|  |  |  |
| --- | --- | --- |
|  |  |  |
| 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:**

**Студент группы** ИКБО-16-20 Голубев Григорий Александрович

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

(подпись студента)

**Руководитель практической работы** преподаватель

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

(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:**

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.

# **Ход работы:**

**Упражнение 1.**

Для выполнения упражнения необходим переданный параметр num, который должен являться числом и быть отличным от null. С помощью побитых операций вытаскиваем следующие данные: цвет, форму, координаты x и y. Далее с помощью тега <svg> отрисовываем фигуру. Кусок кода представлен на рисунке 1.

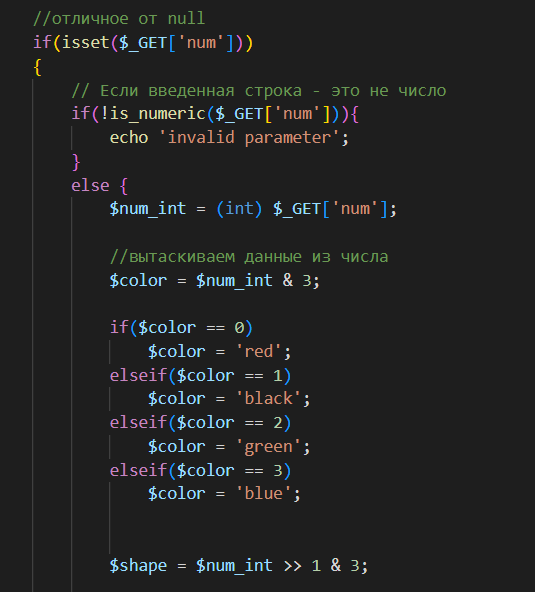


Рисунок 1 – кусок кода для упражнения 1.

Пример при num=10 представлен на рисунке 2.

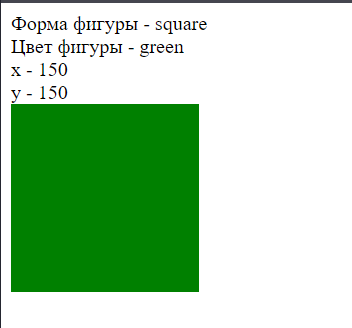


Рисунок 2 – пример при num=10.

Пример при num=15 представлен на рисунке 3.

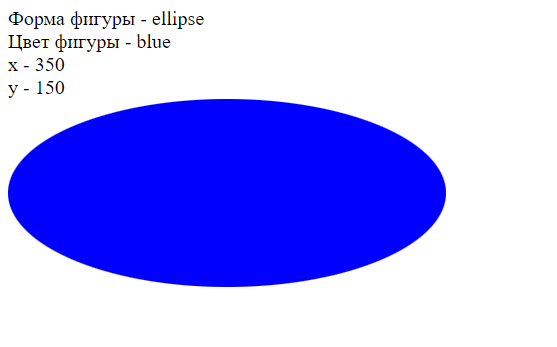


Рисунок 3 – пример при num=15.

**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 (то есть обычной сортировкой вставками). Кусок кода представлен на рисунке 4.

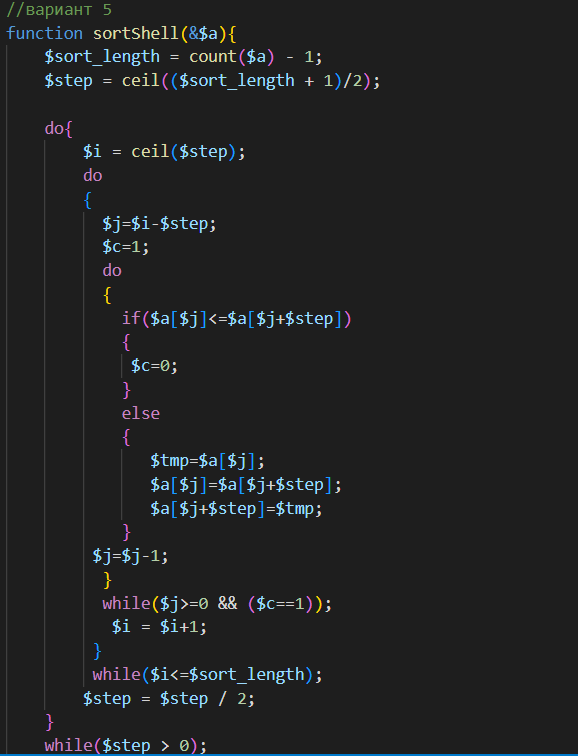


Рисунок 4 – кусок кода для упражнения 2.

Пример при array=78,23,56,1,8,32,1 представлен на рисунке 5.

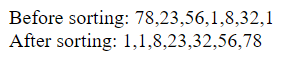


Рисунок 5 – пример при array=78,23,56,1,8,32,1.

