Оглавление

[Введение 2](#_Toc168608154)

[Основная часть 2](#_Toc168608155)

[Ознакомление с кодом 3](#_Toc168608156)

[Анализ по улучшению 6](#_Toc168608157)

# **Введение**

Данный отчет был написан с целью объяснить программный код, написанный на языке программирования JavaScript, для ранее созданного интерфейса небольшого веб-приложения (написанный при помощи языка гипертекстовой разметки HTML и каскадных таблиц стилей CSS) по практическому заданию №1 производственной практике.

# **Основная часть**

В процессе проектирования кода были решены следующие задачи, необходимые для правильной работы приложения:

Реализация некого фильтра (в нашем случае, когда при загрузке страницы или нажатии кнопки «ALL», то появляются все карточки-картинки со всех категорий под блоком, внутри которого располагаются все заголовки (названия) категорий; или же, когда мы нажимаем другую кнопку (кроме «ALL»), то под блоком категорий появляются карточки с картинками той тематики (категории), которую выбрал пользователь)

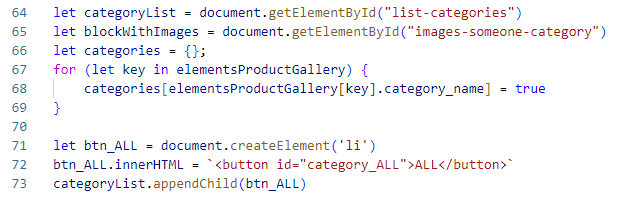
Динамическое добавление разделов. Если в «массиве», у некоторого элемента будет присутствовать иное (вне создание приложения) название категории, то оно будет динамически добавлено; это также касается и разделов, которые должны быть реализованы в текущих планах (all, print, web design, logo, motion). Т.е. если их название будет отсутствовать в одном из названий категорий (поле, находящееся у каждого из добавляющихся элементов-картинок (есть также поле img\_path)), то данной категории не будет существовать, т.к. в ней как раз-таки не существует какой-либо картинки, входящей в эту категорию.

# **Ознакомление с кодом**

На данной картинке:  


Показана часть кода, отвечающая за создание «массива», внутри которого находятся элементы, для каждого из которых присвоены **название категории**, в которой будут присутствовать, а также **путь картинки**

На этой картинке:

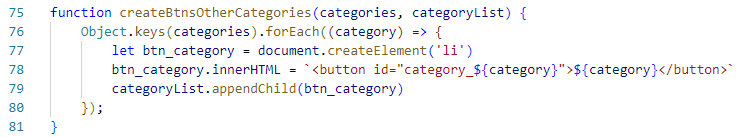


Показано создание родительских блоков, в которых в последующем будут создаваться прочие блоки, такие как «меню» категорий, а также картинки для определенных категорий;

Ниже представлено создание нового объекта, в который будем заносить/собирать (благодаря следующему циклу) уникальные категории из каждого элемента. Т.е. после выполнения цикла мы будем иметь обновленный объект, который будет содержать все уникальные категории из «массива» elementsProductGallery.

После, идет создание независимой кнопки с категорией ALL, в которой будет в любом случае отображаться любое количество картинок, вне зависимости, в какой прочей категории они будут находиться.

На картинках ниже:

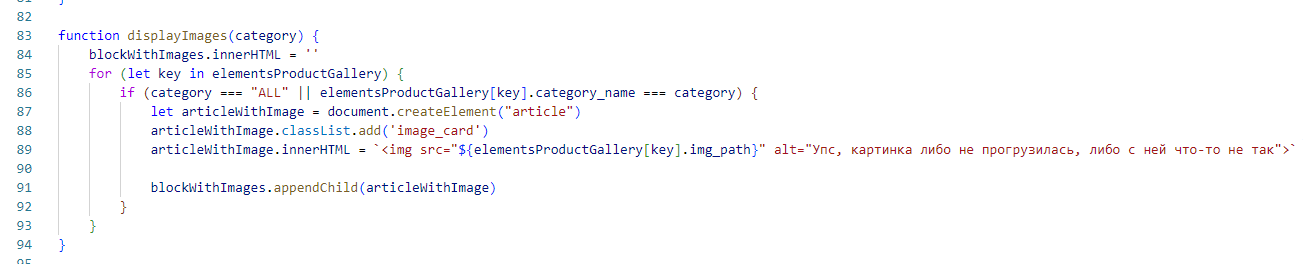




Показана реализация функции и ее вызов, которая принимает в качестве параметров наш новый недавно созданный объект вместе с родительским блоком, в котором будут находиться кнопки с названиями категорий.

Данная функция нужна для создания кнопок с названиями категорий и добавления их в родительский элемент списка (что ранее создавали) на странице.

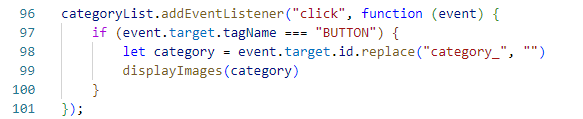
На следующем изображении:





Представлено создание функции displayImages, которая отвечает за отображение изображений на странице, которые фильтруются в зависимости от указанных категорий. При этом, если названием категории будет являться «ALL», то будут отображены все изображения. Также в строчке blockWithImages.innerHTML = '' происходит удаление содержимого , в котором будут отображаться изображения, для успешного удаления предыдущих изображений перед отображением новых картинок. А на строчке №104 происходит вызов функции, куда в качестве аргумента передаем значение «ALL», чтобы добавить категорию ALL, где будут находиться все картинки.

На следующем изображении:



Показан блок кода, который добавляет обработчик событий на «Клик» для кнопок с названиями категорий. Т.е. когда пользователь нажмет на одну из кнопок, обработчик определяет категорию из id кнопки и вызывает функцию displayImages(), для обновления изображений, которые отображаются на странице в соответствии для каждой из категорий.

# **Анализ по улучшению**

В текущей структуре у нас есть объект, где ключами являются id элементов, а их значениями — объекты, тем самым получается объект со вложенными объектами. Для улучшения структуры предлагается использовать структуру данных не как объект (что мы делали ранее, когда работали с такой структурой), а преобразовать его в массив объектов, где каждый объект будет представлять собой элемент некоторой картинки с «ключами» ‘category\_name’ и ‘img\_path’, где их свойствами будут являться название категории и путь к картинке соответственно. Такая структура в свою очередь будет проще для фильтрации и более управляемой для добавления и удаления элементов из массива. То есть для фильтрации и взаимодействия с массивом мы сможем использовать методы forEach, map, filter и другие.

Предполагаемый внешний вид измененной структуры:

До:  


После:

