

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

Ответ: В самой форме:

Метод отправки данных - GET или POST в зависимости от данных. GET для фильтров и поиска. POST для обработки личных данных;

Путь к файлу-обработчику или же пустота, если данные обрабатываются в самом файле верстки;

Enctype, если форма подразумевает прикрепление файлов.

Для элементов формы: Это name и type для input и for для label-a. Name пишется для присвоения названия к данным, а for для привязки label к input. Type пишется для вида данных и меняет форму инпута.

Для кнопки: необязательно, но также можно настроить атрибут type.

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

Ответ: Обработчик это файл со скриптом, где прописана логика действий с данными. Он обрабатывает введенные пользователем данные. Чаще всего логику прописывают в отдельном файле-обработчике для сохранения читабельности кода и во избежание хаоса.

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

Ответ: Методы отправки данных - GET или POST используется в зависимости от данных. GET для фильтров и поиска. POST для обработки личных данных. Главное отличие: GET данные видны в строке запроса, а POST данные нет.

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

Ответ: Сначала нам нужно прописать название хоста, таблицы, логин и пароль аккаунта. Позже соединяем все это через DSN или SQL connect, в зависимости от метода подключения. Передаем все это в одну переменную, обычно называется \$conn. Далее пишем sql запрос и сохраняем ее в переменную. Если это подготовленный запрос то подменяем значение данных во избежание sql инъекций. Под конец выполняем запрос и полученные данные храним в другой переменной.

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

Ответ: `$_SESSION`

6. Опишите пошагово как создать и вывести флеш-сообщение

Ответ: Сначала инициализируем сессий в файлах и прописываем логику обработки данных. Здесь же создаем сообщения в куки сессий, если данные не будут соответствовать условиям. Передаем эти куки через сессию в другой файл и через условные операторы показываем сообщения.

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

Ответ: Я пользуюсь функцией `filter_var($email, FILTER_VALIDATE_EMAIL)`. Указываю его в атрибут условного оператора и продолжаю дорабатывать логику. Сам функционал проверяет наличие в введенных данных «@» и «.com/kz/ru etc».

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

Ответ: Пароль вводится самим пользователем и после хэшируется на стороне сервера через `password_hash($password, PASSWORD_DEFAULT)`. При входе пользователя, пароль проверяется через функцию `password_verify`.

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

Ответ: При взломе данных, злоумышленник может получить все данные пользователей. А хэш пароли не дешируются.

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

Ответ: Сессий могут хранить данные пользователей и помогают их идентифицировать. Таким образом сервер понимает, что пользователь уже заходил на наш сайт и идентифицирует его на основе куки его браузера.

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

Ответ: Сначала создаем форму для ввода данных пользователя. Оттуда берем логин или же почту и через запрос в бд проверяем, есть ли такая запись. Если ответ положительный, то берем пароль и проверяем его с хэш паролем в бд с помощью `password_verify`. Если

данные сходятся, то пользователь авторизуется, если нет, то он получает флэш-сообщение.

12. Какая есть особенность настройки формы для загрузки файлов?

Ответ: В атрибуте формы нужно указать `enctype = «multipart/form-data»`.

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

Ответ: `multiple`

14. Как создать уникальное название файла?

Ответ: С помощью функций `uniqid()`.

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

Ответ: `move_uploaded_file()`, `file_exists()`, `unlink()`.

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

Ответ: `$_FILES`.

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

Ответ: `$_FILES` ассоциированный массив из пяти ключей:

`name` - название файла, `tmp_name` - путь к временной папке, где хранится файл, `error` - код ошибки, `size` - размер файл, `type` - расширение файла.

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: Нужно смотреть на ошибку и на сам код. Возможно он не соединился с базой или на стороне базы ограничение на количество символов в сторке.

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 завардампить получаемый элемент. Наверное тип данных не относится к массиву.

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

Ответ: Забыл точку с запятой после кода. Нужно перепроверить код.

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

Ответ: Нужно посмотреть на код. Наверное не указал action у формы или name для инпута.

22. Чем технически отличается глобальный массив от обычного массива?

Ответ: Глобальный массив уже создан в системе и пользователь туда лишь приписывает значения. Обычные массивы создаются пользователем.

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: Названия нужно указывать по функционалу и смыслу элемента для читабельности кода. Также на работе в больших компаниях нужно понимать, что над ним будут работать еще другие разработчики и им тоже должно быть понятно.

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: Можно начать с обычной схемы для прорисовки логики. Позже на его основе писать код. Или же посмотреть в интернете реализацию других разработчиков.

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: \$_SESSION это ассоциативный массив, который хранится на стороне сервера данные о пользователе. \$_COOKIE это идентификационный номер, который хранится в браузере пользователя.

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: Мало практики. Чем чаще пишешь код, тем яснее становится алгоритм действий.

27. Какие у вас впечатления от первой части практики?

Ответ: Интересно, нравится, что задания уже готовы и тебе остается лишь подумать над кодом и написать его. Но в некоторых уроках нет текстового сопровождения к заданиям и чтобы понять, что нужно сделать, нужно посмотреть ролик.