|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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»

**Тема практической работы:**

**«Основы PHP»**

**Студент группы** ИКБО-30-20 Лазонов А.Е.

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

(подпись студента)

**Руководитель практической работы**  Благирев М.М.

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

(подпись руководителя)

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

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

Москва 2022

# **Цель работы**

Используя серверную конфигурацию, разработанную в прошлой практической работе выполнить упражнения.

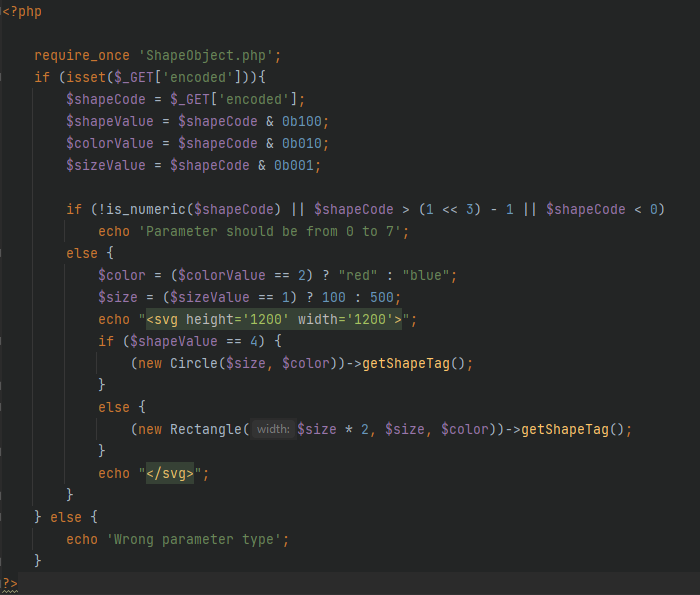
# **Постановка задачи**

Используя серверную конфигурацию, разработанную в прошлой практической работе выполнить следующие упражнения. Предполагается создать 3 независимых сервиса, устойчивых к минимальному набору самых простых ошибок. Предполагается создание 1 общего проекта с разделенными сервисами, разделением проекта на файлы для разделения функционала и переиспользования файлов. Каждый сервис должен состоять как минимум из 2 файлов.

# **Ход работы**

**Упражнение 1**

Предлагается создать веб-сервис Drawer для рисования svg объектов. Ему передается один параметр - целое число, представляющее закодированная фигура для рисования. С помощью побитых операций вытаскиваем следующие данные: форма, цвет, размер.

Рисунок 1 – Фрагмент кода к упражнению 1

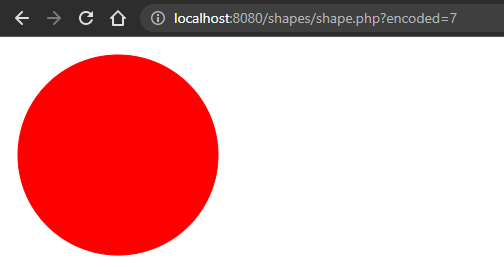
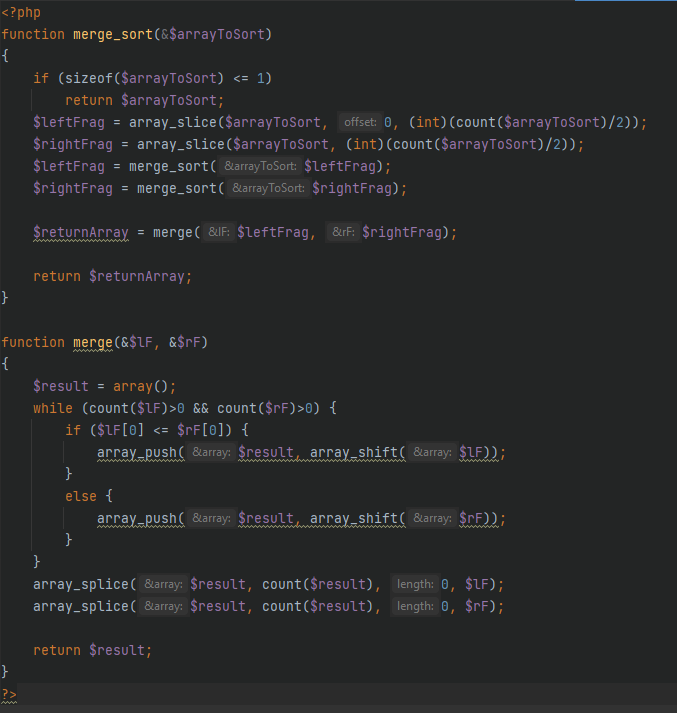
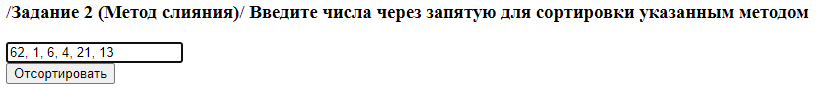
Рисунок 2 — Отображение круга

