|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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-01-19 Кузин Данил Олегович

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

(подпись студента)

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

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

(подпись руководителя)

Работа представлена «\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Допущен к работе «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_ 2021 г.

Москва 2021

**Оглавление**

[Ход работы 3](#_Toc84066915)

[Вывод 10](#_Toc84066916)

[Ответы на вопросы к практической работе 10](#_Toc84066917)

[Список использованной литературы 13](#_Toc84066918)

# Ход работы

1. Был создан общий проект с разделенными сервисами, разделением проекта на файлы для разделения функционала. Его структура представлена на рисунке 1.

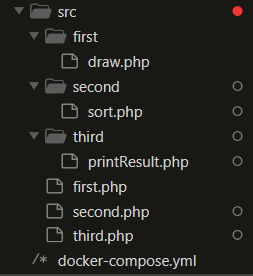


Рисунок - Файловая система проекта

Содержимое файлов представлено на листингах 1-6.

Листинг 1 – Файл first.php

<html lang="ru">

<head>

<title>drawer</title>

</head>

<body>

<?php

$code = htmlspecialchars($\_GET["num"]);

$type\_figure = $code >> 40;

$color = ($code >> 16) & 0xffffff;

$width = ($code >> 8 & 0xff) \* 2.5;

$height = ($code & 0xff) \* 2.5;

require \_\_DIR\_\_ . '/first/draw.php';

draw($width, $height, $type\_figure, $color);

echo '<br><br>type\_figure ' . $type\_figure;

echo ' color ' . $color;

echo ' width ' . $width;

echo ' height ' . $height;

?>

</body>

</html>

Листинг 2 – Файл draw.php

<?php

function draw($width, $height, $type\_figure, $color)

{

$result = '<svg width="' . $width . '" height="' . $height . '" version="1.1" xmlns="http://www.w3.org/2000/svg">';

$hex\_color = calcColor($color);

$is\_correct\_figure = true;

switch ($type\_figure) {

case 3:

$result = $result . ' <polygon points="' . 0 . ',' . $height . ' ' . $width / 2 . ',' . 0 . ' ' . $width . ',' . $height . '" fill="#' . $hex\_color . '"/>';

break;

case 2:

$result = $result . '<ellipse cx="50%" cy="50%" rx="50%" ry="50%" fill="' . "#" . $hex\_color . '"/>';

break;

case 1:

$result = $result . '<rect x="0" y="0" width="100%" height="100%" fill="' . "#" . $hex\_color . '"/>';

break;

case 0:

$result = $result . '<circle cx="50%" cy="50%" r="' . min($height, $width) / 2 . '" fill="#' . $hex\_color . '"/>';

break;

default:

$is\_correct\_figure = false;

break;

}

$result = $result . '</svg>';

if ($is\_correct\_figure) {

echo $result;

} else {

echo 'Фигуры с номером: ' . $type\_figure . ' не существует';

}

}

function calcColor($color)

{

$hex = dechex($color);

return str\_repeat("0", 6 - strlen($hex)) . $hex;

}

Листинг 3 – Файл second.php

<html lang="ru">

<head>

<title>Shell sort</title>

</head>

<body>

<?php

$arrayString = htmlspecialchars($\_GET["arr"]);

$array = array\_map('intval', explode(',', $arrayString));

require \_\_DIR\_\_ . '/second/sort.php';

print\_r(shellSort($array));

?>

</body>

</html>

Листинг 4 – Файл sort.php

<?php

function shellSort($array)

{

$n = count($array);

for ($step = $n / 2; $step > 0; $step /= 2){

for ($i = $step; $i < $n; $i++) {

for ($j = $i - $step; $j >= 0 && $array[$j] > $array[$j + $step]; $j -= $step) {

$t = $array[$j];

$array[$j] = $array[$j+$step];

$array[$j+$step] = $t;

}

}

}

return $array;

}

Листинг 5 – Файл third.php

<html lang="ru">

<head>

<title>Command</title>

</head>

<body>

<?php

$output = null;

