

ЛАБОРАТОРНАЯ РАБОТА №2

ИСПОЛЬЗОВАНИЕ PHP

Цель работы: получить навык разработки сайтов с использованием языка программирования PHP

Задачи:

1. Разработать сайт на PHP, отображающий информацию из файлов и позволяющий проводить запись в файл.
2. Разместить сайт на веб-сервере.

Результатами работы являются:

1. Разработанный сайт, содержащий PHP и CSS файлы.
3. Подготовленный отчет.

СОЗДАНИЕ ДИНАМИЧЕСКИХ САЙТОВ С ПОМОЩЬЮ PHP

Статические сайты состоят из неизменяемых страниц. Это значит, что сайт имеет один и тот же внешний вид, а также одно и то же наполнение для всех посетителей. При запросе такого сайта в браузере сервер сразу предоставляет готовый HTML-документ, CSS и JS файлы.

Динамические сайты, в свою очередь, имеют изменяемые страницы, адаптирующиеся под конкретного пользователя. Такие страницы не размещены на сервере в готовом виде, а собираются заново по каждому новому запросу.

PHP (англ. PHP: Hypertext Preprocessor) — скриптовый язык программирования общего назначения, применяемый для разработки веб-приложений. В настоящее время поддерживается подавляющим большинством хостинг-провайдеров и является одним из самых распространенных среди языков программирования, применяющихся для создания динамических веб-сайтов.

PHP-скрипт в общем случае представляет из себя некий шаблон HTML-страницы, в которой с помощью специального тега `<?php>` добавляются вставки PHP-кода. Интерпретатор выполнит код на странице в порядке появления, в итоговой сгенерированной странице вместо каждого из блоков `<?php>` будет результат выполнения блока кода.

```
<html>
  <head>
    <title>My First Page</title>
  </head>
  <body>
    <?php
      echo "Hello world!";
    ?>
  </body>
</html>
```

Типы данных в PHP

Boolean – логический тип, который содержит значение true или false.

Integer – содержит значения целого числа.

String – содержит значение текста произвольной длины.

Float – вещественное число.

Object – объект.

Array – массив.

Resource – ресурс (например, файл).

NULL – значение NULL.

Для объявления переменных используется специальный знак «\$», команда echo возвращает значение из скрипта, например:

```
$text = "news"  
echo "This is {$text}paper";
```

Таблица 2.1. Логические операторы сравнения

<code>\$a == \$b</code>	Равно
<code>\$a === \$b</code>	Тождественно равно
<code>\$a != \$b</code>	Не равно
<code>\$a <> \$b</code>	Не равно
<code>\$a !== \$b</code>	Тождественно не равно

Для создания массивов можно использовать одну из следующих конструкций:

```
$arr1 = array("php", "html", "css");  
$arr2 = array(1 => "php", "html", "css");  
$arr3 = ["php", "laravel", "yii", "zend", "cakephp"];
```

Для удаления элемента из массива можно использовать метод unset:

```
unset($arr[1])
```

Для итерации по элементам массива можно использовать цикл foreach:

```
$array = array ("Apple", "Limon", "Chery", "Oranges");
foreach ($array as $value)
{
    echo "Вы выбрали фрукт - $value <br>";
}
```

Работа с формами в PHP

Создание формы, где в атрибуте action указан PHP-скрипт обработчик формы:

```
<?php header("Content-Type: text/html;
    charset=utf-8");?>
<form action="app/check.php" method="post">
    <p>Имя: <input name="name" type="text"></p>
    <p>Фамилия: <input name="surname" type="text"></p>
    <p>E-mail: <input name="email" type="text"></p>
    <p>Сообщение: <br /><textarea name="message"
        cols="30" rows="5"></textarea></p>
    <p><input type='submit' value='Отправить'></p>
</form>
<?php>
```

Обработка полученной формы:

```
<?php
    if($_SERVER['REQUEST_METHOD'] == 'POST') {
        $name = $_POST['name'];
        $surname = $_POST['surname'];
        $email = $_POST['email'];
        $message = $_POST['message'];
        echo "name: $name <br/> surname: $surname
            <br/> email: $email
            <br/> message: $message";
    }
?>
```

Работа с файлами в PHP

Для открытия файла используется функция `fopen`, принимающая в качестве параметра имя файла и режим доступа к файлу. Возможные значения режимов доступа:

- `r` – открытие файла только для чтения.
- `r+` – открытие файла одновременно на чтение и запись.
- `w` – создание нового пустого файла. Если на момент вызова уже существует такой файл, то он уничтожается.
- `w+` – аналогичен `r+`, только если на момент вызова файл такой существует, его содержимое удаляется.
- `a` – открывает существующий файл в режиме записи, при этом указатель сдвигается на последний байт файла (на конец файла).
- `a+` – открывает файл в режиме чтения и записи при этом указатель сдвигается на последний байт файла (на конец файла). Содержимое файла не удаляется.