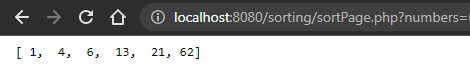
Рисунок 3 — Отображение прямоугольника

**Упражнение 2**

Для выполнения упражнения необходим переданный параметр array, который должен являться строкой и быть отличным от null. Преобразуем данную строку в массив целых чисел. Далее выполним сортировку Шелла (вариант 5). При сортировке Шелла сначала сравниваются и сортируются между собой значения, отстоящие один от другого на некотором расстоянии d. После этого процедура повторяется для некоторых меньших значений d, а завершается сортировка Шелла упорядочиванием элементов при d=1.

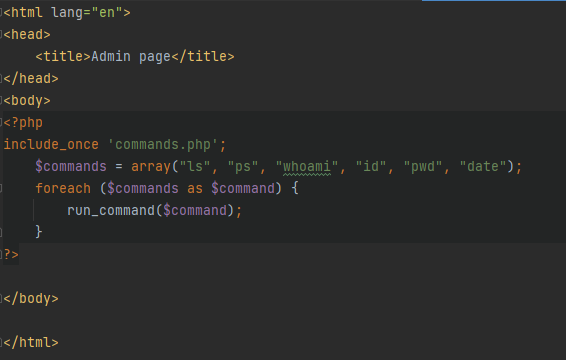
Рисунок 4 – Фрагмент кода с сортировкой массива

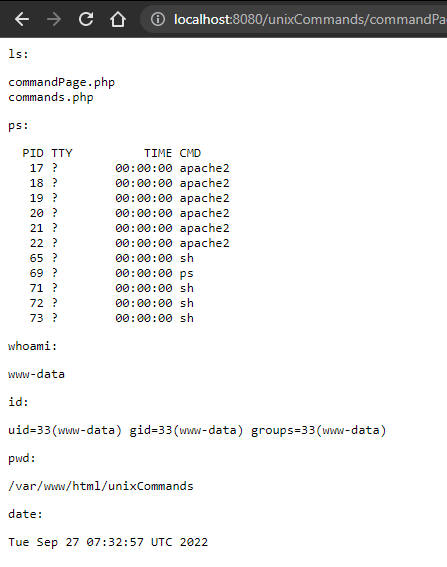
Рисунок 5 — Входной массив чисел

Рисунок 6 — Результат сортировки массива методом Шелла

**Упражнение 3**

Реализовать информационно-административную веб-страницу о сервере с помощью таких команд Unix как: ls, ps, whoami, id и так далее.

 Рисунок 7 – Фрагмент кода оформления сведений о сервере

Рисунок 8 — Информационно-административная веб-страница

# **Выводы**

В результате выполнения практической работы были изучены основы языка PHP, получены знания по «рисованию фигур», сортировке данных в пределах веб-сервиса, а также выведены информационно-административные сведения.

**Ссылка на удаленный репозиторий проекта**

<https://github.com/ISilentI/Second_Practice>

# **Ответы на вопросы к практической работе**

1. **Конфигурационный файл php.ini.**

В конфигурационном файле сосредоточены настройки интерпретатора PHP и его многочисленных расширений. Содержимое данного файла представляет собой секции и директивы. Секции заключаются в квадратные скобки, после них идут директивы формата “directive = value”. Комментариями являются строки, начинающиеся с точки запятой.

1. **Как написать простой скрипт на php.**

Весь код обернуть в <?php>.

1. **Основные правила, связанные с переменными в php**

В PHP как и во многих других языках существует такое понятие, как переменная. Правилом хорошего тона данного языка является не скупиться на количество переменных в коде, т.к. интерпретатору создать новый идентификатор не затратно. Основные правила, связанные с переменными в языке PHP: 1. Объявление переменной начинается со знака $. Данная особенность языка облегчает интерпретатору выделение переменных в тексте. 2. Имена переменных должны состоять из латинских букв, цифр и нижнего подчеркивания. Хотя данное правило не является обязательным, рекомендуется его соблюдать.

1. **Основные типы данных в php.**

1) integer: целое число со знаком, размер зависит от разрядности системы; 2) double: вещественное число, число с плавающей точкой, отводится 8 байт. Для обозначения бесконечности используется INF, а для обозначения несуществующего числа NAN; 3) boolean: логическая переменная с двумя возможными состояниями true и false; 4) string: набор символов; 5) 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; 6) object: ссылка на объект, который реализует несколько принципов ООП; 7) resource: некоторый ресурс, обрабатываемый языком особым образом; 28 8) null: специальное значение; 9) 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; 10) iterable: принимает любой массив или объект. 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.**

Gettype, is\_type, isset, settype, unset.

1. **Предопределенные переменные в php**

PHP предоставляет всем скриптам большое количество предопределённых переменных. Эти переменные содержат всё, от [внешних данных](https://www.php.net/manual/ru/language.variables.external.php) до переменных среды окружения, от текста сообщений об ошибках до последних полученных заголовков.

1. **Арифметические операторы в php.**

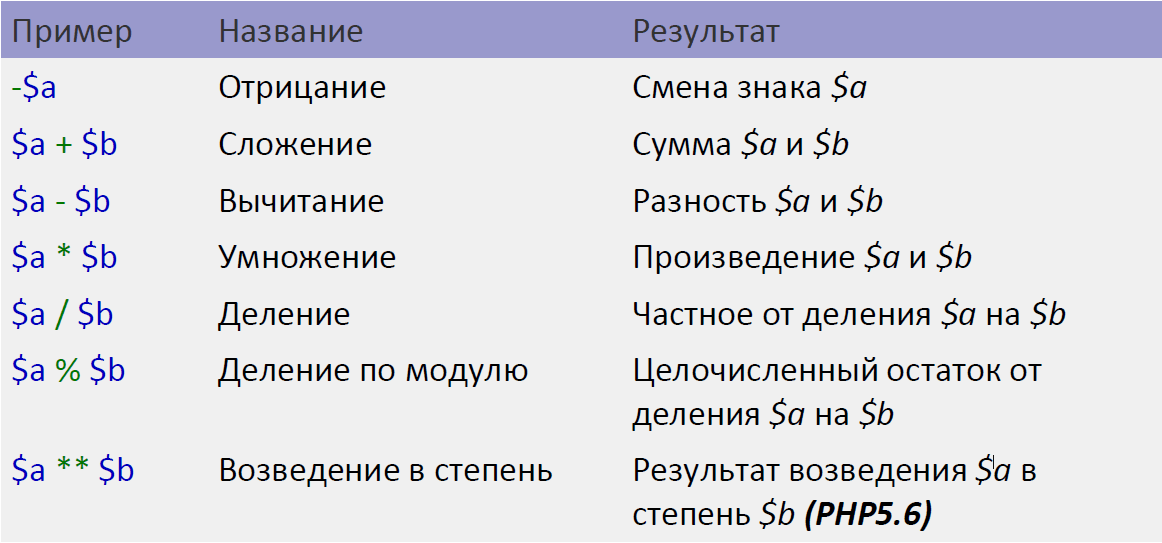


Рисунок 9 — Арифметические операторы

1. **Битовые операции в php**

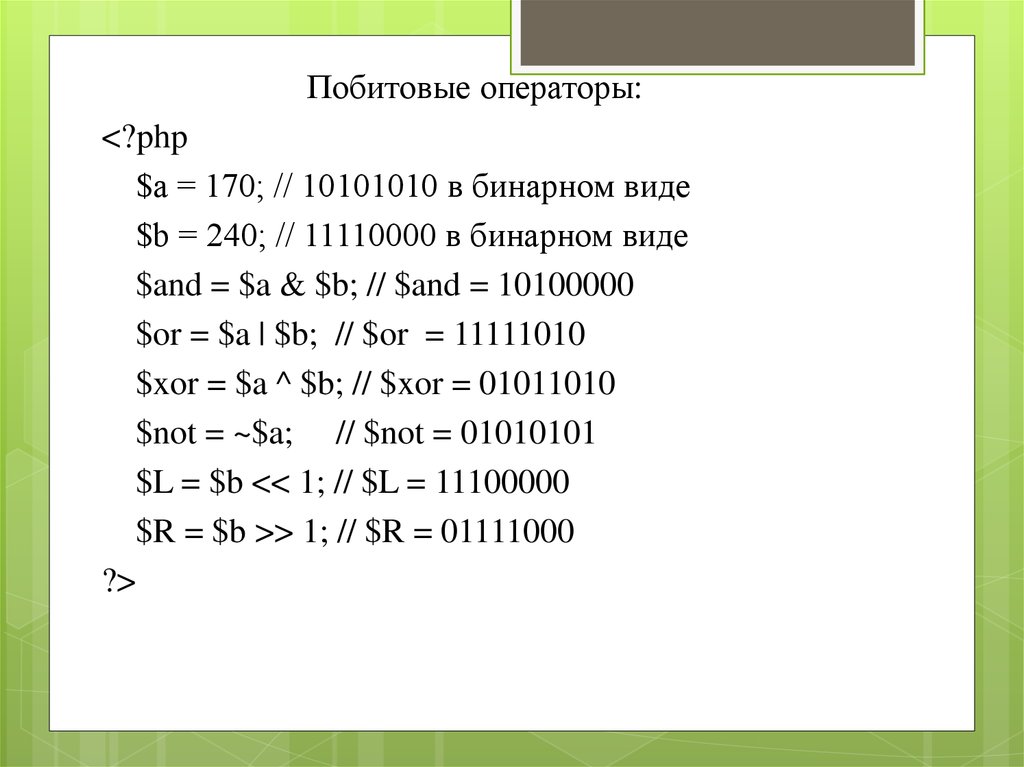


Рисунок 10 — Битовые операции

1. **Операторы присваивания.**



Рисунок 11 — Операторы сравнения

1. **Операторы сравнения**



Рисунок 12 — Операторы сравнения

1. **Логические операторы**

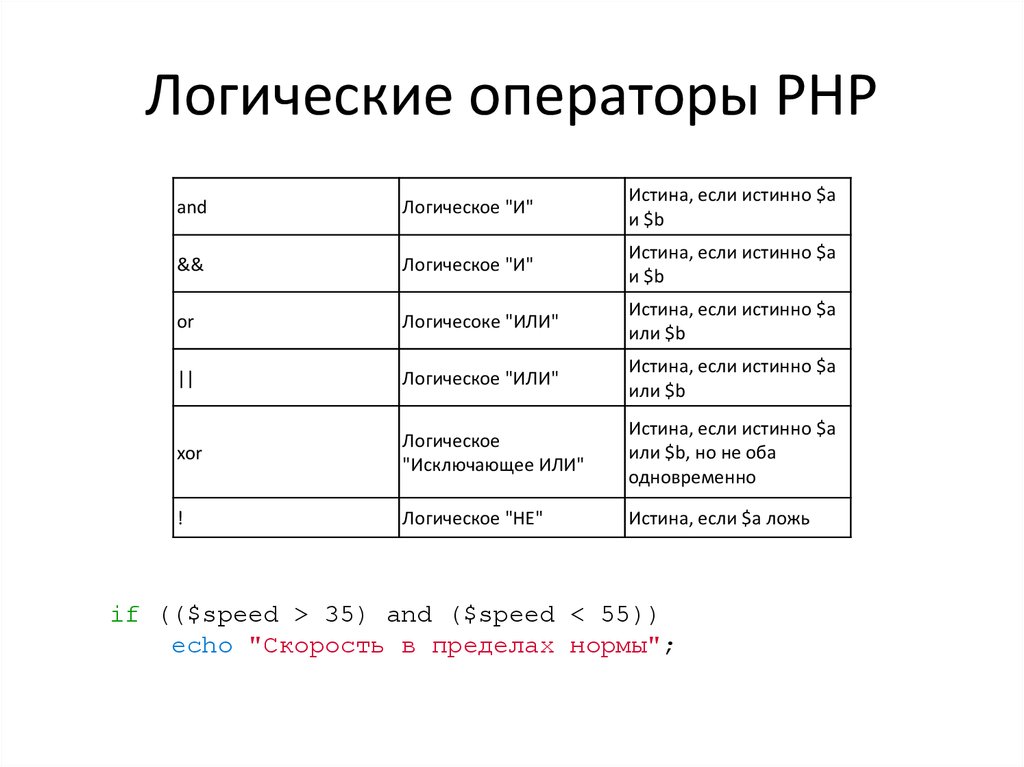


Рисунок 13 — Логические операторы

1. **Условная конструкция в php.**

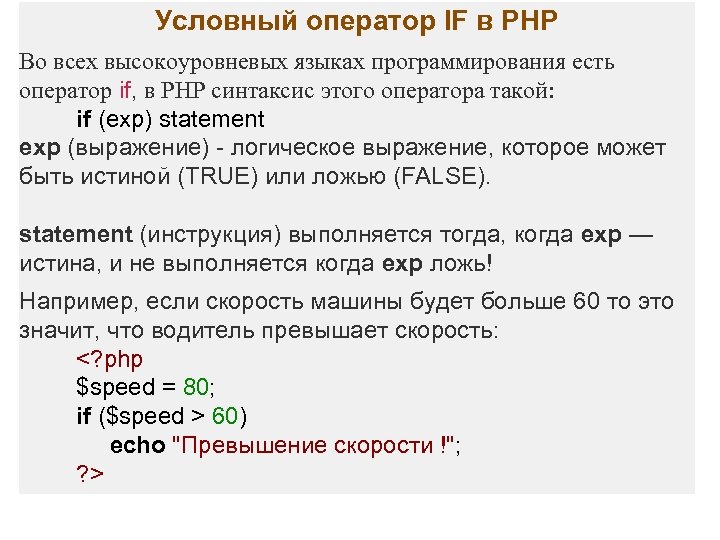


Рисунок 14 — Условный оператор

