

# Яркость и контрастность, `brightness` и `contrast`

CSS-фильтры позволяют применить визуальные эффекты к элементам страницы. Фильтры обычно применяют к изображениям, фонам или рамкам, а иногда и к другим элементам.

Фильтры применяются к элементу до того, как он будет отображён, поэтому количество фильтров влияет на скорость загрузки страницы.

Фильтры задаются с помощью свойства `filter`. Начнём знакомство с ними.

Яркость — `brightness`. Допустимы числовые значения от `0` и выше или проценты. Со значением фильтра `1` или `100%` элемент отображается без изменений.

```
.effect {  
  
  filter: brightness(2); /* увеличивает яркость вдвое */  
  
  filter: brightness(10%); /* снижает яркость до 10% */  
  
  filter: brightness(150%); /* увеличивает яркость на 50% */  
  
}
```

Контрастность — `contrast`. Как и у фильтра яркости в качестве значения можно использовать числа или проценты.

```
.effect {  
  
  filter: contrast(0.5); /* снижает контрастность наполовину */  
  
  filter: contrast(50%); /* тоже снижает контрастность до 50% */  
  
}
```

## Бесцветность и сепия, `grayscale` и `sepia`

Следующие два фильтра исторически относятся к фотографии: изначально фотографии были чёрно-белые или коричневатого оттенка краски сепии.

С помощью фильтров бесцветности и сепии фотографиям можно придать эффект «старины».

Бесцветность — `grayscale`. Допустимы числовые значения от `0` до `1` и процентные значения от `0%` до `100%`. При значениях `0` или `0%` элемент отображается без изменений.

```
.effect {  
  
  filter: grayscale(1); /* полное обесцвечивание */  
  
  filter: grayscale(50%); /* обесцвечивание наполовину */  
  
}
```

Сепия — `sepia`. Допустимые значения такие же, как у `grayscale`.

```
.effect {  
  
  filter: sepia(1); /* полный эффект сепии */  
  
  filter: sepia(25%); /* эффект сепии применяется на 25% */  
  
}
```

## Инверсия и насыщенность, `invert` и `saturate`

Ещё два фильтра для цветовых эффектов — инверсия цвета и цветовая насыщенность. Как понятно из названия, при инверсии цвета меняются на противоположные, и мы получаем «негатив» картинки, а цветовая насыщенность делает краски ярче и противоположна по смыслу фильтру обесцвечивания.

Инверсия цвета — `invert`. Допустимые простые числовые значения от `0` до `1` и процентные значения от `0%` до `100%`.

```
.effect {  
  
  filter: invert(1); /* полная инверсия цвета */  
  
  filter: invert(50%); /* инверсия цвета наполовину, все цвета серые */  
  
}
```

Цветовая насыщенность — `saturate`. Допустимые простые числовые значения от `0` и выше и процентные значения от `0%` и выше. При значении фильтра `1` или `100%` элемент отображается без изменений. Если установить нулевое значение, то картинка обесцветится, как при использовании фильтра `grayscale`.

```
.effect {  
  
  filter: saturate(2); /* цвета ярче вдвое */  
  
  filter: saturate(50%); /* цвета вдвое бледнее */  
  
  filter: saturate(0); /* обесцвеченная картинка */  
  
}
```

## Поворот цвета, hue-rotate



Ещё один интересный цветовой фильтр — поворот цвета `hue-rotate`. Если представить все цвета в виде круга, то этим фильтром можно повернуть цвет картинки вдоль круга на определённый угол.

Кстати, похожую функцию цветового поворота мы рассматривали в [части по препроцессору Less](#).

Итак, поворот цвета — `hue-rotate`. Значение задаётся в градусах от `0deg` до `360deg`. Допускается также использование отрицательного угла. Положительное значение будет вращать цвет по часовой стрелке, а отрицательное — против. При значении фильтра `0deg` элемент отображается без изменений.

