

Работа с API и регулярными выражениями

Описание:

Это проектное задание направлено на практику работы с асинхронными запросами (fetch) и обработку полученных данных с помощью регулярных выражений, минуя стандартный парсинг JSON. Вы будете извлекать и форматировать информацию непосредственно из "сырого" текста.

Требования:

Создание страницы:

- Создайте файл index.html.
- Добавьте на страницу заголовок <h1> с текстом "Обработка данных о товарах".
- Создайте пять кнопок с id:
 - id="load" с текстом "Загрузить товары".
 - id="price" с текстом "Вывести все цены".
 - id="brand" с текстом "Вывести все бренды".
 - id="date" с текстом "Отформатировать даты".
- Создайте многострочное текстовое поле <textarea> с id="output", rows="25" и cols="100".
- Подключите к странице JavaScript-файл (например, script.js).

Получение данных (JavaScript):

- В файле script.js получите ссылки на все HTML-элементы с помощью document.getElementById().
- Привяжите event listener к кнопке "Загрузить товары".
- При клике на эту кнопку выполните fetch()-запрос к <https://dummyjson.com/products>.
- Используйте метод .text() для получения ответа в виде обычной строки.
- Выведите полученный текст целиком в <textarea id="output">. Это нужно, чтобы студенты визуально увидели "сырой" JSON-текст, с которым им предстоит работать.

Обработка данных регулярными выражениями:

Создайте обработчики кликов для остальных трёх кнопок.

- Кнопка "Вывести все цены":
 - Получите текст из `<textarea>`.
 - Используйте регулярное выражение для поиска всех цен.
 - С помощью метода `.matchAll()` или цикла `while` с `.exec()` извлеките только значения (числа).
 - Выведите найденные цены в `<textarea>`.
- Кнопка "Вывести все бренды":
 - Получите текст из `<textarea>`.
 - Используйте регулярное выражение для поиска и извлечения названий брендов.
 - Выведите найденные бренды в `<textarea>`.
- Кнопка "Отформатировать даты":
 - Получите текст из `<textarea>`.
 - Используйте регулярное выражение для поиска дат.
 - Используйте метод `.replace()` с обратными ссылками (`$1`, `$2`, `$3`), чтобы изменить формат даты с `YYYY-MM-DD` на `DD.MM.YYYY`.
 - Выведите изменённый текст в `<textarea>`.

Вывод результата:

- Каждый раз, когда пользователь нажимает на одну из кнопок "Обработать", `textarea` должен обновляться новым, обработанным результатом.

Шпаргалка по регулярным выражениям:

Группы захвата (`()`): Всё, что попадает в скобки, сохраняется как отдельная группа.

`"price":\s*(\d+) — (\d+)` захватывает одну или более цифр.

`"brand":\s*"([^\"]+)" — ([^\"]+)` захватывает один или более символов, кроме кавычки. Это надёжный способ получить текст внутри кавычек.

Флаг `g` (Global): Позволяет найти все совпадения в строке, а не только первое.

`matchAll()`: Возвращает итератор, который содержит все совпадения, включая захваченные группы. Это удобный и современный метод для извлечения данных.

`replace()` с обратными ссылками:

`'текст'.replace(/(\d{4})-(\d{2})-(\d{2})/, '$3.$2.$1')` поменяет 2024-09-11 на 11.09.2024.

`$1`, `$2`, `$3` — это ссылки на содержимое первой, второй и третьей захваченных групп.