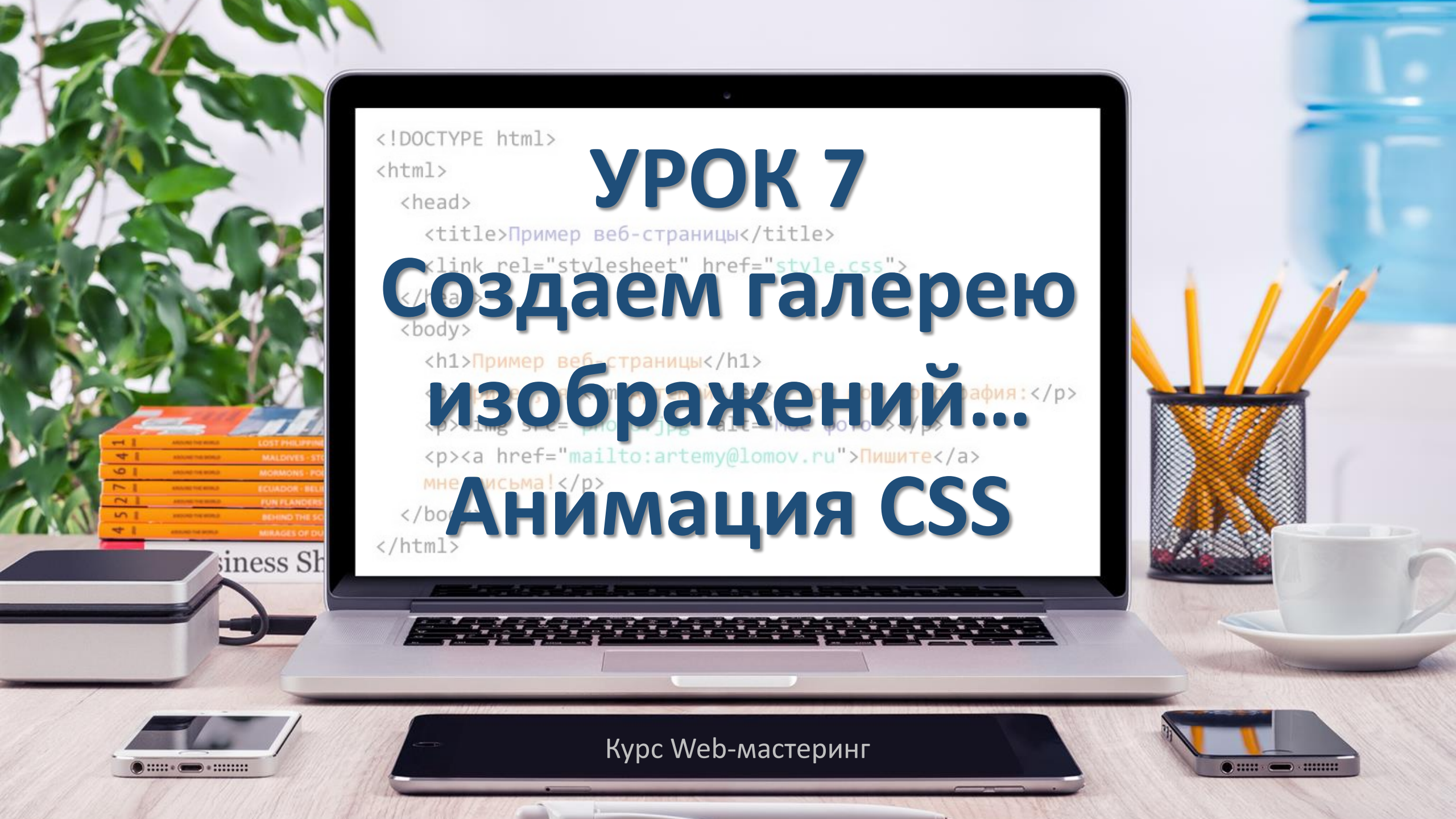
