***Лабораторная работа №7***

**Тема:** Использование форм для передачи данных на сервер

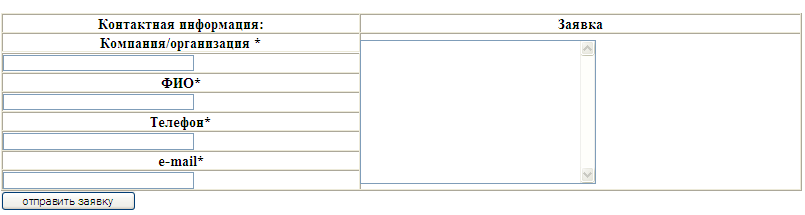
**Цель:** Получить навыки работы формами средствами php

**Ход выполнения работы:**

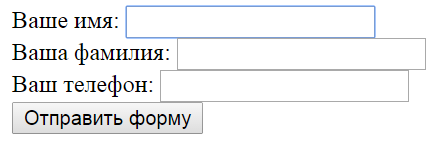
1. Изучить теоретическую часть.
2. Написать отчет по проделанной работе. В отчете представить результат выполненных заданий:

**Задание 1.** Набрать предложенные примеры форм на языке php, продемонстрировать их работоспособность с помощью web-сервера

**Задание 2.** Создать форму следующего вида, которая после нажатия отправки, должна выводить страницу с содержанием: **ФИО отправителя, спасибо, Ваша заявка принята!**



**Задание 3.** Создать форму отправки имени, фамилии и номера телефона. При отправке формы заносить в файл имя, фамилию, телефон и IP-адрес посетителя.



***Примечание:*** для вывода информации о ip-адресе использовать $ip=$\_SERVER['REMOTE\_ADDR']; //массив, содержащий информацию, такую как заголовки, пути и местоположения скриптов. Записи в этом массиве создаются веб-сервером. 'SERVER\_ADDR' IP адрес сервера, на котором выполняется текущий скрипт.

**Пример 1**

|  |  |
| --- | --- |
| *Файл .html* | **Файл process.php** |
|  |  |

***Теоретическая часть***

***Как получить данные и передать их для обработки?***

Для этой цели используются формы — это совокупность стандартных HTML-конструкций ввода текстовой и прочей информации и программы-обработчика этой информации, работающей на Web-сервере. Иными словами, пользовательская форма (или HTML-форма) служит для передачи информационных данных серверу. Результат конструкций языка разметки HTML интерпретируется браузером, с помощью которого пользователь электронного документа получает информацию. Таким образом, объединив все эти формулировки, можно сказать, что HTML-форма выступает в роли посредника между пользователем и сервером. Посетитель Web-страницы вводит в HTML-форму определенные данные, которые обрабатываются программой и отсылаются на Web-сервер. Все эти действия укладываются в три стадии:

* Ввод пользователем информации.
* Обработка введенной информации программой, установленной на сервере.
* Получение результата отправления введенной информации на Web-сервер (открытие нового HTML-документа, переадресация на предыдущую страницу и пр.).

В качестве программы-обработчика чаще всего выступает CGI-сценарий (скрипт, который обычно разрабатывается на языке Perl или C/C++ и который взаимодействует со специальным компонентом Web-сервера —Common Gateway Interface) или программы, написанные на основе таких серверных языков программирования, как PHP, ASP, JSP и др.

Значение пользовательских форм трудно переоценить — они являются особым средством HTML, дающим посетителю возможность не только пассивно просматривать информацию, но и быть задействованным в Обмене информацией между Web-сервером и клиентом содержании Web-сайта.

**Способы передачи параметров сценарию**

Работа с формами в PHP и HTML

Формы обозначаются тегом form:

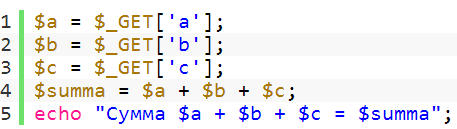
|  |  |
| --- | --- |
|  | ***<form method="post">***  ***</form>*** |

**Атрибут method** указывает каким методом будет производиться отправка формы. Существуют 2 метода – GET и POST.

**Передача переменных при помощи метода GET**

Данный метод передачи переменных применяется в PHP для передачи переменных в файл при помощи адресной строки. То есть переменные передаются сразу через адресную строку браузера.

Теперь давайте более подробно на примере рассмотрим работу метод GET. Пускай нам нужно передать в файл три переменных $a, $b и $c **методом GET**и вывести их сумму на экран. Для этого можно использовать следующий код.



Поскольку все переменные перед передачей будут помещаться в **глобальный массив GET**, то мы первым делом присваиваем нашим переменным значения соответствующих элементов массива GET. Делаем это в самом начале во избежание разнообразных ошибок при передаче переменных. Далее для демонстрации работы прописываем произвольную формулу и выводим результат на экран.

Для проверки работы метода GET достаточно просто добавить к ссылке на файл знак вопроса «?» и через амперсанд «&» перечислить переменные с их значениями.

**localhost/php/ file.php?a=1&b=2&c=3**

Теперь при переходе по этой ссылке нам выведется сумма переменных $a, $b и $c.

При использовании <form method="GET" ... >, к URL просто добавляются знак вопроса и данные формы.

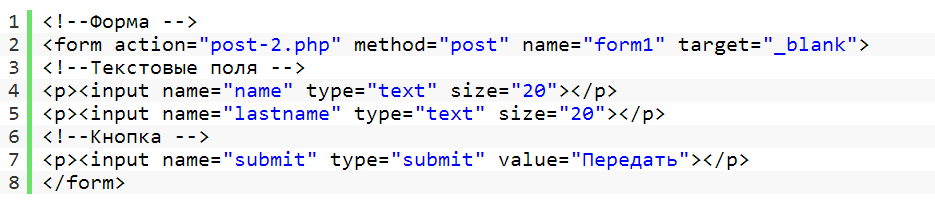
Размеры данных, которые можно передать таким способом ограничены 32 килобайтами, поэтому для передачи более массивных данных используется метод POST, который встраивает данные в тело HTTP запроса к серверу.

**Передача переменных в PHP при помощи метода POST**

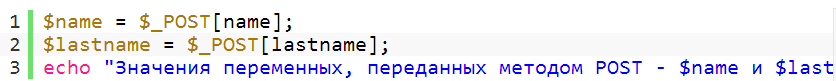
Преимущество метода POST - отсутствие ограничения на длину строки с данными формы.

Данный способ позволяет скрыто передавать переменные с одного файла в другой. Как вы уже поняли, обычно для этих целей используются два файла. В первом находится форма для ввода исходных данных, а во втором исполнительный файл, который принимает переменные. Для демонстрации давайте рассмотрим следующий код.

Код первого файла с формой для отправки данных. Дадим ему название post1.php



Код второго файла, который будет служить приемником переменных. Назовем его post2.php



Как и с методом GET, мы первым делом присваиваем переменным значения соответствующих элементов **глобального массива POST**. Далее для наглядности выводим эти переменные на экран при помощи оператора вывода echo.

Теперь при загрузке первого файла у нас загрузится форма. После введения данных жмем по кнопке «Отправить», в результате чего в новой вкладке у нас откроется страница со вторым файлом, на которой выведутся значения, прописанные в форме на предыдущей странице. То есть второму файлу будут переданы значения переменных с первого файла.