Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет информационных технологий и управления

Кафедра информационных технологий автоматизированных систем

Отчет по лабораторной работе №1 по дисциплине

ТЕХНОЛОГИИ ИНТЕРНЕТ-ПРОГРАММИРОВАНИЯ

на тему

«Основы протокола *HTTP*. Установка веб-сервера. Основы *PHP*»

|  |  |
| --- | --- |
| Выполнил ст. гр. 820601  Проверил преп. каф. ИТАС | А.С. Хасан  А.Л. Гончаревич |

Минск 2022

СОДЕРЖАНИЕ

[Введение 3](#_Toc94186760)

[1 Постановка задачи 4](#_Toc94186761)

[2 Теоритическая часть 5](#_Toc94186762)

[3 Ход работы 6](#_Toc94186763)

[Заключение 11](#_Toc94186764)

# ВВЕДЕНИЕ

Протокол передачи гипертекста (*Hypertext Transfer Protocol* — *HTTP*) —это прикладной протокол для передачи гипертекстовых документов, таких как *HTML*. Он создан для связи между веб-браузерами и веб-серверами, хотя в принципе *HTTP* может использоваться и для других целей. Протокол следует классической клиент-серверной модели, когда клиент открывает соединение для создания запроса, а затем ждёт ответа. *HTTP* — это протокол без сохранения состояния, то есть сервер не сохраняет никаких данных (состояние) между двумя парами «запрос-ответ». Несмотря на то, что *HTTP* основан на *TCP*/*IP*, он также может использовать любой другой протокол транспортного уровня с гарантированной доставкой.

*PHP* (рекурсивный акроним словосочетания *PHP*: *Hypertext Preprocessor*) — это распространённый язык программирования общего назначения с открытым исходным кодом. *PHP* специально сконструирован для веб-разработок и его код может внедряться непосредственно в *HTML*.

# ПОСТАНОВКА ЗАДАЧИ

Ознакомиться с основами *PHP*: базовый синтаксис, работа с переменными, константы, типы данных, преобразование типов, операторы и операции.

# ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

*HTTP* — это протокол, позволяющий получать различные ресурсы, например *HTML*-документы. Протокол *HTTP* лежит в основе обмена данными в Интернете. *HTTP* является протоколом клиент-серверного взаимодействия, что означает инициирование запросов к серверу самим получателем, обычно веб-браузером (*web-browser*). Полученный итоговый документ может состоять из различных поддокументов, являющихся частью итогового документа: например, из отдельно полученного текста, описания структуры документа, изображений, видео-файлов, скриптов и многого другого.

Клиенты и серверы взаимодействуют, обмениваясь одиночными сообщениями (а не потоком данных). Сообщения, отправленные клиентом, обычно веб-браузером, называются запросами, а сообщения, отправленные сервером, называются ответами.

*PHP* – это широко используемый язык сценариев общего назначения с открытым исходным кодом.

Говоря проще, *PHP* это язык программирования, специально разработанный для написания *web*-приложений (сценариев), исполняющихся на *Web*-сервере. Аббревиатура *PHP* означает «*Hypertext Preprocessor* (Препроцессор Гипертекста)». Синтаксис языка берет начало из *C*, *Java* и *Perl*. *PHP* достаточно прост для изучения. Преимуществом *PHP* является предоставление *web*-разработчикам возможности быстрого создания динамически генерируемых *web*-страниц.

Значительным отличием *PHP* от какого-либо кода, выполняющегося на стороне клиента, например, *JavaScript*, является то, что *PHP*-скрипты выполняются на стороне сервера. Вы даже можете сконфигурировать свой сервер таким образом, чтобы *HTML*-файлы обрабатывались процессором PHP, так что клиенты даже не смогут узнать, получают ли они обычный *HTML*-файл или результат выполнения скрипта.

*Open Server Panel* — это портативная программная среда, созданная специально для веб-разработчиков с учётом их рекомендаций и пожеланий. Включает в себя: *Apachi, Nginx, MySQL, MongoDB, PostgreSQL, PHP, PHPMyAdmin, FTP FileZilla.*

# 3 ХОД РАБОТЫ

Работа выполняется с помощью редактора кода – *Visual Studio Code*. Работа с программой начинается с создания *PHP* файла, в который будут помещаться скрипты.

В задании 1.1 необходимо было с помощью оператора *echo* вывести на страницу целочисленную переменную, переменную дробного типа, переменную булевского типа, строковую переменную, константу. Результат выполнения задания показан на рисунке 3.1.

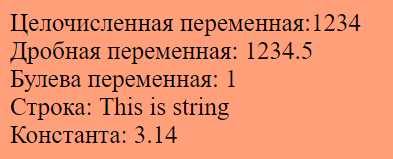


Рисунок 3.1 – Задание 1.1

В задании 1.2 необходимо было повторить вывод, заключив переменные в двойные кавычки (“). Результат выполнения задания продемонстрирован на рисунке 3.2.