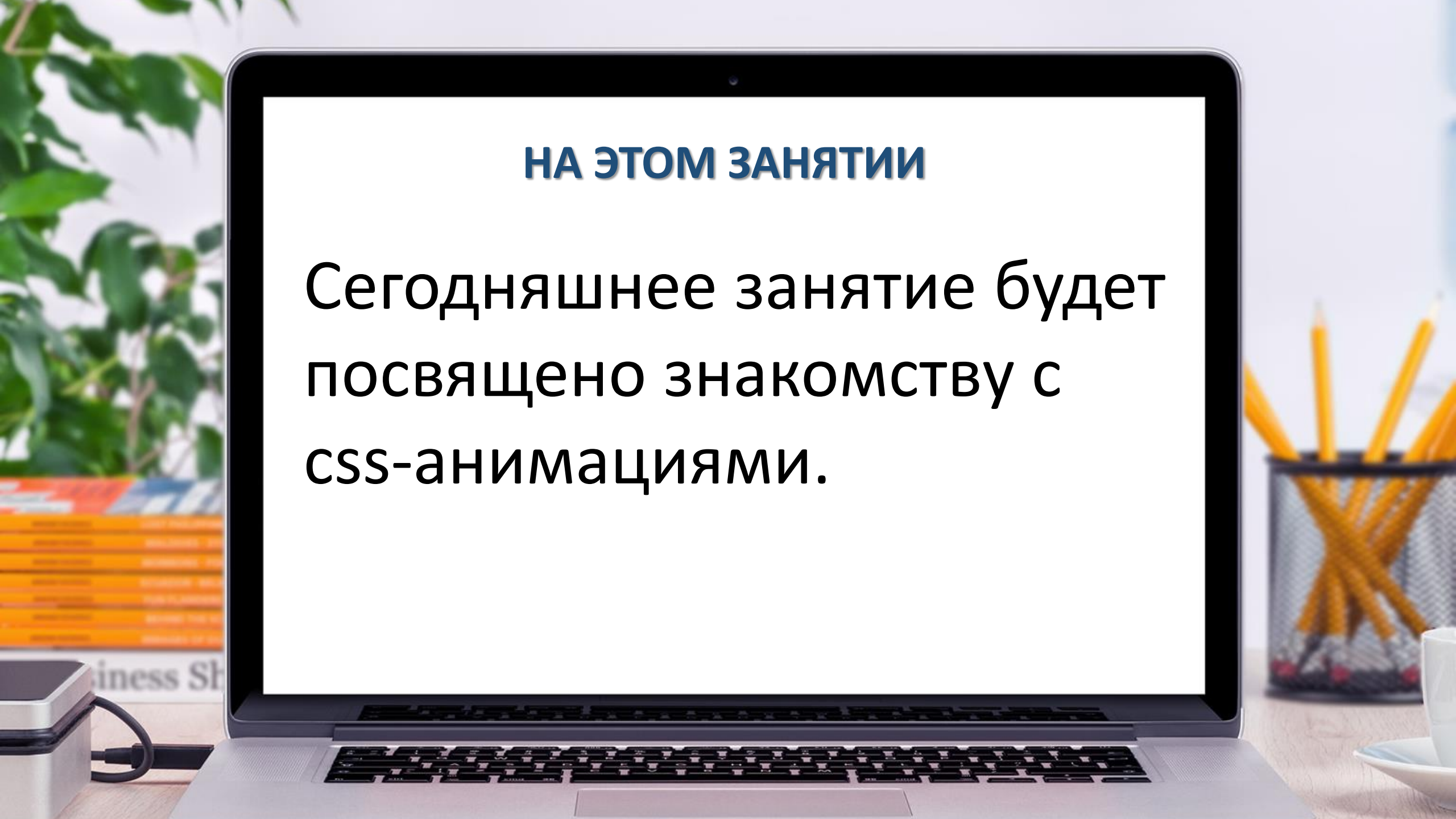


