

Можно использовать **любую парадигму** (ООП, ФП и т.д.), и код организовать можно **любым удобным способом**.

При написании решений **предпочтительнее использовать чистый ES2015** (кроме задачи №6 - там React). Результат лучше представить в виде ссылки на ресурс вроде <https://jsfiddle.net/> (это просто способ передачи кода, не нужно добиваться его работы именно в пределах ресурса).

Пожалуйста, не присылайте .rar-архивы — у нас тут открытое ПО, поэтому проприетарные форматы и архиваторы не поддерживаются.

1. Предположим, есть страница, на которой всего одно поле:

```
<style>.red {color: red;}</style>
<input type="text" name="name" id="name_input" value="Xxxx">
```

Задача: без хардкода и без использования глобальных переменных написать скрипт, который при вводе данных в поле будет добавлять ему класс red, если его текущее значение поля не совпадает с изначальным, и удалять этот класс, если значения совпадают

2. Допустим, что по url

http://любой_домен/filter?size=M&color=1&color=2&manufacturer=aaa&manufacturer=ddd находится страница, на которой есть такие поля:

- радио для size (значения - S, M, L)
- несколько чекбоксов для color (значения - 1, 2, 3, 4, 5)
- мультиселект (select multiple) для manufacturer (значения - "aaa", "b&c", "ddd", "eee")
- чекбокс "распродажа" (значение - 1)

Допустим также, что сервер при генерации html ни одно из полей не заполняет, то есть радио не выбран, чекбоксы пустые и т.д.

Задача: набросать самую элементарную разметку для указанных инпутов и написать скрипт, который

- при загрузке страницы разберёт значения фильтров из url и расставит их по соответствующим полям
- при изменении состояния в любом инпуте, кроме "распродажа", выведет в консоль аналогичный приведённому в условии url с актуальными значениями фильтров

3. Нужно написать скрипт на веб-странице, который должен обмениваться данными с сервером. При этом нужно послать два разных xhr-запроса подряд, а затем выполнить какой-то код.

Задача: написать пример кода, который отправит на сервер сразу два асинхронных запроса подряд и выведет **один** раз после получения **обоих** ответов в консоль сообщение “оба ответа получены”.

4. Существует страница на домене `domain.one`, на которой загружается `iframe` с другого домена - `domain.two`. Нужно на странице `domain.one` использовать скрипт для записи/чтения/удаления данных из `localStorage` домена `domain.two`. При этом оба домена под нашим управлением, то есть, мы можем изменять файлы и там, и там, как нам удобно.

Задача: написать реализацию методов для чтения/записи/удаления данных из доступного `localStorage` другого домена. Пусть при успешном чтении данные выводятся в консоль, а при успешной записи/удалении в консоль отправляется лог вроде “written” или “removed”.

5. Для работы с кросс-доменным `localStorage` понадобилось не только чтение/запись/удаление данных из него, но и их дополнительная обработка.

Задача: добавить в решение задачи 4 возможность передать callback и вызвать его на `domain.one` после выполнения операции чтения/записи/удаления.

6. С использованием библиотеки **React** реализовать приложение, которое умеет показывать следующие страницы:

- / — главная
- /employees — страница со списком сотрудников

На сайте, в шапке реализовать ссылки:

- Главная
- Сотрудники

Получение данных должно быть реализовано посредством вызова стороннего api — https://reqres.in/api/users?per_page=12

Страница Сотрудники содержит простой список сотрудников (только имена), на ней также есть возможность удаления и добавления нового сотрудника. Разумеется, отправлять результаты добавления и удаления никуда не нужно — их просто должно быть видно в текущем представлении.

Оформление (дизайн) — не важно.