

1. Как правильно настроить форму для отправки данных в обработчик? Какие атрибуты прописать, какие значения?

Ответ. В тег `<form>` нужно прописать атрибут `action` со значением пути, на который будет отправлена форма (например `action='handler.php'`) и атрибут `method` со значением `'post'` или `'get'`. Внутри тега `form` нужно прописать все необходимые поля. У каждого поля есть атрибуты: `name` и `value`. Также необходима кнопка в форме.

2. Что такое обработчик? Какую задачу он решает? А зачем выделять его в отдельный файл? В чем польза такого подхода?

Ответ. Обработчик занимается обработкой запросов, которые поступают к нему из форм в верстке. В отдельном файле его выделяют для того чтобы было легче поддерживать код и масштабировать его. Таким образом получается разделение верстки и внутренней логики проекта.

3. Какие методы бывают отправки данных через форму? В чем их разница? В каком случае использовать один, а когда другой?

Ответ. Метод `GET` отправляет данные через URL, добавляя к нему параметры, разделённые знаком `'&'`. Метод подходит для коротких запросов, так как длина запроса ограничена размером URL. Данные видны в адресной строке. Можно использовать для фильтров поиска на странице. Метод `POST` передает данные на сервер в теле запроса, и они не видны в адресной строке. Можно использовать для передачи конфиденциальной информации. В большинстве случаев используется метод `POST`.

4. Опишите процесс записи строки в базу данных: как подключиться, подготовить запрос и выполнить.

Ответ. Подключаемся с помощью класса `PDO`. Указываем тип СУБД, адрес хоста, название базы данных, логин и пароль для БД (`$pdo = new PDO("mysql:host=localhost;dbname=db24", "root", "");`). Далее подготавливаем запрос SQL (`$sql = "INSERT INTO texts (text) VALUES (:text)";`) методом `prepare` (`$statement = $pdo->prepare($sql);`). Затем выполняем запрос методом `execute` (`$statement->execute(["text" => $text]);`).

5. Какой глобальный массив используется при создании флеш-сообщений?

Ответ. Используется глобальный массив `$_SESSION`.

6. Опишите пошагово как создать и вывести флеш-сообщение.

Ответ. Настраиваем форму, указываем в атрибуте `action` путь до обработчика, а в `method` ставим `'post'`. Далее в обработчике записываем нужные нам данные в сессию и возвращаемся обратно на страницу с формой. В этот момент в массиве `$_SESSION` хранятся нужные нам данные и мы можем вывести их в виде флеш-уведомления в определенном месте в верстке.

7. Как проверить текст на формат эл. адреса средствами php?

Ответ. Можно использовать функцию `filter_var` с флагом `FILTER_VALIDATE_EMAIL`. Например `filter_var('user@mail.com', FILTER_VALIDATE_EMAIL);`

8. Какие функции используются для генерации пароля и его проверки?

Ответ. Для генерации пароля можно использовать функцию `password_hash()`. Для проверки

пароля существует функция `password_verify()`.

9. Почему важно сохранять пароль в виде хэша? Почему не рекомендуется хранить пароль в открытом виде?

Ответ. Таким способом пароль хранится в БД в зашифрованном виде и его очень проблематично расшифровать. В случае утечки данных пароли будут защищены. Получается, что никто кроме пользователя не знает пароля, даже админ БД.

10. Напишите какая есть особенность у веб-сервера которую покрывает Сессия? Иными словами, зачем нужна Сессия?

Ответ. Сессия нужна для сохранения данных в процессе работы. Даже после перезагрузки страницы, данные в массиве `$_SESSION` не пропадают. Используется для запоминания данных авторизованного пользователя, а также для флеш-уведомлений.

11. Опишите пошагово реализацию функционала Аутентификации(входа в систему).

Ответ. Пользователь заполняет поля логина и пароля в форме и нажимает кнопку отправки данных. Обработчик подхватывает данные и проверяет наличие такого логина в БД. Если логин есть, то проверяется пароль к нему. При совпадении пары логин-пароль перенаправляем пользователя на нужную страницу. При несоответствии комбинации перенаправляем на страницу авторизации и выводим флеш-сообщение «Неверный логин или пароль».

12. Какая есть особенность настройки формы для загрузки файлов?

Ответ. В теге `<form>` нужно указать атрибут `enctype="multipart/form-data"`. А также в теге `<input>` написать атрибут `type='file'`.

13. Какое значение должно быть у инпута для множественной загрузки файлов?

Ответ. В теге `<input>` значение атрибута `name` должно быть с `[]` на конце. Например `name='image[]'`. Также должен быть добавлен атрибут `multiple`.

14. Как создать уникальное название файла?

Ответ. Можно использовать функцию `uniqid()`.

15. Напишите названия функций для загрузки файла, проверки существует ли файл в папке, удаления файла.