УРОК 7
Создаем галерею изображений… Анимация CSS

Курс Web-мастеринг


НА ЭТОМ ЗАНЯТИИ
Сегодняшнее занятие будет посвящено знакомству с css-анимациями.

Анимация CSS: 1 галерея

- Итак, начнем с анимации при наведении. За этот тип анимации отвечает псевдо-класс hover. Он автоматически добавляется к элементу при наведении на него мышкой.

Анимация CSS: 1 галерея

- Добавим псевдо-класс к нашим изображениям.

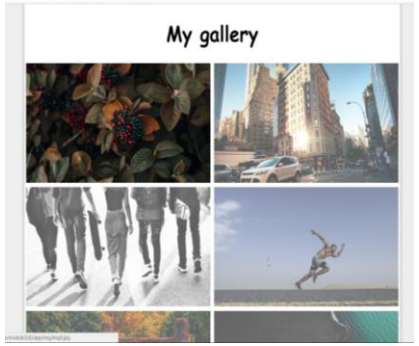
```
34
35  }
36
37  .img_link:hover {
38      opacity: 1;
39  }
40
```

Свойство **opacity** отвечает за непрозрачность элемента. 1 – элемент полностью непрозрачный, 0 – элемент прозрачный. Добавив этот код, мы не увидим никаких изменений на странице, потому что по умолчанию свойство opacity имеет значение.

Анимация CSS: 1 галерея

- Изменим значение **opacity** на 0.6 у селектора .img_link

```
48  .img_link {
49      display: block;
50      opacity: 0.65;
51  }
```



Анимация CSS: 1 галерея

Теперь добавим плавности нашей анимации. За плавность отвечает свойство **transition**.

```
47
48  .img_link {
49      display: block;
50      opacity: 0.65;
51      transition: 0.4s;
52  }
```

Анимация CSS: 1 галерея

С помощью transition мы установили время анимации. Но это свойство может принимать и другие параметры, например, имя, анимированного свойства, функция плавности перехода.

То есть свойство opacity будет анимировать **0.4s** по линейной траектории. Изменим linear на ease.

```
48  .img_link {
49      display: block;
50      opacity: 0.65;
51      transition: opacity 0.4s linear;
52  }
```

Анимация CSS: 1 галерея

Изменим linear на ease.

```
48  .img_link {
49      display: block;
50      opacity: 0.65;
51      transition: opacity 0.4s ease;
52  }
```

Теперь изменение свойства opacity будет происходит по параболе.
Также можно использовать значения ease-in, ease-out, ease-in-out.
Предлагаю поэкспериментировать и выбрать наиболее подходящий.

Анимация CSS: 1 галерея

Но анимировать можно не только непрозрачность.

transform очень универсальное свойство. Оно умеет вращать, увеличивать, скашивать, отдалять. Значение **scale** отвечает за масштабирование элемента. С его помощь изображение увеличивается на 20% при наведении.

```
52  .img_link:hover {
53      opacity: 1;
54      transform: scale(1.2);
55  }
```

Анимация CSS: 1 галерея

Добавим еще одно свойство. С помощью свойства **z-index** мы получили перекрытие остальных изображений. z-index отвечает за перспективу. Чем больше z-index, тем ближе к нам элемент.

```
52  .img_link:hover {
53      opacity: 1;
54      transform: scale(1.2);
55      z-index: 10;
56  }
```

Анимация CSS: 1 галерея

Изменим свойство transition для анимирования transform.

```
52  .img_link:hover {
53      opacity: 1;
54      transform: scale(1.2);
55      z-index: 10;
56  }
```

Анимация CSS: 1 галерея

Добавим тень при наведении.

```
52  .img_link:hover {
53      opacity: 1;
54      transform: scale(1.2);
55      z-index: 10;
56      box-shadow: 0 0 200px 30px rgba(0,0,0,0.4);
57  }
```

Анимация CSS: 1 галерея

Уберем отступ у изображения…

```
23
24  .item {
25      max-height: 265px;
26      max-width: 490px;
27      width: calc(90vw - 10px);
28      height: calc(95vw / 2 / 16 * 9);
29      background-color: dodgerblue;
30      /*margin: 5px; */
31      background-position: center;
32      -webkit-background-size: cover;
33      background-size: cover;
34
35  }
36
```

Анимация CSS: 1 галерея

…и добавляем его у ссылки.

```
45
46  .img_link {
47      display: block;
48      opacity: 0.65;
49      margin: 5px;
50      transition: opacity 0.4s ease, transform 0.2s ease-in;
51  }
52
53  .img_link:hover {
54      opacity: 1;
55      transform: scale(1.2);
56      z-index: 10;
57      box-shadow: 0 0 200px 30px rgba(0,0,0,0.4);
58  }
```