Примеры:

```
.effect {  
  
  filter: hue-rotate(90deg); /* поворот на 90 градусов по часовой */  
  
  filter: hue-rotate(-90deg); /* поворот на 90 градусов обратно */  
  
}
```

## Размытость и непрозрачность, blur и opacity

Не все фильтры работают с цветом. Например, фильтр размытости создаёт у фотографии эффект расфокусировки, а фильтр непрозрачности может сделать объект частично прозрачным.

Фильтр `blur` задаёт размытость объекта. Его значение задаётся в пикселях от `0px` и выше.

```
.effect {  
  
  filter: blur(5px); /* размытость в 5px */  
  
}
```

Фильтр `opacity` задаёт уровень *непрозрачности* объекта. Работает точно так же, как свойство `opacity`. Допустимые значения от `0` до `1` или от `0%` до `100%`.

```
.effect {  
  
  filter: opacity(0.5); /* объект наполовину непрозрачный */  
  
  filter: opacity(0%); /* полностью прозрачный объект */  
  
}
```

Между фильтром и свойством `opacity` всё же есть «невидимое» отличие. Некоторые браузеры используют аппаратное ускорение для отрисовки фильтров, а для свойства `opacity` ускорение не используется. Поэтому применение фильтра `opacity` может существенно улучшить производительность по сравнению с аналогичным свойством.

## Тень, drop-shadow

С помощью фильтра `drop-shadow` можно добавить тень к объекту. Синтаксис фильтра такой же, как и у свойства `box-shadow`.

```
.effect {  
  
  /*  
  
    чёрная тень, смещённая на 10px по горизонтали и  
  
    на 5px по вертикали, с радиусом размытия 3px  
  
  */  
  
  filter: drop-shadow(10px 5px 3px #000000);  
  
  
  /* несмещённая зелёная тень с радиусом размытия 5px */  
  
  filter: drop-shadow(0px 0px 5px green);  
  
}
```

В отличие от `box-shadow` фильтр тени не поддерживает параметр `inset` для внутренней тени.

## Применение CSS-фильтров к тексту

Фильтры можно применять без ограничений не только к картинкам, но и к другим HTML-элементам.

## Комбинация фильтров

К одному и тому же блоку можно применить несколько фильтров одновременно. Для этого все фильтры просто перечисляются через пробел после слова `filter`:

```
.effect {  
  
  filter: sepia(50%);  
  
  filter: sepia(50%) blur(5px);  
  
  filter: sepia(50%) blur(5px) opacity(50%);  
  
}
```

Стоит помнить, что перечисление фильтров в разном порядке даёт разный результат. Это так, потому что каждый следующий фильтр применяется уже к изображению с эффектом.

## Комбинация фильтров drop-shadow

Если применить несколько фильтров drop-shadow к одному блоку, то их комбинация даст интересный эффект. Так как каждый следующий фильтр в комбинации применяется к изображению с предыдущими фильтрами, то ещё одна тень у блока с тенью создаст двойную тень.

```
.effect {  
  
  /* обычная тень */  
  
  filter: drop-shadow(0px 0px 0px black);  
  
  
  
  /* двойная тень */  
  
  filter: drop-shadow(0px 0px 0px black) drop-shadow(0px 0px 0px black);  
  
}
```

В этом ещё одно отличие фильтра drop-shadow от свойства box-shadow.

Кстати, фильтр drop-shadow бывает полезно применять не только к изображениям, но и к обычным элементам. Тень будет, как и в случае с картинками, идти по непрозрачному контуру. В область тени включаются также псевдоэлементы блока.

## Анимация CSS-фильтров

К CSS-фильтрам можно применять анимации и плавные переходы.

Например, очень оригинальный эффект получится, если применить бесконечную анимацию к фильтру поворота цвета hue-rotate на цветной фотографии.

## Плавные переходы и CSS-фильтры

Теперь рассмотрим применение к фильтрам плавных переходов.

К примеру, можно фильтрами сделать фотографию чёрно-белой и размытой, по наведению убрать эффект фильтра, а само визуальное изменение сделать плавным.

Кстати, у свойства `filter` значение по умолчанию — `none`. Воспользуемся им для отмены эффекта фильтра по наведению на фотографию.