$command = htmlspecialchars($\_GET["command"]);

exec($command, $output);

require \_\_DIR\_\_ . '/third/printResult.php';

printResult($output, $command);

?>

</body>

</html>

Листинг 6 – Файл printResult.php

<?php

function printResult($output, $command) {

echo "Команда " . $command . "<br>";

foreach ($output as $line\_output) {

echo $line\_output . "<br>";

}

}

?>

1. Веб-сервис для рисования svg объектов. Ему передается параметр num с целым числом, первые два бита которого отвечают за тип фигуры, двадцать четыре – за цвет, три – за ширину и три – за высоту. Результаты работы представлены на рисунках 2-5.



Рисунок – Отображение круга

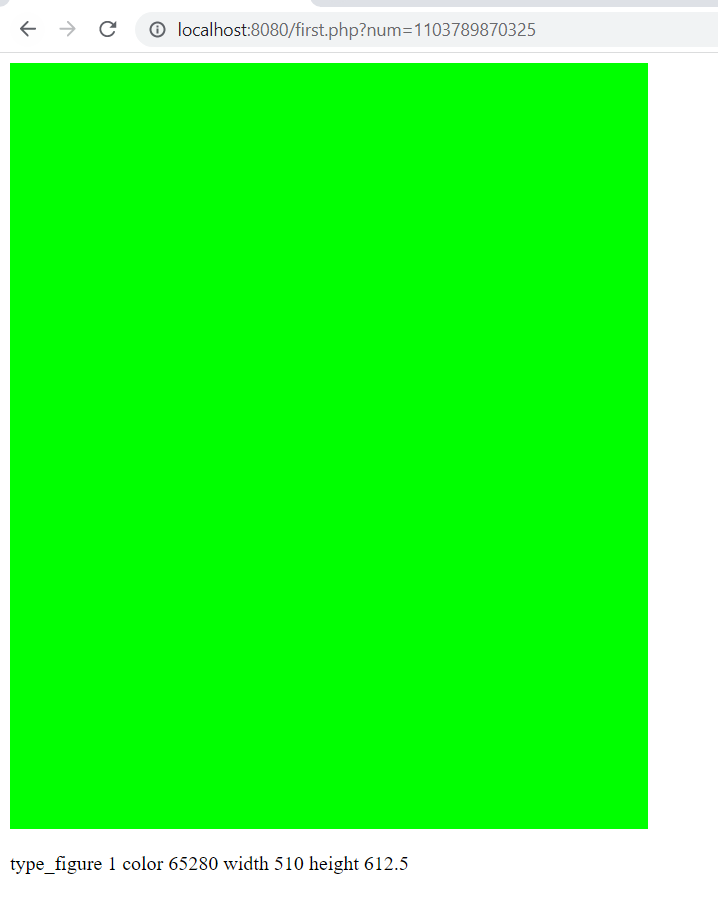


Рисунок – Отображение прямоугольника

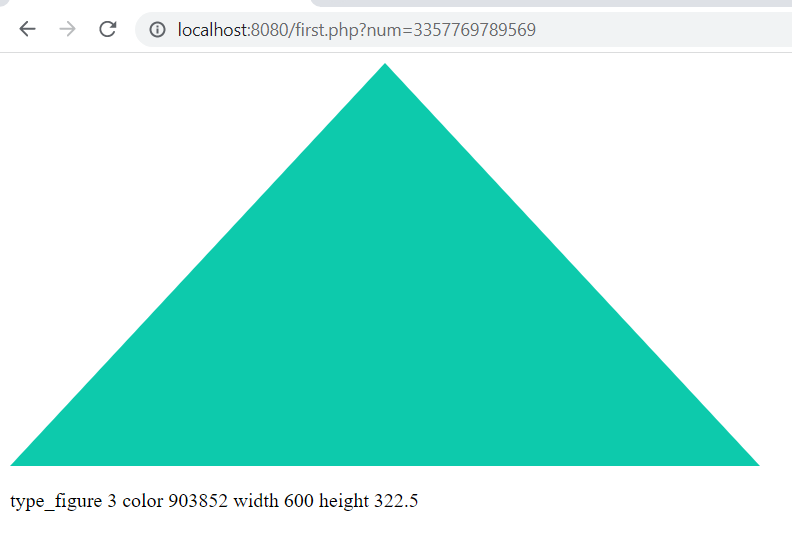


Рисунок – Отображение треугольника

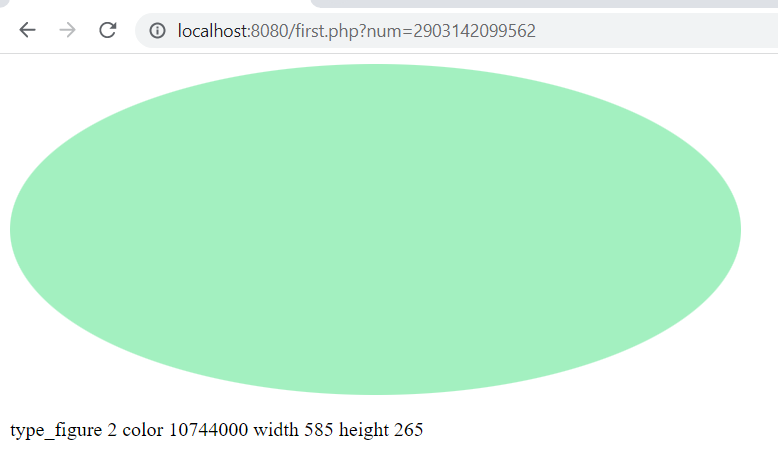


Рисунок - Отображение эллипса

