«Создание Parallax эффекта на сайте при движении мыши»

Параллакс-скроллинг — техника в веб-дизайне, при которой пользователям кажется, что элементы на экране движутся с разной скоростью, создавая эффект 3D.

В день 1 вы ознакомились с функцией calc для css.

Замечания по работе 1 дня. ОБЯЗЯТЕЛЬНО К ПРОЧТЕНИЮ.

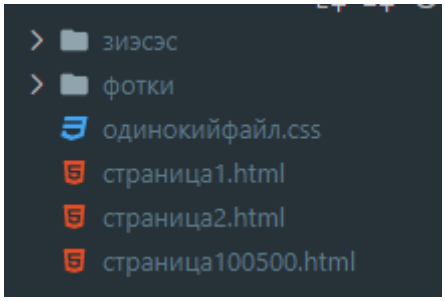
1. Структура проекта.

Чтобы не путаться самим и не путать преподавателя, создавайте четкую и понятную структуру.

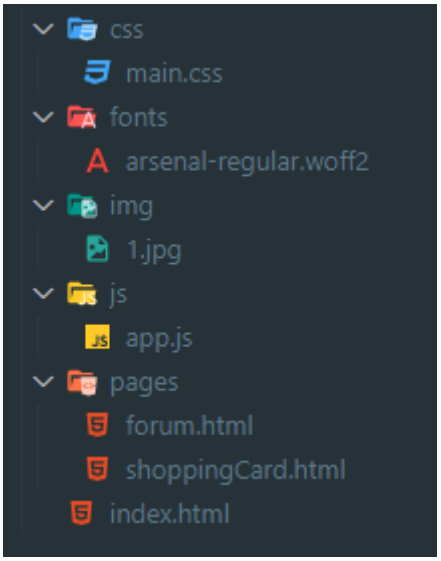
Вот некоторые советы:

- главная страница вашего сайта должна называться **«index.html»**;
- каталоги (папки) называются на английском языке **(img, css, pages, fonts)**;
- основной файл стилей называется **«style.css»** или же **«main.css»**;
- **index.html** находится в папке проекта, все остальные файлы по своим папкам
- допускается не помещать файл style.css в папку css (если у вас 1 css файл)

Как не надо делать:



Как нужно делать:



2. Структура-HTML.

Называйте классы так, чтобы не только вы, но и другой человек понимал, что вы имеете ввиду. Ни ds, ни kfk, старайтесь определять назначение блока html и давать ему соответствующее название. Блок хранит посты? Пусть будет «posts-list», «feed-rows», а сами посты «post» .

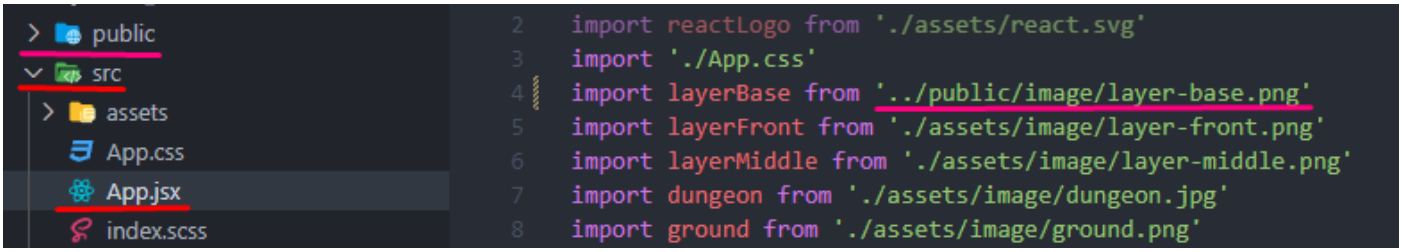
3. Ссылки.

Не стоит использовать абсолютные ссылки вида «DISK://papka/file.png», вместо этого используйте относительные ссылки.

Как?

Объясню на примере, мне нужно обратиться из файла App.jsx (папка src) в папку assets, там внутри есть папка image, а в ней нужное мне изображение. Для этого указываем начало ссылки как «./» - это значит «текущая папка» и далее уже путь к файлу.

В тоже время если нужно обратиться в папку которая не находится в том же каталоге что и файл, используем «../», как бы говоря «выйди из папки на 1 уровень выше».



Задание 0: Делаем историю проекта

Создали проект – делаем коммит. Написали html – делаем коммит. Не обязательно после каждого коммита делать «git push» (отправку на github.com), все ваши коммиты хранятся локально и никуда не пропадут. В конце у вас в репозитории на github.com должно быть как минимум 3 коммита.

Задание 1: Написать HTML-разметку

Ресурсы для текущего занятия будут размещены на курсе.