Ответ. Загрузка файла: `move_uploaded_file`. Проверка существования файла: `file_exists`.
Удаление файла: `unlink`.

16. Какой глобальный массив хранит все файлы передаваемые через форму?

Ответ. Глобальный массив `$_FILES`.

17. Опишите структуру глобального массива хранящего файлы: какие имеет ключи и значения.

Ответ. Ключ `name` (имя файла на компьютере пользователя), ключ `type` (тип файла), ключ `tmp_name` (временное имя файла на сервере), ключ `error` (код ошибки, если она возникла), ключ `size` (размер файла в байтах).

18. Помогите человеку решить задачу. «Пользователь загрузил файл и нажимает отправить. Он ожидает, что файл сохранится в папку, а название в базу. Файл сохраняется в папку, но в базу название не сохраняется» В чем может быть ошибка?

Ответ. Ошибка в обработке взаимодействия с БД. Возможные причины: ошибка при подключении к БД, неправильный sql запрос, неправильное название таблицы БД.

19. Помогите человеку решить задачу. «Создал массив элементов, и хочу вывести при помощи цикла. Вроде бы все написал, но выходит ошибка: illegal offset type» Что нужно сделать, чтобы решить эту проблему?

Ответ. Нужно убедиться, что при выводе массива в ключах содержатся допустимые значения. Ключи могут иметь типы только string или integer.

20. Что означает эта ошибка. «**Parse error Unexpected (, expected ;**» Что нужно сделать, чтобы избавиться от ошибки?

Ответ. Проблема в синтаксисе. Нужно найти место, где возникает ошибка, и проверить синтаксис в этом месте. Убедиться, что правильно используются скобки, и что каждая инструкция заканчивается символом ; .

21. Помогите человеку решить задачу. «Создал форму с инпутом. Пишу текст и отправляю. В обработчике \$_POST пустой» В чем может быть ошибка?

Ответ. Возможно отсутствует атрибут name в теге input. Или значения name в форме и обработчике разные. Например в форме name='email', а в обработчике \$_POST['e-mail'].

22. Чем технически отличается глобальный массив от обычного массива?

Ответ. Глобальный массив доступен для всех функций и скриптов на сервере, а обычный массив доступен только внутри той функции или скрипта, где он был создан. Если изменить значение элемента в глобальном массиве, то это изменение отразится во всех функциях и скриптах на сервере, использующих этот массив. С обычным массивом такое не произойдет.

23. Дайте человеку совет. Он говорит: «Когда я пишу код, я не обращаю внимания на названия переменных, функций и массивов. Для меня все и так понятно. Поэтому я даю короткие названия, типа \$a, \$b, \$smth. Но потом, понимаю, что мне тяжело ориентироваться в них»

Ответ. Нужно давать емкие названия для переменных и других элементов, чтобы при просмотре и изменении кода не вспоминать для чего эта переменная, функция и т.д. Тем более, если с кодом работает не один человек.

24. Дайте человеку совет. Он говорит: «Хочу самостоятельно написать регистрацию и авторизацию. Приступаю и у меня ступор: не знаю с чего начать. Вроде бы понимаю, что регистрация — это добавление пользователя в базу, но не понимаю с чего начинать?»

Ответ. Необходимо разбить общую задачу на несколько маленьких. Создать форму с полями для ввода логина и пароля. Затем определить, что будет происходить при нажатии на кнопку и в какой обработчик будут отправлены данные. Создать обработчик и сделать там взаимодействие с БД. Таким образом нужно последовательно решить все задачи для достижения общей цели.

25. Объясните человеку в чем разница между \$_SESSION и \$_COOKIE: «Не пойму в чем разница между сессией и куки? Вроде бы одно и то же...пока полностью не осознал.»

Ответ. COOKIE хранятся в веб-браузере клиента в открытом виде. Сессия хранится на сервере. В сессии никто не может подсмотреть данные на пользователя. Сервер присваивает уникальный идентификатор сессии для пользователя и отправляет ему. Он хранится в COOKIE у пользователя. Затем по присвоенному идентификатору сервер может идентифицировать пользователя, когда тот снова зайдет на сайт.

26. Дайте человеку совет. Он говорит: «Скачал кучу курсов. По одним названиям можно понять, что я скоро точно стану программистом. Но сколько бы я не смотрел, у меня не получается самостоятельно написать даже простенький скрипт. Что я делаю не так? Как надо подходить к обучению?»

Ответ. Нужно много писать код руками, а не только смотреть видео. Мозг намного быстрее научится, если много практиковаться, а не изучать только теорию. Если начинает получаться, то можно постепенно усложнять задачи или пробовать написать код за меньший промежуток времени.

27. Какие у вас впечатления от первой части практики?

Ответ. Всем доволен.