1. Назначение языка JavaScript. Способы размещения сценариев JavaScript. Структура

кода

2. Переменные в JavaScript. Требования к переменным

3. Типы данных в JavaScript. Функции преобразования типов данных

4. Виды диалоговых окон JavaScript. Функции их вызова

5. Условные конструкции языка JavaScript, их синтаксис

6. Операторы цикла JavaScript, их синтаксис. Операторы break и continue.

7. Массива в JavaScript. Основные виды, способ организации. Свойства и методы

8. События в JavaScript. Пользовательские функции. Синтаксис создания функции

9. Назначение языка PHP. Возможности PHP. Основы синтаксиса

10. Переменные, константы и типы данных в PHP

11. Управляющие конструкции в PHP, их синтаксис

12. Организация передачи данных пользователя через форму в PHP. Методы передачи

данных.

13. Подключение внешних файлов средствами PHP

14. Сессии и куки

15. Основы работы с MySQL через PHP

**Ответы:**

1. JavaScript используется для создания интерактивных веб-страниц и придания им динамического поведения. Сценарии JavaScript можно разместить внутри тега <script> в HTML-документе, внешнем файле .js или встроить непосредственно в HTML-код страницы. Структура кода включает объявление переменных, операторы, условные конструкции, циклы и функции.

2. Переменные в JavaScript используются для хранения данных. Они могут быть объявлены с помощью ключевых слов var, let или const. Требования к переменным включают правила именования (должны начинаться с буквы, знака подчеркивания или знака доллара), не могут совпадать с зарезервированными словами и чувствительны к регистру.

3. В JavaScript есть различные типы данных, такие как числа, строки, логические значения (true/false), объекты, массивы и другие. Функции преобразования типов данных, такие как parseInt(), parseFloat(), String() и Boolean(), позволяют изменять тип данных одного значения на другой.

4. В JavaScript существуют разные виды диалоговых окон, такие как окно alert, confirm и prompt. Функция alert выводит сообщение, confirm выводит окно с вопросом и кнопками для выбора, а prompt выводит окно с полем ввода. Для вызова этих окон используют соответствующие функции: alert(), confirm() и prompt().

5. Условные конструкции в JavaScript позволяют выполнять различные действия в зависимости от условий. Их синтаксис включает ключевые слова if, else if и else для выполнения различных блоков кода в зависимости от условий.

6. Операторы цикла в JavaScript, такие как for, while и do...while, позволяют многократно выполнять блок кода. Синтаксис for: for (начальное значение; условие; шаг) { // блок кода }. Операторы break и continue используются для контроля выполнения цикла: break прерывает цикл, а continue пропускает текущую итерацию и переходит к следующей.

7. Массивы в JavaScript используются для хранения упорядоченных наборов данных. Основные виды массивов включают числовые, строковые и объектные массивы. Для создания массива можно использовать квадратные скобки и запятые для разделения элементов. Массивы имеют свойства и методы, такие как length (длина массива), push (добавление элемента в конец массива) и join (объединение элементов массива в строку).

8. События в JavaScript позволяют отслеживать действия пользователя или изменения веб-страницы. Пользовательские функции могут быть созданы для обработки событий. Синтаксис создания функции включает ключевое слово function, имя функции, аргументы (необязательно) и блок кода, который будет выполняться при вызове функции.

9. PHP - это язык программирования общего назначения, который используется преимущественно для разработки веб-приложений. Он обладает мощными возможностями для обработки данных на сервере и генерации динамического контента. Основы синтаксиса PHP включают теги <?php и ?> для обозначения PHP-кода, а также использование точки с запятой (;) в конце каждой инструкции.

10. В PHP переменные используются для хранения и обработки данных. Константы - это именованные значения, которые не могут изменяться в процессе выполнения программы. Типы данных в PHP включают целые числа (int), числа с плавающей запятой (float), строки (string), булевы значения (bool), массивы (array), объекты (object) и другие.

11. Управляющие конструкции в PHP используются для управления потоком выполнения программы. Они включают условные операторы (if, else, elseif), циклы (for, while, do-while, foreach) и операторы перехода (break, continue, return). Синтаксис этих конструкций в PHP достаточно похож на другие языки программирования.

12. Для организации передачи данных пользователя через форму в PHP можно использовать методы GET и POST. Метод GET передает данные через URL в виде параметров, а метод POST отправляет данные в теле HTTP-запроса. В PHP эти данные могут быть доступны через глобальные переменные $\_GET и $\_POST, соответственно.

13. В PHP внешние файлы могут быть подключены с помощью функции include или require. Функция include позволяет включить файл и продолжить выполнение программы, даже если файл не существует или содержит ошибки. Функция require работает аналогично, но если файл не может быть подключен, программа будет прервана.

14. Сессии и куки используются в PHP для хранения состояния между запросами пользователя. Сессии сохраняют данные на сервере и привязывают их к уникальному идентификатору сессии, который обычно передается через куки. Куки представляют собой небольшие файлы, которые хранятся на компьютере пользователя и содержат информацию, отправляемую сервером.

15. Для работы с MySQL через PHP можно использовать расширение mysqli или PDO. С помощью этих расширений PHP может устанавливать соединение с базой данных, выполнять запросы (например, SELECT, INSERT, UPDATE, DELETE) и получать результаты. Работа с MySQL в PHP включает выполнение SQL-запросов, обработку результатов и управление транзакциями. (mysqli\_connect())