1. Веб-сервис для сортировки массива из целых чисел, принимающий в качестве параметра array – строку, где числа записаны через запятую. Результаты работы представлены на рисунке 6.

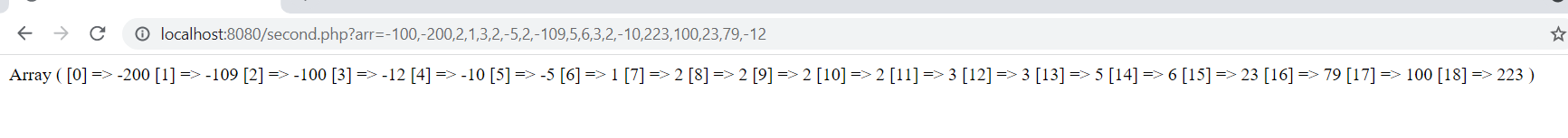


Рисунок – Результат сортировки массива

1. Веб-сервис для генерации информационно-административной веб-страницы, принимающий как параметр команду Unix. Результаты работы представлены на рисунках 7-8.

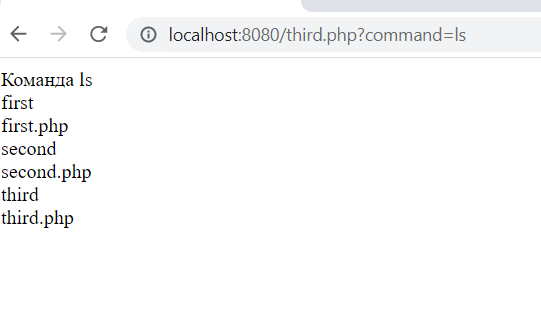


Рисунок - Вывод команды ls

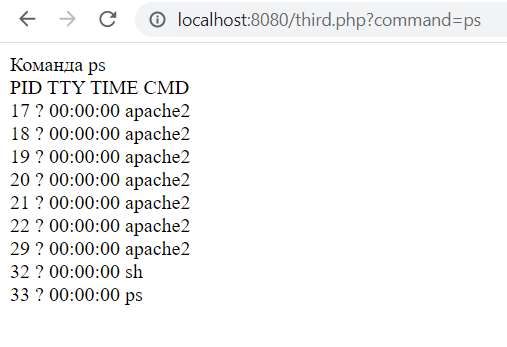


Рисунок - Вывод команды ps

# Вывод

В работе была продолжена настройка серверной конфигурации, разработанная в прошлой практической работе. В результате было изучено использование разделения проекта нa сервисы и файлы. Также были использованы на практике различные возможности языка php для обработки входящих параметров.