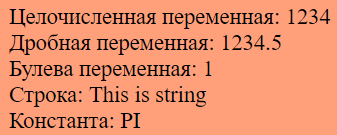


Рисунок 3.2 – Задание 1.2

За исключением константы, результат не отличается от предыдущего задания, так как если внутри строки с двойными кавычками написать имя переменной, то интерпретатор *PHP* подставит значение этой переменной при результирующем выводе.

В задании 1.3 необходимо было повторить вывод, заключив переменные в двойные кавычки (‘). Результат выполнения задания продемонстрирован на рисунке 3.3.

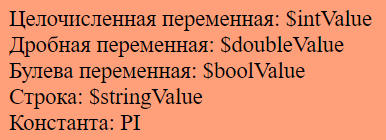


Рисунок 3.3 – Задание 1.3

Результат отличается от задания 1.1, так как если внутри строки с одинарными кавычками написать имя переменной, то подстановки значения не происходит.

В задании 1.4 необходимо было вывести на экран любое четверостишие. Для каждой новой строки использовать отдельный оператор *echo*. Каждая строчка должна быть отдельной строковой переменной. Также необходимо использовать переводы строки. После четверостишия поставить инициалы автора и сделать их подчёркнутыми. Результат выполнения задания показан на рисунке 3.4.

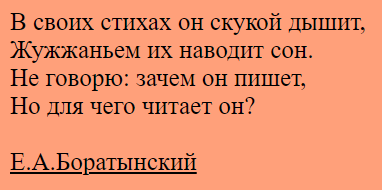


Рисунок 3.4 – Задание 1.4

Скрипт к этому заданию:

*<?php*

*$line1 = "В своих стихах он скукой дышит,";*

*$line2 = "Жужжаньем их наводит сон.";*

*$line3 = "Не говорю: зачем он пишет,";*

*$line4 = "Но для чего читает он?";*

*echo $line1."</br>";*

*echo $line2."</br>";*

*echo $line3."</br>";*

*echo $line4."</br></br>";*

*echo "<u>Е.А.Боратынский</u>";*

*?>*

В задании 1.5 необходимо выполнить эти же действия, с помощью одного оператора *echo*. Результат выполнения задания показан на рисунке 3.5.

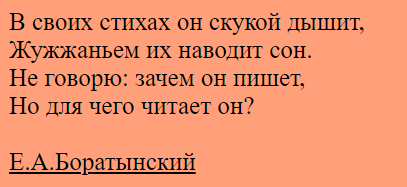


Рисунок 3.5 – Задание 1.5

Скрипт к этому заданию:

*<?php*

*$line1 = "*В своих стихах он скукой дышит*,";*

*$line2 = "*Жужжаньем их наводит сон*.";*

*$line3 = "*Не говорю: зачем он пишет*,";*

*$line4 = "*Но для чего читает он?*";*

*echo "$line1</br>$line2</br>$line3</br>$line4</br></br><u>Е.А.Боратынский</u>";*

*?>*

В задании 1.6 необходимо в выражении использовать разные типы данных. В данном задание используется число «100» и строка «44 человека». Результат выполнения задания показан на рисунке 3.6.



Рисунок 3.6 – Задание 1.6

Скрипт к этому заданию:

*<?php*

*$intValue = 100;*

*$stringValue = "44 человека";*

*echo $intValue + $stringValue;*

*?>*

В результате было получено число «144», так как при сложении этих переменных строка была неявно приведена к целому числу.

В задании 1.7 необходимо было продемонстрировать работу оператора *xor*. Результат выполнения задания показан на рисунке 3.7.

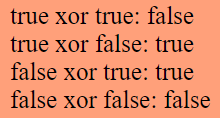


Рисунок 3.7 – Задание 1.7

Результат выражения $*a xor* $*a* для любых значений $*a* всегда будет равен *false.*

В задании 1.8 необходимо поменять значения переменных местами. При этом, использовать другие переменные запрещается. Результат выполнения задания показан на рисунке 3.8.

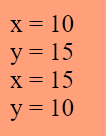


Рисунок 3.8 – Задание 1.8

Скрипт для этого задания:

*<?php*

*$x = 10;*

*$y = 15;*

*echo "x = $x</br> y = $y</br>";*

*$x = $x + $y;*

*$y = $x - $y;*

*$x = $x - $y;*

*echo "x = $x</br> y = $y</br>";*

*?>*

# ЗАКЛЮЧЕНИЕ

*HTTP* — лёгкий в использовании расширяемый протокол. Структура клиент-сервера, вместе со способностью к простому добавлению заголовков, позволяет *HTTP* протоколу продвигаться вместе с расширяющимися возможностями Сети.

*PHP* позволяет создавать качественные *web*-приложения за очень короткие сроки, получая продукты, легко модифицируемые и поддерживаемые в будущем.

*PHP* прост для освоения, и вместе с тем способен удовлетворить запросы профессиональных программистов.

Язык *PHP* постоянно совершенствуется, и ему наверняка обеспечено долгое доминирование в области языков *web*-программирования, по крайней мере, в ближайшее время.

При выполнении лабораторной работы мною были изучены основы *PHP*: базовый синтаксис, работа с переменными, преобразование типов, операторы и операции.