1. **Циклы в php.**

PHP поддерживает 4 вида циклов:

- Цикл с предусловием (while)

- Цикл с постусловием (do-while)

- Цикл со счетчиком (for)

- Специальный цикл перебора массивов (foreach)

1. **Конструкции switch и match в php.**

Конструкция switch..case является альтернативой использованию конструкции if..elseif..else. Оператор switch получает некоторое выражение и сравнивает его с набором значений. После ключевого слова switch в скобках идет сравниваемое выражение. Значение этого выражения последовательно сравнивается со значениями, помещенными после операторов сase. И если совпадение будет найдено, то будет выполняться определенный блок сase.

1. **Include и require в php.**

Конструкция include предназначена для включения файлов в код сценария PHP во время исполнения сценария PHP. В отличие от конструкции require конструкция include позволяет включать файлы в код PHP скрипта во время выполнения сценария.

Конструкция require позволяет включать файлы в PHP сценарий до выполнения сценария PHP. Общий синтаксис require такой: require имя\_файла; При запуске (именно при запуске, а не при исполнении!) программы интерпретатор просто заменит инструкцию на содержимое файла имя\_файла (этот файл может также содержать сценарий на PHP, обрамленный, как обычно, тэгами< ? И ?>).

1. **Функции в php**

Функция — очень мощный инструмент повторного использования кода. Создав свою функцию и записав туда необходимый код, вы сможете вызывать и использовать его столько раз, сколько необходимо. В противном случае пришлось бы копировать и вставлять фрагмент кода каждый раз, когда он понадобится. Чтобы упростить себе работу, мы можем оформить в виде функции некоторую часть кода, которая используется в сценарии несколько раз. Затем, вместо копирования и вставки этой части кода, будет достаточно только вызывать эту функцию, как если бы мы обращались к переменной. Разделяют два типа функций — встроенные**и**пользовательские**.