# Ответы на вопросы к практической работе

Ini **-** это конфигурационный файл с подавляющим большинством настроек PHP. По умолчанию файл конфигурации входит в дистрибутив PHP и называется php. ini-development. Чтобы интерпретатор PHP находил этот файл его нужно переименовать в php.

Правила для переменных 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 (a-z, 0-9 и \_)
* Имена переменных чувствительны к регистру

В PHP есть десять базовых типов данных:bool**,** int**,** float**,** string**,** array**, o**bject**,** callable**,** mixed

Анонимные функции, также известные как замыкания (closures), позволяют создавать функции, не имеющие определённых имён.

PHP предоставляет всем скриптам большое количество предопределённых переменных. Эти переменные содержат всё, от внешних данных до переменных среды окружения, от текста сообщений об ошибках до последних полученных заголовков.

Переменная переменной берет значение переменной и рассматривает его как имя переменной. Выражения в php.

Арифметические операторы в php**:** +$a **–** Идентичность**;** -$a **–** Отрицание**;** $a + $b **–** Сложение**;** $a - $b **–** Вычитание**;** $a \* $b **–** Умножение**;** $a / $b **–** Деление**;** $a % $b **-** Деление по модулю**;** $a \*\* $b **–** Возведениев степень**;**

Побитовые операторы:$a & $b **-** Побитовое 'и'**;** $a | $b **-** Побитовое 'или**’;** $a ^ $b **-** Исключающее или**;** ~ $a **–** Отрицание**;** $a << $b **-** Сдвиг влево**;** $a >> $b **-** Сдвиг вправо**.**

Базовый оператор присваивания обозначается как "=". Операторы сравнения в php.Результатом выполнения оператора присваивания является само присвоенное значение.

Логические операторы в php**:** $a and $b **–** И**;** $a or $b **–** Или**;** $a xor $b **-** Исключающее или**;** ! $a **–** Отрицание**;** $a && $b **–** И**;** $a || $b **–** Или

Условные конструкции позволяют направлять работу программы в зависимости от условия по одному из возможных путей. И одной из таких конструкций в языке PHP является конструкция if..else

Тернарная операция состоит из трех операндов и имеет следующее определение: [первый операнд - условие] ? [второй операнд] : [третий операнд].

Циклы позволяют повторять определенное (и даже неопределенное - когда работа цикла зависит от условия) колличество раз различные операторы. Данные операторы называются телом цикла. Проход цикла называется итерацией.

PHP поддерживает три вида циклов:Цикл с предусловием (while);Цикл с постусловием (do-while);Цикл со счетчиком (for);Специальный цикл перебора массивов (foreach).

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).

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 (==).

Оператор include выдает предупреждение и продолжает исполнение кода, если запрашиваемый файл не был найден. Оператор require вызывает фатальную ошибку и останавливает выполнение скрипта.

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. В качестве регулярного выражения оно может быть выражено так: ^[a-zA-Z\_\x80-\xff][a-zA-Z0-9\_\x80-\xff]\*$.

# Список использованной литературы

1. Документация Docker — Текст : электронный // Docker docs : [Электронный ресурс]. — URL: https://docs.docker.com/ (дата обращения: 04.09.2021).
2. Руководство по php — Текст : электронный // php : [Электронный ресурс]. — URL: https://www.php.net/ (дата обращения: 22.09.2021).
3. Docker Hub — Текст : электронный // Docker Hub : [Электронный ресурс]. — URL: https://hub.docker.com/ (дата обращения: 04.09.2021).
4. Справочник по php — Текст : электронный // Docker Hub : [Электронный ресурс]. — URL: https://php.ru/manual/ (дата обращения: 22.09.2021).