**Упражнение 3.**

Для выполнения упражнения необходим переданный параметр cmd, который должен быть отличным от null. Обращаемся к system, которая выполнит переданные команды в Linux. Код представлен на рисунке 6.

 Рисунок 6 – код для упражнения 3.

Пример при cmd=ls представлен на рисунке 7.



Рисунок 7 – пример при cmd=ls.

# **Выводы:**

В ходе выполнения практической работы были успешные выполнены 3 упражнения.

# **Ссылка на репозиторий:**

https://github.com/Grigalgol/PHP\_3\_kurs

# **Ответы на вопросы:**

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.**

PHP поддерживает десять основных типов переменных: 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: 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.

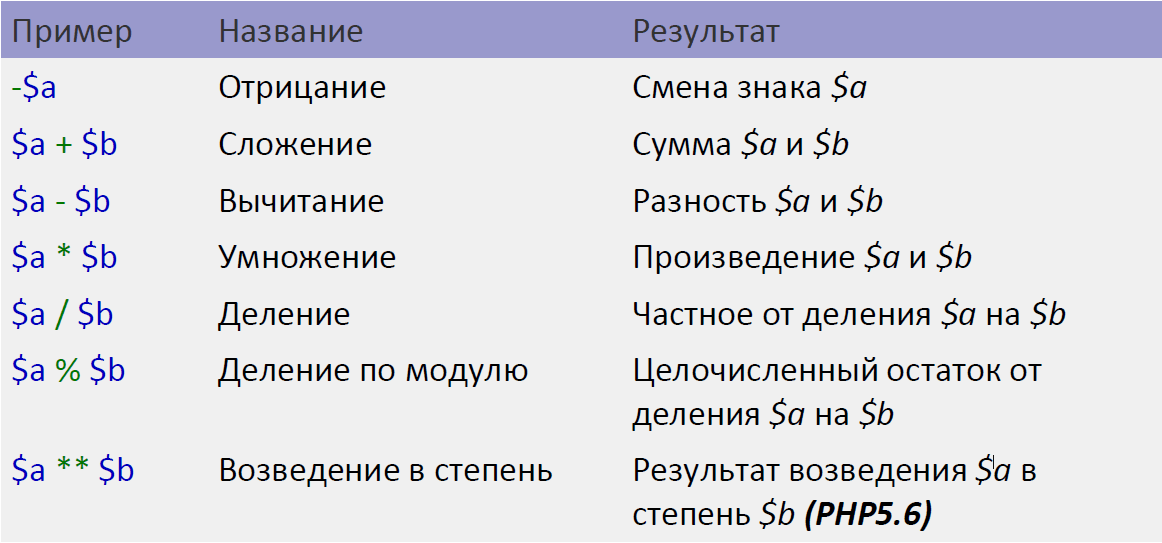
1. **Какие существуют функции для работы с переменными в php вне зависимости от типа данных.**

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

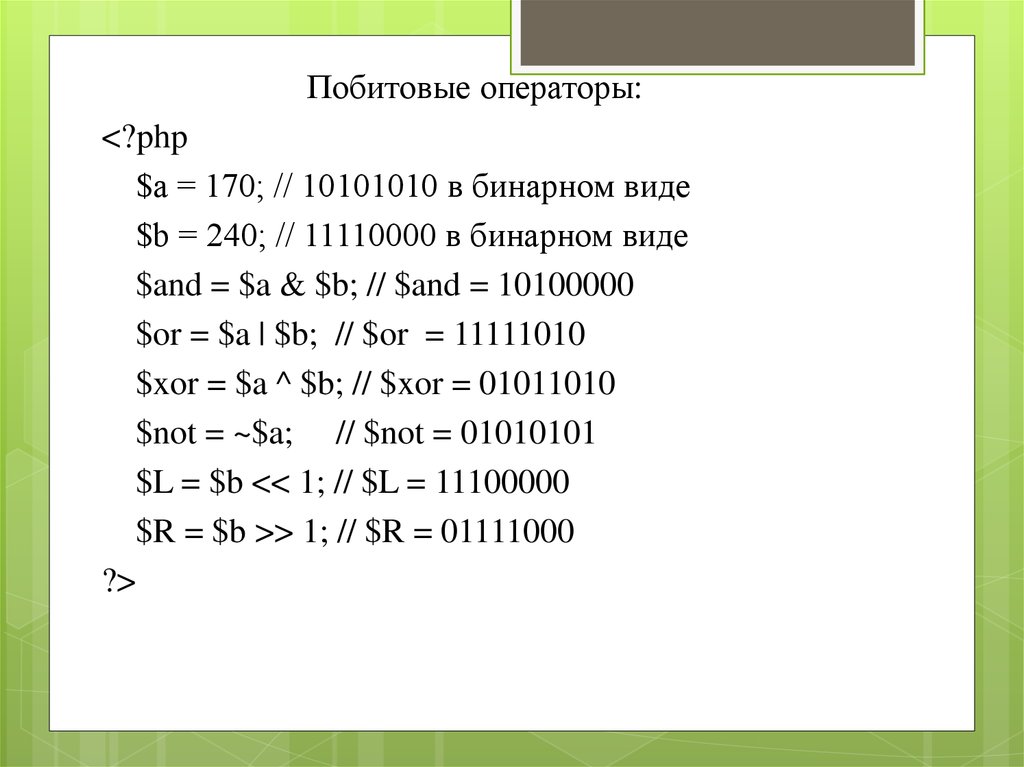
1. **Предопределенные переменные в php**

