**Лабораторная работа 21**

Настройка ПО сервера, разработка простейших скриптов на php

**Задание №1.**

Установка и настройка

программного обеспечения, необходимого для разработки php-скриптов.

1.1. Установить с помощью пакета OpenSerwer локальный web-сервер Apache, интерпретатор PHP, СУБД MySQL и другие вспомогательные расширения.

1.2. Проверка корректности работы установленного программного обеспечения.

**Задание №2.**

Изучение синтаксиса PHP, разработка простейших PHP-скриптов.

1. Создайте отдельную страницу, на которой будут размещаться ссылки на php-cкрипты.

2. Получите информацию о настройках php с помощью команды phpinfo().

3. Создайте php-скрипт, который создает страницу с текстом "Привет всем!!!" и содержит информацию о разработчике скрипта.

4. Используя переменные $color и $size сформировать php-скрипт , который выводит на экран строку текста ФИО РАЗРАБОТЧИКА заданным цветом и размером.

5. Создайте константу NUM\_E со значением 2.71828 (число е). Выведите на экран значение этой константы в виде строки "Число e равно …".

Присвойте переменной $num\_e1 значение константы и отобразите тип переменной $num\_e1. Последовательно измените тип переменной $num\_e1 на строковый, целый и булевский; отобразите тип и получившееся значение переменной $num\_e1.

6. Приведите пример использования предопределенных констант и переменных в PHP. Выведите с помощью скрипта значения некоторых из них.

Контрольные вопросы:

1 Опишите назначение пакета OpenSerwer. Укажите назначение утилит, входящих в его состав.

Назначение пакета OpenServer: OpenServer — это локальный веб-сервер, который позволяет разработчикам тестировать и разрабатывать веб-приложения на PHP и других языках программирования. Он включает в себя такие компоненты, как Apache (веб-сервер), MySQL (система управления базами данных), PHP (язык программирования) и другие утилиты. Утилиты, входящие в его состав, могут включать:

* phpMyAdmin: интерфейс для управления базами данных MySQL.
* Composer: менеджер зависимостей для PHP.
* Xdebug: отладчик и профайлер для PHP.
* Node.js: для работы с JavaScript на серверной стороне.

Эти компоненты позволяют разработчикам легко настраивать и управлять средой разработки.

2 Почему возникает необходимость использовать локальный web-сервер при написании и тестировании php-скриптов?

Необходимость использования локального web-сервера: Локальный веб-сервер необходим для разработки и тестирования PHP-скриптов, так как PHP является серверным языком программирования. Он обрабатывается на сервере, и без веб-сервера невозможно увидеть результат работы скрипта в браузере. Локальный сервер позволяет разработчикам тестировать свои приложения в безопасной среде без необходимости загружать их на удаленный сервер.

3 Опишите методы встраивания PHP-кода в html-код. Приведите примеры.

Методы встраивания PHP-кода в HTML-код: PHP-код можно встраивать в HTML с помощью тега <?php ... ?>. Например:

<html>

<body>

<h1>Добро пожаловать!</h1>

**<?php**

echo "Это PHP-код, встроенный в HTML.";

**?>**

</body>

</html>

Также можно использовать короткие теги:

<html>

<body>

<h1>Добро пожаловать!</h1>

**<?=** "Это короткий PHP-код." **?>**

</body>

</html>

4 Какие операторы PHP используются для вывода результатов работы скрипта в html-документ? Приведите примеры.

Операторы PHP для вывода результатов: Для вывода результатов работы скрипта в HTML-документ используются операторы echo и print. Примеры:

<?php

**echo** "Привет, мир!";

**print** "Это пример использования print.";

?>

5 Опишите правила использования переменных в PHP. Приведите примеры.

Правила использования переменных в PHP: Переменные в PHP начинаются с символа $, за которым следует имя переменной. Имя переменной может содержать буквы, цифры и символы подчеркивания, но не может начинаться с цифры. Примеры:

**$name** = "Иван";

**$age** = 25;

6 Что такое предопределенные переменные в PHP. Приведите примеры.

Предопределенные переменные в PHP: Это переменные, которые создаются PHP автоматически и содержат информацию о текущем окружении. Примеры:

**$\_GET**: массив, содержащий данные, переданные через метод GET.

**$\_POST**: массив, содержащий данные, переданные через метод POST.

**$\_SESSION**: массив, содержащий данные сессии.

7 Какая функция PHP позволяет определить тип данных переменной? Какая функция PHP позволяет проверить существование переменной? Какая функция PHP позволяет удалить переменную?

Функции для работы с переменными:

* Для определения типа данных переменной используется функция gettype(). Пример:

$var = 10;

echo gettype($var); // выводит "integer"

* Для проверки существования переменной используется функция isset(). Пример:

if (isset($var)) {

echo "Переменная существует.";

}

* Для удаления переменной используется функция unset(). Пример:

unset($var); // удаляет переменную $var

8 Опишите правила использования констант в PHP. Приведите примеры.

Правила использования констант в PHP: Константы в PHP определяются с помощью функции define(). Имя константы обычно записывается в верхнем регистре. Пример:

define("PI", 3.14);

echo PI; // выводит 3.14

9 Что такое предопределенные константы в PHP. Приведите примеры.

Предопределенные константы в PHP: Это константы, которые автоматически создаются PHP. Примеры:

PHP\_VERSION: возвращает версию PHP.

PHP\_OS: возвращает название операционной системы, на которой работает PHP.

10 Опишите правила использования комментариев к PHP-коду. Приведите примеры.

Правила использования комментариев в PHP: В PHP можно использовать однострочные и многострочные комментарии. Однострочные комментарии начинаются с // или #, а многострочные — с /\* и заканчиваются на \*/. Примеры:

// Это однострочный комментарий

/\*

Это многострочный комментарий.

\*/