**Занятие 3**

Задача третьего занятия - добавить эффектов в наше портфолио:

1. Эффек для bacground первого экрана (particles.js);
2. Плавное появление элементов блока при скролле;
3. Typing-эффект для текста
4. hover-эффект для фото;
5. hover-эффекты для кнопок;
6. В футере эффекты для иконок при наведении;
7. Эффекты полей формы (псевдокласс focus);
8. Активные ссылки меню (якоря) + вид меню;
9. Filter
10. Как сделать адаптивное меню слайдеры в React

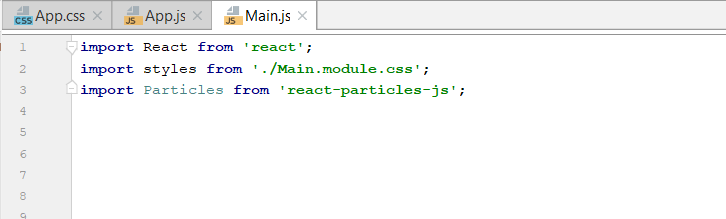
1. **Эффект для bacground первого экрана (particles.js).**

Как подключить и использовать Particles в проектах на TypeScript см. в этом [видео](https://www.youtube.com/watch?v=u8y2vCkZubk&t=347s).

Инструкция для проектов НЕ использующих TypeScript 👇 (принцип одинаковый)

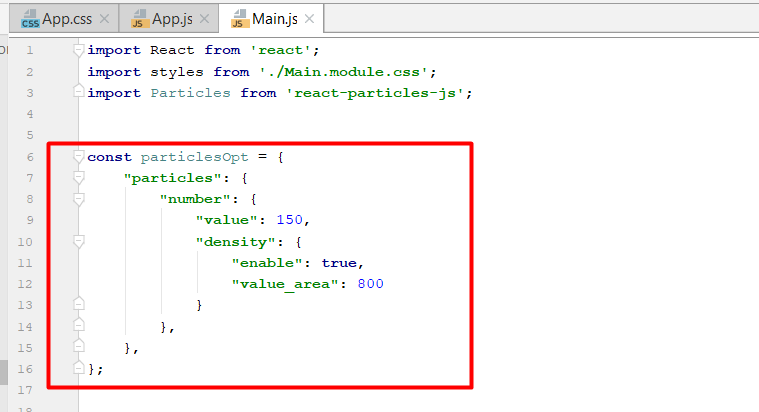
1. Устанавливаем npm-пакет [react-particles-js](https://www.npmjs.com/package/react-particles-js) (react-tsparticles обновилась, нужно ставить

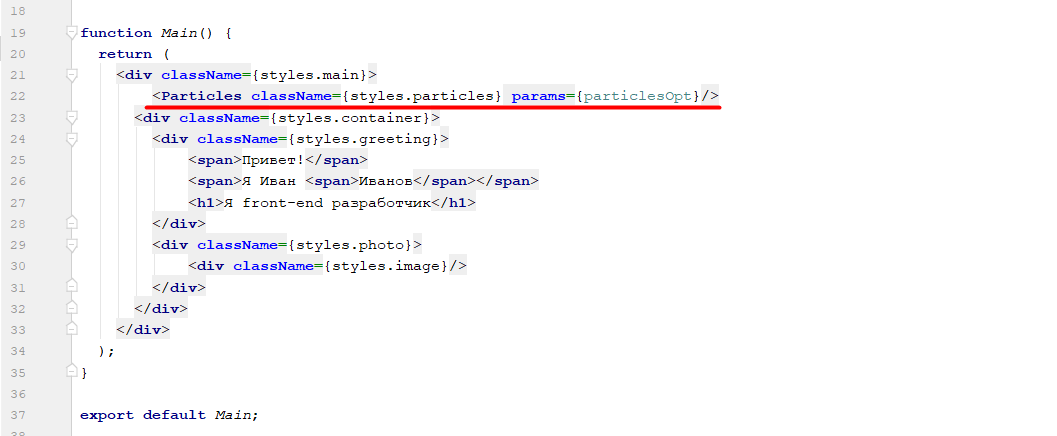
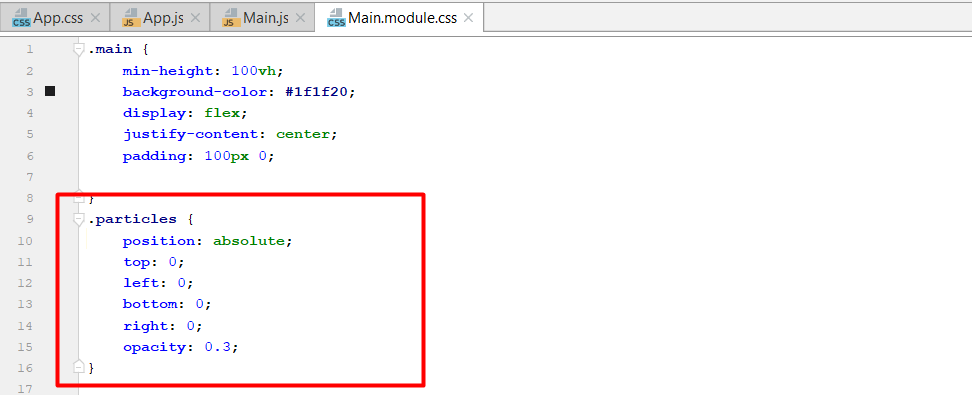
версию 1.42.1)

1. Импортируем его в компонент Main
2. Создадим переменную, в которой настроим наш particle.js (стандартные минимальные настройки).