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.**



1. **Битовые операции в php**



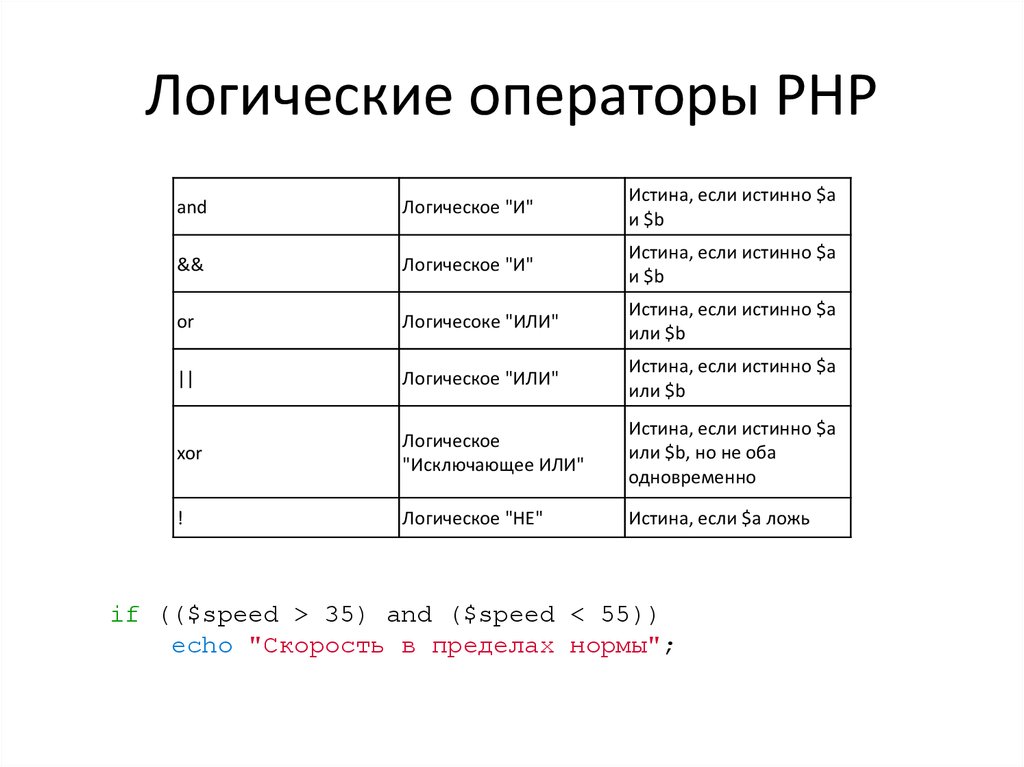
1. **Операторы присваивания.**



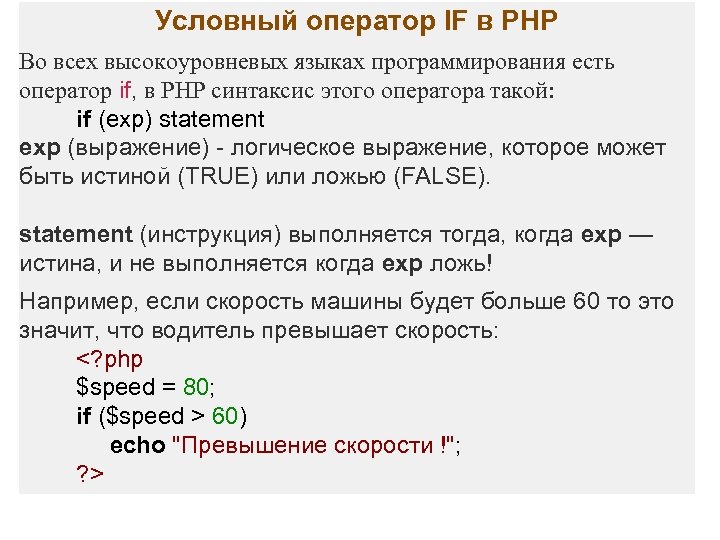
1. **Операторы сравнения**



1. **Логические операторы**



1. **Условная конструкция в php.**



1. **Циклы в php.**

PHP поддерживает три 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.**

