Министерство науки и высшего образования Российской Федерации

Федерального государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г.В. Плеханова»

**МОСКОВСКИЙ ПРИБОРОСТРОИТЕЛЬНЫЙ ТЕХНИКУМ**

Специальность: 09.02.07 Информационные системы и программирование

Квалификация: Программист

ПРАКТИЧЕСКИЕ РАБОТЫ

ПО «РАЗРАБОТКЕ ПРОГРАММНЫХ МОДУЛЕЙ»

Листов: 5

|  |  |
| --- | --- |
| Выполнил студент | Проверил преподаватель |
| Группы П50-4-22 | \_\_\_\_\_\_\_\_\_\_\_\_Д.В.Серяк |
| Юрченко Вячеслав Валерьевич | «\_\_\_» \_\_\_\_\_\_\_\_2024 года |

Москва 2024

ПРАКТИЧЕСКАЯ РАБОТА №9

***Цель работы:***

* Создать объекты с различными свойствами и методами, представляющими различные сущности (Это тема вашей работы: автомобили, компьютеры, фильмы и тд.).
* Реализовать функционал для взаимодействия с этими объектами, включая добавление новых объектов, изменение и удаление существующих объектов.
* Отобразить информацию о объектах на веб-странице, используя DOM для динамического обновления содержимого.
* Использовать чистый JavaScript без использования библиотек или фреймворков.
* Обеспечить возможность добавления, удаления и изменения объектов через пользовательский интерфейс.
* Реализовать валидацию вводимых данных при добавлении или изменении объектов.
* Обеспечить удобное отображение информации о объектах на веб-странице с использованием HTML и CSS.

***Ход работы:***

1. Для написания методов и функций сайта необходимо понимать с чем требуется работать. Для этого стоит написать простейшую страницу использую html и css (по желанию), а затем уже приступить к программированию на js.



Рисунок 1 – разметка html

Добавляем на страницу поля для вводов и указываем для каждого поля подсказки для пользователя, чтобы он понимал, какие значения должны указывать в них. В семантическом теге head с помощью линка мы привязываем файл style.css, находящийся в папке проекта style. А скрипт-файл script.js привязываем в конце файла перед концом тега body. Также указываем для файла полный путь.

1. Теперь требуется открыть файл script.js и в нём начать создавать разные элементы, которые будут функционировать прямиком на сайте.



Рисунок 2 – скрипт 1 часть

В первой части скриншота скрипт-файла можно увидеть, что здесь создаётся функция создания фильма, внутри которой созданы постоянные переменные заголовка, жанра, года и рейтинга по фильму, которые принимают значения через область document по идентификатору (id), который указан в самом index.html файле.

С помощью функции trim() нужно убрать все пробелы в полях, дабы не получилось так, что можно создать элемент на странице, не имеющий ничего в своих значениях.

Выполнить проверку на наличие данных в полях, если в них ничего нету, то следует предупредить пользователя, что дальнейшая работа с добавлением не возможна, и ему необходимо повторить действия снова.

Для переменной film создаём определённые элементы, после чего использовать что-то вроде конструктора, который возвращает значения имеющихся элементов.

Далее, с помощью метода push() необходимо отослать данные из переменной film и использовать метод updateFilmList(), разбор которого будет позже.

И за основу начальных значений полей следует указать пустые строки.



Рисунок 3 – скрипт 2 часть

Функция updateFilmList обновляет содержимое элемента на веб-странице с идентификатором film-list, чтобы отобразить список фильмов.

Функция очищает содержимое элемента film-list. Проходит по массиву объектов films и для каждого фильма создает новый элемент списка `<li>` и добавляет ему класс `film-item`. Задает текстовое содержимое элемента списка как результат вызова метода displayInfo() для фильма. Создает кнопку "Изменить" и добавляет ее в элемент списка. Создает кнопку "Удалить" и добавляет ее в элемент списка.

Функция возвращает обновленный список фильмов в элементе film-list.

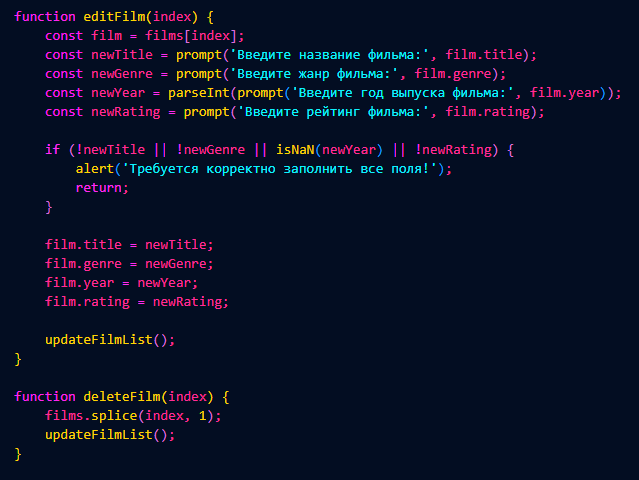


Рисунок 4 – скрипт 3 часть

Функция editFilm редактирует фильм в массиве films по указанному индексу.

Запрашивает с помощью prompt значения для добавления их на место существующих.

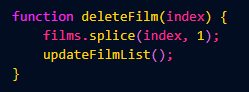


Рисунок 5 – скрипт 4 часть

Функция deleteFilm удаляет фильм из массива films по указанному индексу.

Удаляет фильм из массива films по указанному индексу. Вызывает функцию updateFilmList для обновления отображения списка фильмов.

***Вывод:*** Реализована система для управления объектами типа "Фильм" с помощью чистого JavaScript, которая позволяет добавлять новые фильмы, изменять их содержание и полностью удалять. Также есть возможность просмотра списка добавленных фильмов.