**Лабораторная работа 23**

Использование стандартных функций PHP для работы с файлами, датой, временем и календарем.

**Задание 1.**

Использование стандартных функций PHP для работы с файлами и каталогами. (выполнить ПРИМЕРЫ №1-4)

**Задание №2.**

Использование встроенных функций PHP, предназначенных для работы с датой, временем и календарем.

1. Подготовить текст программы для вывода текущей даты и времени в три строки:

- в первой строке дата в кратком формате (например, 1. 04. 2023),

- во второй строке время,

- в третьей – день недели.

2. Написать специальную функцию, которая бы возвращала дату с написанием только дня недели на русском языке. Для проверки работоспособности функции вызвать ее для текущей даты.

3. Подготовить текст программы , обеспечивающий следующее:

- На экран выводится текущая дата в числовом формате и форма с одной кнопкой, например: При нажатии кнопки на экран выводится сообщение “Сегодня день недели (по-русски) “.- Вывод русского написания дня недели должен обеспечиваться сконструированной Вами функцией.

Рекомендации.

В функции определения дня недели можно использовать следующую конструкцию.

Ввести вспомогательную переменную $weekday=date(&#39;D&#39;);

А затем произвести сравнение значения переменной $weekday со всеми английскими сокращенными названиями дней недели (Mon, Tue, Wed, Thu, Fri, Sat, Sun). При выполнении условия произвести переопределение вспомогательной текстовой переменной $weekday.

if ($weekday = = &#39;Mon&#39;) { $weekday=&quot;понедельник&quot;}; и д.т.

\*Во всех разработанных скриптах использовать комментарии к программному коду.

Контрольные вопросы:

1. Опишите основные этапы обработки файлов в PHP. Для реализации каких функциональных элементов сайта целесообразно использовать работу с файлами?

Обработка файлов в PHP включает несколько основных этапов:

* Открытие файла: Для работы с файлом его необходимо открыть с помощью соответствующих функций. Это может быть сделано в режиме чтения, записи или добавления.
* Чтение или запись данных: После открытия файла можно либо читать данные из него, либо записывать данные в него. В зависимости от режима открытия файла используются разные функции.
* Закрытие файла: После завершения работы с файлом его нужно закрыть, чтобы освободить ресурсы.
* Обработка ошибок: Важно обрабатывать возможные ошибки, такие как отсутствие файла, недостаток прав на чтение/запись и т.д.
* Функциональные элементы сайта, где целесообразно использовать работу с файлами:
* Загрузка и хранение пользовательских данных: Например, сохранение профилей пользователей или настроек.
* Логи и отчеты: Запись действий пользователей или системных событий для последующего анализа.
* Импорт и экспорт данных: Чтение и запись данных в формате CSV или текстовых файлов.
* Хранение изображений и медиафайлов: Сохранение загруженных пользователями файлов на сервере.

2. Укажите какие функции PHP предназначены для выполнения следующих действий с файлами: открытие файла, чтение данных из файла, запись данных в файл, закрытие файла, блокировка файла, проверка существования файла, признак конца файла, удаление файла. Приведите примеры их использования.

Вот функции PHP, предназначенные для выполнения различных действий с файлами:

Открытие файла: fopen()

$file = fopen('example.txt', 'r'); // Открыть файл для чтения

Чтение данных из файла: fread(), fgets(), file\_get\_contents()

$content = fread($file, filesize('example.txt')); // Чтение всего содержимого файла

Запись данных в файл: fwrite(), file\_put\_contents()

file\_put\_contents('example.txt', "Новый текст\n"); // Запись данных в файл

Закрытие файла: fclose()

fclose($file); // Закрытие файла

Блокировка файла: flock()

if (flock($file, LOCK\_EX)) { // Блокировка файла для эксклюзивного доступа

// Работа с файлом

flock($file, LOCK\_UN); // Снятие блокировки

}

Проверка существования файла: file\_exists()

if (file\_exists('example.txt')) {

echo "Файл существует.";

}

Признак конца файла: feof()

while (!feof($file)) {

$line = fgets($file);

// Обработка строки

}

Удаление файла: unlink()

unlink('example.txt'); // Удаление файла

3. Укажите назначение функций PHP, предназначенных для работы с датой, временем и календарем, которые вы использовали при выполнении заданий из лабораторной работы.

Вот некоторые функции PHP, которые используются для работы с датой и временем:

date(): Форматирует текущую дату и время.

echo date('j. n. Y'); // Выводит текущую дату в формате "1. 04. 2023"

strtotime(): Преобразует строку даты в метку времени.

$timestamp = strtotime('now'); // Получение текущего времени в формате timestamp

time(): Возвращает текущее время в формате timestamp.

$currentTime = time(); // Получение текущего времени

mktime(): Создает метку времени из заданных параметров даты и времени.

$timestamp = mktime(0, 0, 0, 4, 1, 2023); // Получение timestamp для 1 апреля 2023 года

getdate(): Возвращает ассоциативный массив с информацией о дате.

$dateInfo = getdate(); // Получение информации о текущей дате