Анимация CSS: 1 галерея

Добавляем значение к transition.

```
45
46  .img_link {
47      display: block;
48      opacity: 0.65;
49      margin: 5px;
50      transition: opacity 0.4s ease, transform 0.2s ease-in, box-shadow 0.3s ease-out;
51  }
52
53  .img_link:hover {
54      opacity: 1;
55      transform: scale(1.2);
56      z-index: 10;
57      box-shadow: 0 0 200px 30px rgba(0,0,0,0.4);
58  }
```

Анимация CSS: 1 галерея

Добавляем поворот изображения при наведении.

```
45
46  .img_link {
47      display: block;
48      opacity: 0.65;
49      margin: 5px;
50      transition: opacity 0.4s ease, transform 0.2s ease-in, box-shadow 0.3s ease-out;
51  }
52
53  .img_link:hover {
54      opacity: 1;
55      transform: scale(1.2) rotate(5deg);
56      z-index: 10;
57      box-shadow: 0 0 200px 30px rgba(0,0,0,0.4);
58  }
```

Анимация CSS: 2 галерея

Теперь сделаем тоже самое для второй галереи.

```
14  .image-wrapper {
15      display: block;
16      opacity: 0.8;
17      transition: 0.4s;
18  }
19
20  .image-wrapper + .image-wrapper {
21      margin: 3px 0;
22  }
23
24  .image-wrapper:hover {
25      opacity: 1;
26  }
```

Анимация CSS: 2 галерея

К сожалению, в этой сетке мы не можем воспользоваться свойством transform. Но есть и другие способы добавить интерактивности нашей галереи.

```
13
14 .image-wrapper {
15     display: block;
16     opacity: 0.8;
17     transition: 0.4s;
18     filter: blur(1px);
19 }
20
21 .image-wrapper + .image-wrapper {
22     margin: 3px 0;
23 }
24
25 .image-wrapper:hover {
26     opacity: 1;
27 }
```

Анимация CSS: 2 галерея

К сожалению, в этой сетке мы не можем воспользоваться свойством transform. Но есть и другие способы добавить интерактивности нашей галереи.

Добавим свойство filter и немного заблюрим наши изображения.

```
13
14 .image-wrapper {
15     display: block;
16     opacity: 0.8;
17     transition: 0.4s;
18     filter: blur(1px);
19 }
20
21 .image-wrapper + .image-wrapper {
22     margin: 3px 0;
23 }
24
25 .image-wrapper:hover {
26     opacity: 1;
27 }
```

Анимация CSS: 2 галерея

Добавим свойство filter и немного заблюрим наши изображения.

```
14  .image-wrapper {
15      display: block;
16      opacity: 0.8;
17      transition: 0.4s;
18      filter: blur(1px);
19  }
20
21  .image-wrapper + .image-wrapper {
22      margin: 3px 0;
23  }
24
25  .image-wrapper:hover {
26      opacity: 1;
27      filter: blur(0);
28  }
```

Анимация CSS: 2 галерея

Свойство filter имеет множество интересных значений.

```
24
25  .image-wrapper:hover {
26      opacity: 1;
27      filter: sepia(1.3);
28  }
```

```
24
25  .image-wrapper:hover {
26      opacity: 1;
27      /*filter: sepia(1.3);*/
28      filter: brightness(1.7);
29  }
```

Анимация CSS: 2 галерея

Свойство filter имеет множество интересных значений.

```
24
25  .image-wrapper:hover {
26      opacity: 1;
27      /*filter: sepia(1.3);*/
28      /*filter: brightness(1.7);*/
29      filter: grayscale(10);
30  }
```

```
25  .image-wrapper:hover {
26      opacity: 1;
27      /*filter: sepia(1.3);*/
28      /*filter: brightness(1.7);*/
29      /*filter: grayscale(10);*/
30      filter: hue-rotate(45deg);
31  }
```

Значения свойства filter в css

`blur()`

Значение задается в единицах длины, например `px`, `em`. Применяет размытие по Гауссу к исходному изображению. Чем больше значение радиуса, тем больше размытие. Если значение радиуса не задано, по умолчанию берется `0`.

Синтаксис

```
filter: blur(3px);
```

`brightness()`

Значение задается в `%` или в десятичных дробях. Изменяет яркость изображения. Чем больше значение, тем ярче изображение. Значение по умолчанию `1`.