Добавляя «`t`» работа с файлом будет происходить в текстовом режиме.

Примеры открытия файлов:

```
<?php
    $fp = fopen('counter.txt', 'r'); // Бинарный режим
    $fp = fopen('counter.txt', 'rt'); // Текстовый режим
    $fp = fopen("http://www.yandex.ru", "r");
    $fp = fopen("ftp://user:password@example.ru", 'w');
?>
```

Для чтения данных из файла можно воспользоваться одной из функций:

- `fgets($file)` – строчное чтение файла;
- `file_get_contents($file_name)` – загрузка всего содержимого файла;
- `fread($file, 100)` – чтение из файла указанного количества байт, переданного вторым параметром.

Для генерации исключения в случае невозможности открыть файл используется функция `die()`, принимающая на вход текст исключения. Примеры чтения данных из файлов:

```
<?php
    $fd = fopen("file.dat", 'r') or die("не удалось
        открыть файл");
    while(!feof($fd))
    {
        $str = fgets($fd);
        echo $str;
    }
    fclose($fd);

    $str = file_get_contents("file.dat");
    echo $str;

    $fd = fopen("file.dat", 'r') or die("не удалось
        открыть файл");
    while(!feof($fd))
    {
        $str = fread($fd, 600);
        echo $str;
    }
    fclose($fd);
?>
```

Запись в файл выполняется с помощью функции `fwrite`:

```
<?php
    $fd = fopen("hello.txt", 'w') or die("не удалось
        создать файл");
    $str = "Привет мир";
    fwrite($fd, $str);
    fclose($fd);
?>
```

После того как работа с файлом завершена, его необходимо закрыть с помощью функции `fclose()`.

ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

Разработать сайт тематики, согласно полученному варианту.

Сайт должен быть стилизован с помощью CSS (возможно использование CSS-фреймворка Bootstrap).

На сайте должна быть отображена таблица, содержащая информацию, хранимую на веб-сервере в отдельном текстовом файле (возможно использование формата xml или json).

На сайте должна быть предусмотрена возможность добавления новых данных (с сохранением в файл).

ВАРИАНТЫ ЗАДАНИЙ

1. Сайт школы
2. Сайт фитнес-центра
3. Новостной сайт
4. Метеорологический сайт
5. Сайт музыкального исполнителя
6. Сайт кинотеатра
7. Сайт отеля
8. Сайт аэропорта
9. Сайт телеканала
10. Сайт радиостанции
11. Сайт автосалона
12. Сайт ресторана
13. Сайт университета
14. Сайт библиотеки
15. Сайт театра
16. Сайт ветеринарной клиники
17. Сайт туристической фирмы
18. Сайт агентства по продаже недвижимости
19. Сайт букмекерской компании
20. Сайт биржи криптовалют

КОНТРОЛЬНЫЕ ВОПРОСЫ И ЗАДАНИЯ

1. Перечислите типы данных в PHP.
2. Приведите код для работы с массивами в PHP.
3. Приведите команду, с помощью которой можно вернуть значение из PHP-скрипта.
4. Приведите логические операторы сравнения в PHP.
5. Приведите функцию PHP для генерации исключения.
6. Опишите механизм обработки формы с помощью PHP.
7. Перечислите режимы доступа к файлам в PHP.
8. Опишите механизм работы с файлами с помощью PHP.

ФОРМА ОТЧЕТА ПО ЛАБОРАТОРНОЙ РАБОТЕ

На выполнение лабораторной работы отводится 5 часов: 4 часа на выполнение работы, 1 час на подготовку и защиту отчета по лабораторной работе.

Структура отчета: титульный лист, цель и задачи, формулировка задания (вариант), этапы выполнения работы, исходный код разработанного сайта, результаты выполнения работы (скриншоты), выводы.