





Workshop

Урок 4





Регламент

-  Получаем задание
-  Выполняем задание определенное время
-  Проверяем правильность выполнения
-  Переходим к новому заданию



Задания





Задание 1 (тайминг 20 минут)

1. Создайте функцию `delayedMessage(message, delay)`, которая принимает аргументы `message` (строка) и `delay` (число). Функция должна выводить заданное сообщение в консоль через указанную задержку.
2. Вызовите функцию `delayedMessage()` три раза с разными сообщениями и задержками. Например:
 - a. `delayedMessage("Сообщение 1", 2000)`
 - b. `delayedMessage("Сообщение 2", 1000)`
 - c. `delayedMessage("Сообщение 3", 3000)`
3. После вызова всех функций `delayedMessage()`, добавьте сообщение вида "Конец программы" с помощью `console.log()`.





Ожидаемый результат

Сообщение 2

Сообщение 1

Сообщение 3

Конец программы





Задание 2

Напишите программу, которая загружает данные с сервера с использованием объекта XMLHttpRequest и отображает полученную информацию в консоли.

1. Создайте функцию `loadData(url)`, которая принимает аргумент `url` (строка) - адрес сервера для загрузки данных.
2. Внутри функции `loadData()` создайте объект XMLHttpRequest с помощью `new XMLHttpRequest()`.
3. Зарегистрируйте обработчик события `onreadystatechange`, который будет вызываться при изменении состояния запроса. Проверьте, если `readyState` равен 4 (успешно выполнен запрос) и `status` равен 200 (успешный статус ответа сервера), то выведите полученные данные в консоль с помощью `console.log(xhr.responseText)`.





Задание 2 продолжение

4. Откройте запрос с помощью `xhr.open("GET", url, true)`, где "GET" - тип запроса, url - адрес сервера, true - асинхронный режим запроса.
5. Отправьте запрос на сервер с помощью `xhr.send()`.

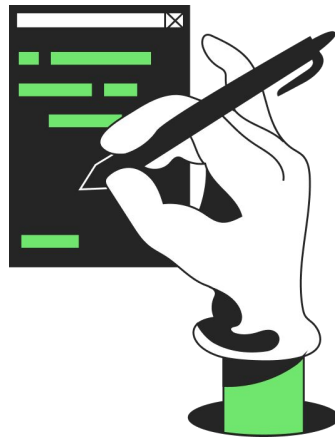




Задание 3 (тайминг 35 минут)

Напишите функцию `getData`, которая делает асинхронный запрос к API и возвращает данные в виде объекта.

Внутри функции происходит асинхронный запрос к API с помощью функции `fetch`. Затем, с использованием оператора `await`, ожидается ответ от сервера и его парсинг в формате JSON с помощью метода `response.json()`. Полученные данные возвращаются из функции.





Задание 3 Дополнение

В случае возникновения ошибки при выполнении асинхронных операций, используется конструкция `try/catch` для обработки и отображения ошибки.


В функции `main` вызывается функция `getData` с использованием `await`, чтобы получить данные асинхронно. Полученные данные выводятся в консоль. Если происходит ошибка, она ловится и выводится сообщение об ошибке. Функция `main` также объявлена как асинхронная с использованием ключевого слова `async`. В конце вызывается функция `main` для запуска процесса получения данных.





Итоги



Спасибо 
за внимание