Их не придумываем сами :). Полный список можно найти в документации, а эти я нашла здесь

<https://www.rockyourcode.com/basic-particle-js-setup-for-react>



1. И вставим компонент particle в разметку, задав ему класс и передав нашу переменную с настройками
2. Для класса particles создадим стили 

Ссылки по теме:

<https://www.youtube.com/watch?v=MD5Uy0GJA7U&t=365s>

<https://www.npmjs.com/package/react-particles-js>

1. **Плавное появление элементов блока при скролле**

Мы можем написать анимацию для плавного появления, но она сработает сразу после перезагрузки страницы. А как сделать так, чтобы анимация сработала именно в тот момент времени, когда пользователь долистает до определенного блока?.. В верстке без использования фреймворков для создания такого эффекта существует специальная библиотека wow.js (ну или можно самим написать скрипт). Подключается просто, через CDN, как jquery. Но у нас react...В react есть другая библиотека - react-reveal, которая быстро и легко решает эту задачу.

Для проектов на TypeScript нужно будет проимпортировать [эту библиотеку](https://www.npmjs.com/package/react-awesome-reveal)

Инструкция для проектов НЕ использующих TypeScript 👇 (принцип одинаковый)

1. Установим в проект пакет из npm [react-reveal](https://www.npmjs.com/package/react-reveal).

- Примеры анимаций react-reveal <https://www.react-reveal.com/examples/common/lightspeed/>.

- Статья про библиотеки для анимаций в react <https://habr.com/ru/company/skillbox/blog/456972/>.

1. Импортируем его в компонент, в котором собираемся использовать эффект. Но будьте внимательны, импортировать нужно именно те эффекты, которые будем использовать.

В нашем случае Fade (или на ваш вкус).

1. Обернем тегом <**Fade**></**Fade**> компонент, который хотим анимировать и передадим ему атрибут, который будет указывать на тип анимации: 

Больше анимаций для кастомных эффектов <https://www.npmjs.com/package/react-animated-css>

Как потом их юзать <https://www.react-reveal.com/examples/common/custom/>

1. **Typing-эффект для текста**

Для этого понадобится библиотека [react-typing-effect](https://www.npmjs.com/package/react-typing-effect). После установки нужно просто проимпортировать ее в компоненту и использовать, передав ей строку или массив строк.

Для проектов на TypeScript нужно будет проимпортировать библиотеку [отсюда](https://www.npmjs.com/package/@types/react-typing-effect).

1. **Hover-эффект для фото**

1-й способ (простой):

1. Добавим классу .photo при наведении на него мыши (:hover) свойство transform: scale(x, y). Про свойство transform можно почитать здесь <https://html5book.ru/css3-transform/>
2. И чтобы наша фотография увеличивалась плавно добавим классу .photo свойство transition: 0.5s

Про transition можно почитать здесь <https://developer.mozilla.org/ru/docs/Web/CSS/transition>

2-й способ (3D):

С помощью библиотеки <https://www.npmjs.com/package/react-tilt>

1. **Hover-эффекты для кнопок**
2. Напишем поведение для псевдоэлемента. Т.е. как он должен поменяться когда на кнопку наведут мышкой. Тут нужно быть внимательными...наводим на кнопку - меняется псевдоэлемент. В стилях такая запись будет выглядеть так: .btnShow:hover:after {}. Постарайтесь это осознать...ну или запомните =)
3. **В футере эффекты для иконок при наведении**
4. Все тот же псевдокласс :hover...а какое поведение будет, зависит от вашего макета =)
5. **Эффекты полей формы**

Чтобы добавить стили для инпутов, в которые установлен курсор, существует специальный псевдокласс :focus. По аналогии с :hover. Попробуйте его применить.

1. **Активные ссылки меню (якоря) + вид меню**
2. Зададим id каждому блоку (всем кроме main)
3. В хедере для ссылок пропишем href на эти блоки: href=”#id-блока”
4. Зададим плавную прокрутку. В элементе, в котором происходит перемещение по ссылкам, пропишем свойство scroll-behavior: smooth.

**8.1 Плавный скролл для react:**

1. npm i -S react-scroll. Для проектов на TypeScript нужно дополнительно установить типы [отсюда](https://www.npmjs.com/package/@types/react-scroll).
2. import { Link, animateScroll as scroll } from "react-scroll";
3. <Link to="id-блока">

**9. Filter**

Настолько просто, что даже не буду описывать как. Вы легко это сделаете самостоятельно!

Про фильтры можно почитать здесь <https://html5book.ru/css3-filtry/>

**10. Как сделать адаптивное меню и слайдеры в React**

<https://abcinblog.blogspot.com/2019/05/react-i.html>