкремент ($a++), префиксный декремент (--$a), постфиксный декремент ($a--).

Бинарные операторы: +, -, \*, /, %, \*\* (возведение в степень).

* 1. Битовые операции в php

Унарные операции: ~ (отрицание);

Бинарные операции: &, |, ^, <<, >>.

(И, ИЛИ, ИСКЛ ИЛИ, Сдвиги)

* 1. Оператор присваивания в php

Оператор присваивания означает, что левый операнд получает значение правого выражения, то есть устанавливается значением.

<?php

$a = ($b = 4) + 5; // $a теперь равно 9, а $b было присвоено

4.

$a = 3;

$a += 5; // устанавливает $a в 8, как если бы мы написали: $a

= $a + 5;

$b = "Привет";

$b .= "-привет!"; // устанавливает $b в "Привет-привет!", как

и $b = $b . "-привет!";

?>

* 1. Операторы сравнения в php

$a == $b (Равно): true если $a равно $b после преобразования типов.

$a === $b (Тождественно равно): true если $a равно $b и имеет тот же тип.

$a != $b (Не равно): true если $a не равно $b после преобразования типов.

$a <> $b (Не равно): true если $a не равно $b после преобразования типов.

$a !== $b (Тождественно не равно): true если $a не равно $b, или они разных типов.

$a < $b (Меньше): true если $a строго меньше $b.

$a > $b (Больше): true если $a строго больше $b.

$a <= $b (Меньше или равно): true если $a меньше или равно $b.

$a >= $b (Больше или равно): true если $a больше или равно $b.

$a <=> $b (Космический корабль (spaceship)): число типа int (-1 1 0), когда $a соответственно меньше, больше или равно $b.

* 1. Логические операторы в php

$a and $b (И): true, если и $a, и $b true.

$a or $b (Или):true, если или $a, или $b true.

$a xor $b (Исключающее или): true, если $a, или $b true, но не оба.

! $a (Отрицание): true, если $a не true.

$a && $b (И): true, если и $a, и $b true.

$a || $b (Или): true, если или $a, или $b true.

* 1. Условная конструкция в php

Конструкция if является одной из наиболее важных во многих языках программирования, в том числе и PHP. Она предоставляет возможность условного выполнения фрагментов кода. Структура if реализована в PHP по аналогии с языком C: if (выражение) инструкция

Часто необходимо выполнить одно выражение, если определённое условие верно, и другое выражение, если условие не верно. Именно для этого else и используется. else расширяет оператор if, чтобы выполнить выражение, в случае, если условие в операторе if равно false.

Конструкция elseif, как её имя и говорит есть сочетание if и else. Аналогично else, она расширяет оператор if для выполнения различных выражений в случае, когда условие начального оператора if эквивалентно false. Однако, в отличие от else, выполнение альтернативного выражения произойдёт только тогда, когда условие оператора elseif будет являться равным true.

* 1. Циклы в php

While

Циклы while являются простейшим видом циклов в PHP. Они ведут себя так же, как и в языке C.

Do-While

Цикл do-while очень похож на цикл while, с тем отличием, что истинность выражения проверяется в конце итерации, а не в начале.

For

Вспоминая уже не в первый раз такой язык программирования как Си, то данный тип цикла чаще всего используется для перебора значений счетчика и каких-либо действий с использованием данного счетчика.

Foreach

В PHP существует и особый тип цикла для перебора элементов массива и итерируемых объектов.

<?php

// вывод всех переменных окружения

foreach($\_SERVER as $key => $value){

echo “<b>$k</b> => <tt>$value</tt><br />\n”;

}

// вывод всех элементов списка

// под списком понимается массив, где ключами являются последовательные

// числовые индексы

$array = array("foo", "bar", "hallo", "world");

foreach($array as $value){

echo “$value<br />\n”;

}

?>

* 1. Конструкции switch и match в php

Оператор switch похож на ряд операторов IF с одинаковым условием. Во многих случаях вам может понадобиться сравнивать одну и ту же переменную (или выражение) с множеством различных значений и выполнять различные участки кода в зависимости от того, какое значение принимает эта переменная (или выражение). Это именно тот случай, для которого удобен оператор switch.

switch ($i) {

case 0:

echo "i равно 0";

break;

case 1:

echo "i равно 1";

break;

case 2:

echo "i равно 2";

break;

default:

echo “i не равно 0, 1 и 2”;

}

Выражение match предназначено для ветвления потока исполнения на основании проверки совпадения значения с заданным условием. Аналогично оператору switch, выражение match принимает на вход выражение, которое сравнивается с множеством альтернатив. Но, в отличие от switch, оно обрабатывает значение в стиле, больше похожем на тернарный оператор. Также, в отличие от switch, используется строгое сравнение (===), а не слабое (==).

$return\_value = match ($food) {

'apple' => 'На столе лежит яблоко',

'banana' => 'На столе лежит банан',

'cake' => 'На столе стоит торт',

};

* 1. Include и require в php

Выражение include включает и выполняет указанный файл.

Файлы включаются исходя из пути указанного файла, или, если путь не указан, используется путь, указанный в директиве include\_path. Если файл не найден в include\_path, include попытается проверить директорию, в которой находится текущий включающий скрипт и текущую рабочую директорию перед тем, как выдать ошибку. Конструкция include выдаст E\_WARNING, если не сможет найти файл; поведение отлично от require, который выдаст E\_ERROR.

require аналогично include, за исключением того, что в случае возникновения ошибки он также выдаст фатальную ошибку уровня E\_COMPILE\_ERROR. Другими словами, он остановит выполнение скрипта, тогда как include только выдал бы предупреждение E\_WARNING, которое позволило бы скрипту продолжить выполнение.

* 1. Функции в php
* У функции может быть переменное количество параметров и параметры по умолчанию;
* Функция имеет собственную область видимости. Это нужно учитывать при работе с глобальными переменными программы и с локальными переменными функции, которые уничтожаются при окончании ее работы;
* В php тип возвращаемого значения может быть любым и тип возвращаемого значения не регламентируется. То есть функция может возвращать и число, и строку одновременно;
* Допускается создание и использование при необходимости анонимных функций;

$greet = function($name)

{

printf("Привет, %s\r\n", $name);

};

* В пределах одного сценария не должно быть определений двух функций с одинаковыми именами.

Приведём пример синтаксиса, используемого для описания функций:

<?php

function foo($arg\_1, $arg\_2, /\* ..., \*/ $arg\_n)

{

echo "Пример функции.\n";

return $retval;

}

?>

Внутри функции можно использовать любой корректный PHP-код, в том числе другие функции и даже объявления классов.

Имена функций следуют тем же правилам, что и другие метки в PHP. Корректное имя функции начинается с буквы или знака подчёркивания, за которым следует любое количество букв, цифр или знаков подчёркивания.

1. Ссылка на удаленный репозиторий проекта

Полный код проекта можно найти по ссылке:

<https://github.com/Chidorick/DotSPIR>

ЗАКЛЮЧЕНИЕ

Была созданы сервисы Drawer, Sort, Unix и проверена их работоспособность. Данная практическая работа позволила изучить на практике работу с языком PHP.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. «Учебное пособие РСЧИР 2021-2022»: [Электронный ресурс]. URL: <https://online-edu.mirea.ru/mod/resource/view.php?id=406048> (дата обращения 20.09.2022)
2. «Руководство по PHP»: [Электронный ресурс]. URL: <https://www.php.net/manual/ru/index.php> (дата обращения 20.09.2022)
3. «Документация Docker»: [Электронный ресурс]. URL: <https://docs.docker.com/> (дата обращения 20.09.2022)