

## Домашние задание #5

### #17 Учебный проект: в движении

#### Задача

Добавить интерактивности на страницу. Сделать диалог редактирования персонажа перетаскиваемым (draggable)

В модуле для работы с диалогом (`dialog.js`) реализуйте возможность перетаскивания диалога:

- Диалог должен начинать двигаться за курсором мыши при нажатии (`mousedown`) на блок `.setup-user-pic`;
- Диалог должен переставать двигаться за курсором мыши при отпускании (`mouseup`) кнопки мыши и оставаться на новом месте;
- При повторном открытии/закрытии диалога, положение диалога должно сбрасываться на изначальное;

### #18 Личный проект: максимум подвижности

#### Задача

Доработать алгоритм перемещения элементов по странице

Кекстаграм

В этом задании мы закончим работу над слайдером, задающим глубину эффекта, заставив его перемещаться.

Теперь, когда вы знакомы с тем, как работает механизм перетаскивания элементов, вы можете закончить работу над слайдером.

Вам нужно описать полный цикл Drag n Drop для пина маркера, то есть добавить обработчики событий `mousedown`, `mousemove` и `mouseup` на маркер.

Обработчики `mousemove` и `mouseup` должны добавляться только при вызове обработчика `mousedown`.

Обработчик `mousemove` должен запускать логику изменения положения пина: в нём должны вычисляться новые координаты пина на основании смещения, применяться через стили к элементу и записываться в поле уровня эффекта (с поправкой на то, что в это поле записываются координаты середины пина).

При перемещении, кроме состояния слайдера, должна меняться глубина эффекта, наложенного на изображение, то есть меняться значение CSS-фильтра, добавленного на изображение. Это нетривиальная задача, потому что значение CSS-фильтра записывается в одних границах, а положение слайдера в других. Вам нужно использовать пропорцию, чтобы рассчитать насыщенность правильно.

Учтите, что расчёт координат пина и их запись в поле должны дублироваться и в обработчике `mouseup`, потому что в некоторых случаях, пользователь может нажать мышью на слайдере, но никуда его не переместить.

Ещё один момент касается ограничения перетаскивания: не забудьте сделать так, чтобы слайдер можно было двигать только горизонтально и при этом движение должно быть ограничено пределами слайдера.

---

## Кексобукинг

---

В этом задании мы закончим работу с перемещением главного маркера по карте.

Теперь, когда вы знакомы с тем, как работает механизм перетаскивания элементов, вы можете закончить работу над перемещением маркера `.map_pin--main`.

Вам нужно описать полный цикл Drag n Drop для маркера, то есть добавить обработчики событий `mousedown`, `mousemove` и `mouseup` на маркер.

Обработчики `mousemove` и `mouseup` должны добавляться только при вызове обработчика `mousedown`.

Обработчики `mousemove` и `mouseup` должны запускать логику изменения положения маркера: в нём должны вычисляться новые координаты маркера на основании смещения, применяться через стили к элементу и записываться в поле адреса (с поправкой на то, что в адрес записываются координаты острого конца).

Учтите, что расчёт координат маркера и их запись в поле адреса должна дублироваться и в обработчике `mouseup`, потому что в некоторых случаях пользователь может нажать мышь на маркере, но никуда его не переместить.

Ещё один момент касается ограничения перетаскивания: не забудьте сделать так, чтобы маркер невозможно было переместить за пределы карты (см. [пункт 3.4](#)).

Вспомните, что в прошлом задании вы уже добавляли обработчик на событие `mouseup`, который переводил страницу в активный режим. Теперь, когда у вас есть синхронизация с координатами, вам нужно выбрать стратегию, вы можете использовать или несколько обработчиков или один обработчик со сложной логикой.

---

## #19 Учебный проект: разделяй и властвуй

### Задача

Инкапсулируйте код всех ранее созданных модулей с помощью немедленно вызывающихся функций ([IIFE](#)). В тех модулях, которые имеют внешний интерфейс, экспортируйте то, что используется снаружи в глобальную область видимости, записав эти значения в объект `window`. При импорте значений из внешних модулей точно так же следует использовать глобальную область видимости `window`.

Свяжите полученные модули через файл `index.html` подключив необходимые файлы модулей в конце страницы при помощи тега `<script></script>` в правильном порядке, так чтобы родительские модули объявлялись раньше дочерних и порядок загрузки сохранялся.

Обратите внимание, что не все функции должны что-то экспортировать в глобальную область видимости. Некоторые функции могут просто выполняться и ничего не возвращать

## #20 Личный проект: модуляция

### Задача

Инкапсулируйте код всех ранее созданных модулей с помощью немедленно вызывающихся функций (IIFE). В тех модулях, которые имеют внешний интерфейс, экспортируйте то, что используется снаружи в глобальную область видимости, записав эти значения в объект `window`.

**Пример разбиения на модули:**

#### Кекстаграм

- `data.js` — модуль, который создаёт данные;
- `gallery.js` — модуль, который работает с галереей изображений; Использует вспомогательные модули:
  - `picture.js` — модуль для отрисовки миниатюры;
  - `preview.js` — модуль для отрисовки увеличенного изображения;
- `form.js` — модуль, который работает с формой редактирования изображения.

#### Кексобукинг

- `data.js` — модуль, который создаёт данные;
- `map.js` — модуль, который управляет карточками объявлений и пинами: добавляет на страницу нужную карточку, отрисовывает пины и осуществляет взаимодействие карточки и метки на карте;
  - `card.js` — модуль, который отвечает за создание карточки объявлений;
  - `pin.js` — модуль, который отвечает за создание пина — метки на карте;
- `form.js` — модуль, который работает с формой объявления.

### Необязательно делить на модули так же как в примере

Указанное выше разделение на модули — это только пример. Вы можете разделить код на модули так, как считаете нужным.