** Встроенные**функции**— это функции, которые за нас уже написали создатели языка программирования, и мы можем просто брать их и использовать. В PHP существуют тысячи готовых функций на все случаи жизни. Пользовательские функции программист создает самостоятельно. Эти функции, как правило, используются только внутри одного проекта или, даже, сценария.

**17. Переменные переменных в php**

**Иногда бывает удобно иметь переменными имена переменных. То есть, имя переменной, которое может быть определено и изменено динамически.** Переменная переменной берет значение переменной и рассматривает его как имя переменной. Оно может быть использовано как имя переменной при помощи двух знаков доллара.

**18. 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". Основными формами выражений являются константы и переменные. Однако PHP, как и многие другие языки, понимает гораздо больше выражений. PHP - это язык, ориентированный на выражения и рассматривающий почти все как выражение. Очень распространённым типом выражений являются выражения сравнения.

**Список использованных источников и литературы:**

1. Временные указания к практической работе №2 [Файл для чтения]. Режим доступа: Сайт ЦДО МИРЭА, свободный (дата обращения: всё время работы);
2. Документация по языку PHP [Электронный ресурс]. Режим доступа: https://www.php.net/, свободный (дата обращения: всё время работы);
3. Команды Unix [Электронный ресурс]. Режим доступа: https://www.opennet.ru/docs/RUS/intro\_sun\_adm/unixbeg.html, свободный (дата обращения: всё время работы).