В самом HTML-файле вам нужно:

1. Самостоятельно определить структуру HTML-файла основываясь на демонстрацию страницы преподавателем.\
2. Следующие строчки должны быть записаны в HTML-файле

```
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

3. Создаем файл css и подключаем его к нашему HTML-файлу

На выходе получаем готовую HTML-страницу.

Задание 2: Написание CSS-файла

1. Создайте корневой элемент (:root), создайте 4 переменных (узнайте, как это сделать!), с первыми двумя вы должны поработать в коде. Подумайте, как это сделать.
   - В одной укажите цвет #141414
   - Во второй укажите calc(1vw + 1vh)
   - В третей 1.25s cubic-bezier(.2, .5, 0, 1);
   - В четвертой 2s cubic-bezier(.05, .5, 0, 1);

Для третьей и четвертой переменной строчки приведены ниже куда и зачем это нужно:

Кубическая функция ослабления Безье (cubic-bezier) — это тип функции ослабления, определяемый четырьмя действительными числами, которые определяют две контрольные точки, P1 и P2, кубической кривой Безье, конечные точки P0 и P3 которой фиксированы в (0, 0) и (1, 1) соответственно. Координаты x P1 и P2 ограничены диапазоном [0, 1].

То есть, это нам нужно для создания красивой, плавной анимации по кривой линии. Итак, для третьей переменной нужно:

- В «.magic-list__header» указать следующую строку: «transition: transform var(--transition);»
- В «.magic-list__item::before» (подумайте для чего это и что еще нужно туда дописать) – «transition: transform var(--transition);»

Для четвертой переменной следующее:

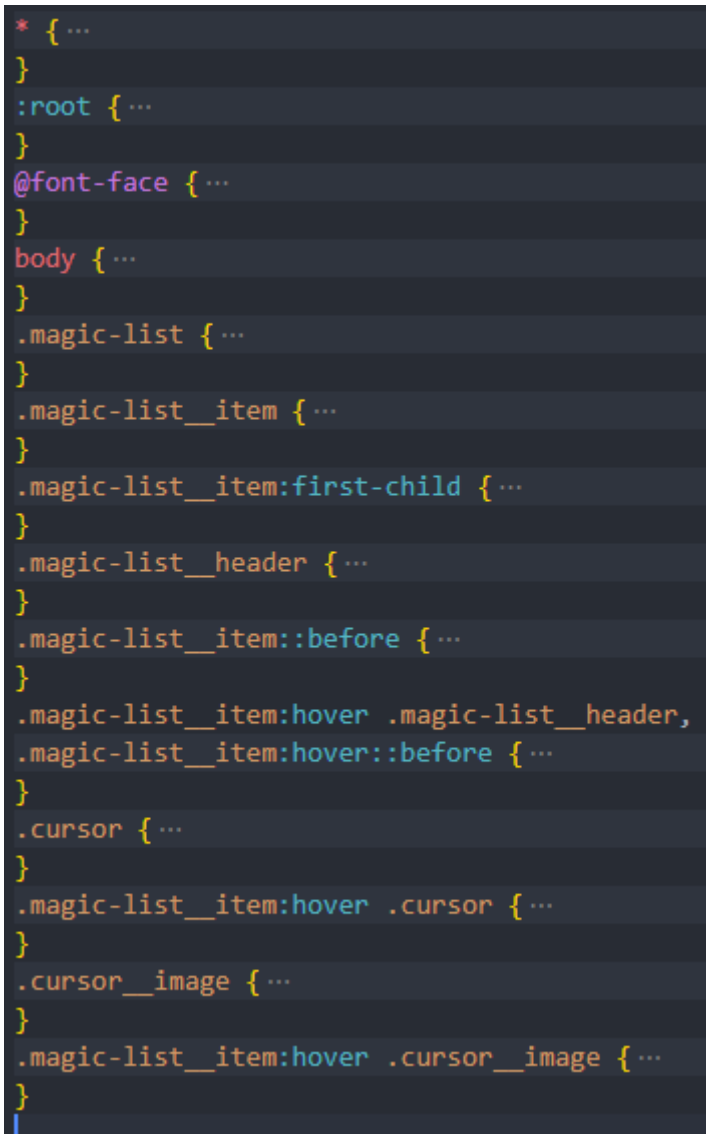
- В «.cursor» указать следующее: «transition: transform var(--transition-cursor), height 1s ease;»
- В «.cursor__image» – «transition: transform var(--transition-cursor);»

2. Подключите шрифт, расположенный в материалах
3. Используя calc() задайте всем существующим классам значения, к примеру, такие как высота, ширина и другое. Большая часть размеров должна задаваться при помощи calc()
4. Укажите нужный шрифт.

5. Поработайте с внешним видом сайта, нужно добиться максимальной схожести с образцом.

6. Не забудьте подключить js-файл из ресурсов!

Маленькая подсказка! общий вид css файла должен содержать следующие элементы:



Критерии оценивания:

1. Соблюдены советы по созданию структуры проекта
2. Функция calc() активно используется.
3. Ваш сайт схож с макетом-образцом.
4. Функционал, ожидаемой от работы, сделан.
5. Проект залит на github (методичка по работе с GitCMD лежит в первом дне)

УВАЖАЕМАЯ ПЕРВАЯ ПОДГРУППА, ЕСЛИ ВАМ ТЯЖЕЛО РАБОТАТЬ НА КОМПЬЮТЕРАХ В АУДИТОРИИ 1416 ПРОШУ ЗАХОДИТЬ НА УДАЛЕНКУ!!!