Синтаксис

```
filter: brightness(50%);
filter: brightness(.5);
```

Значения свойства filter в css

`contrast()`

Значение задается в `%` или в десятичных дробях. Регулирует контрастность изображения, т.е. разницу между самыми темными и самыми светлыми участками изображения/фона. Значение по умолчанию `100%`. Нулевое значение скроет исходное изображение под темно-серым фоном. Значения, увеличивающиеся от `0` до `100%` или от `0` до `1`, будут постепенно открывать исходное изображение до оригинального отображения, а значения свыше будут увеличивать контраст между светлыми и темными участками.

Синтаксис

```
filter: contrast(20%);
filter: contrast(.2);
```

Значения свойства filter в css

`drop-shadow()`

Фильтр действует подобно свойствам `box-shadow` и `text-shadow`. Использует следующие значения: смещение по оси X смещение по оси Y размытость растяжение цвет тени. Отличительная особенность фильтра заключается в том, что тень добавляется к элементам и его содержимому с учетом их прозрачности, т.е. если элемент содержит текст внутри, то фильтр добавит тень одновременно для текста и видимых границ блока. В отличие от других фильтров, для этого фильтра обязательно задание параметров (минимальное - величина смещения).

Синтаксис

```
filter: drop-shadow(2px 3px 5px black);
```

Значения свойства filter в css

`grayscale()`

Извлекает все цвета из картинки, делая на выходе черно-белое изображение. Значение задается в `%` или десятичных дробях. Чем больше значение, тем сильнее эффект.

Синтаксис

```
filter: grayscale(.5);
filter: grayscale(50%);
```

`hue-rotate()`

Меняет цвета изображения в зависимости от заданного угла поворота в цветовом круге. Значение задается в градусах от `0deg` до `360deg`. `0deg` - значение по умолчанию, означает отсутствие эффекта.

Синтаксис

```
filter: hue-rotate(180deg);
```

Значения свойства filter в css

`invert()`

Фильтр делает негатив изображения. Значение задается в `%`. `0%` не применяет фильтр, `100%` полностью преобразует цвета.

Синтаксис

```
filter: invert(100%);
```

`opacity()`

Фильтр работает аналогично со свойством `opacity`, добавляя прозрачность элементу. Отличительная особенность - браузеры обеспечивают аппаратное ускорение для фильтра, что позволяет повысить производительность. Дополнительный бонус - фильтр можно одновременно сочетать с другими фильтрами, создавая при этом интересные эффекты. Значение задается только в `%`, `0%` делает элемент полностью прозрачным, а `100%` не оказывает никакого эффекта.

Синтаксис

```
filter: opacity(30%);
```

Значения свойства filter в css

`saturate()`

Управляет насыщенностью цветов, работая по принципу контрастного фильтра. Значение `0%` убирает цветность, а `100%` не оказывает никакого эффекта. Значения от `0%` до `100%` уменьшают насыщенность цвета, выше `100%` - увеличивают насыщенность цвета. Значение может задаваться как в `%`, так и целым числом, `1` эквивалентно `100%`.

Синтаксис

```
filter: saturate(300%);
```

`sepia()`

Эффект, имитирующий старину и «ретро». Значение `0%` не изменяет внешний вид элемента, а `100%` полностью воспроизводит эффект сепии.

Синтаксис

```
filter: sepia(150%);
```

Значения свойства filter в css

`url()`

Функция принимает расположение внешнего XML-файла с svg-фильтром, или якорь к фильтру, находящемся в текущем документе.

Синтаксис

```
filter: url(#filterId); /* если фильтр находится в этом документе */
filter: url(filter.svg#filterId); /* если фильтр с id="filterId" находится в файле filter.svg*/
```

`none`

Значение по умолчанию. Означает отсутствие эффекта.

`initial`

Устанавливает это свойство в значение по умолчанию.

`inherit`

Наследует значение свойства от родительского элемента.

ДОМАШНЕЕ ЗАДАНИЕ

- Поэкспериментируйте дома с различными фильтрами и их значениями.
- Примеры работы фильтров вы найдете тут https://html5book.ru/css3-filtry/

=========**ДЛЯ ПРОДВИНУТЫХ**==========

- Сделать 3 отдельные фотогалереи на одной странице с использованием разных фильтров.

Презентации предыдущих уроков:

https://muzeinauki.ru/kursy-html/


</HTML>