

# border-radius

радиус округления углов



# Возможности **border-radius**:

**border-radius: 10px;** (округляет все углы)

**border-radius: 10px 50px;** (округляет углы по диагоналям)

**border-radius: 10px 50px 30px;** (левый верхний, диагональ, правый нижний)

**border-radius: 20px 7px 15px 2px;** (левый верхний и по часовой)

**border-radius: 50%/30%;** (округление угла по горизонтали и вертикали)

**border-radius: 50%;** (окружность, если применяется для квадрата)



# box-shadow

радиус округления углов



# Возможности **box-shadow**:

Задаёт тень для блока

**box-shadow: -1px 1px 3px -4px rgba(0, 0, 0, .5) inset;**

- первое значение – смещение по вертикали
- второе значение – смещение по горизонтали
- третье значение – степень размытия тени
- четвёртое значение – степень сжатия тени
- пятое значение – цвет тени
- шестое – указывает на внутреннюю тень

\* значения 4 и 6 не обязательны! Тогда сжатие тени будет равно 0 и она будет внешней

\*\* можно задавать несколько теней для блока (через запятую)





# box-sizing

радиус округления углов



# Возможности **box-sizing**:

Изменяет алгоритм расчёта высоты и ширины элемента

**box-sizing: content-box;**

(значение по умолчанию, поля и границы влияют на размер элемента)

**box-sizing: border-box;** (включает в размеры элемента поля и границы)



# background-size

радиус округления углов



# Возможности background-size:

Управление размерами фона (background-image)

**background-size: 1000px;**

(выставляет ширину фона в 1000 пикселей, высота выставляется автоматически)

**background-size: 1000px 500px;**

(указывает на ширину и высоту фона соответственно)

**background-size: contain;**

(фон всегда будет отображаться в блоке полностью)

**background-size: cover;**

(фон всегда будет собой заливать весь блок)





# linear-gradient

линейный градиент

Градиент это практически та же самая картинка, которую генерирует браузер. А значит мы можем им управлять при помощи свойства `background`, как обычными рисунками.



# Возможности **linear-gradient**:

`linear-gradient(#999, #1f1f1f)` (градиент от одного цвета к другому)

`linear-gradient(180deg, #999, #1f1f1f)` (угол наклона градиента)

`linear-gradient(to left/right top/bottom , #999, #1f1f1f)`

(наклон градиента вместо углов)

`linear-gradient(135deg, #999 50%, #1f1f1f 50%)` (размеры градиента)

`background-size: 20px 20px;`

(размер рисунка градиента – по умолчанию растягивается на всю фигуру)

`linear-gradient(135deg, #999 50%, #1f1f1f 50%, transparent 53%) -200px -200px no-repeat #444;`

(прозрачный градиент, позиция, отмена повторения, фон)





# Возможности **linear-gradient**:

`linear-gradient(#ddd 1px, transparent 1px),`

`linear-gradient(90deg, #ddd 1px, transparent 1px),`

`linear-gradient(#eee 1px, transparent 1px),`

`linear-gradient(90deg, #eee 1px, transparent 1px),`

`#f5f5f5;`

`background-size: 100px 100px, 100px 100px, 10px 10px, 10px 10px;`

(градиенты с указанными размерами и заливкой могут создавать красивые фоны на сайте)



# radial-gradient

радиальный градиент



# Возможности radial-gradient:

`radial-gradient(#999, #1f1f1f)` (градиент от одного цвета к другому)

`radial-gradient(circle/ellipse, #999, #1f1f1f)` (круг и эллипс)

`radial-gradient(70px 30px, #999, #1f1f1f)` (размеры градиента)

`radial-gradient(70px at (70px 70px) / (left/right top/bottom), #999, #1f1f1f)`  
(позиция градиента)

`closest-side` (совпадает с ближайшей к нему стороной)

`closest-corner` (градиент вычисляется на основании расстояния до ближайшего угла)

`farthest-side` (совпадает с дальней к нему стороной)

`farthest-corner` (градиент вычисляется на основании расстояния до дальнего угла)

`radial-gradient(circle closest corner at 40% 10%, #999, #1f1f1f)`





# Возможности radial-gradient:

background:

```
radial-gradient(#333, #222 25%, #eee 27%, transparent 29%) 0 0,  
radial-gradient(#777, #555 25%, #eee 27%, transparent 29%) 25px 0,  
radial-gradient(#777, #555 25%, #eee 27%, transparent 29%) -25px 0,  
radial-gradient(#777, #555 25%, #eee 27%, transparent 29%) 0 25px,  
radial-gradient(#777, #555 25%, #eee 27%, transparent 29%) 0 -25px,  
#1f1f1f;
```

background-size: 100px 100px;

(множественный радиальный градиент)





# Повторяющиеся градиенты:

`repeating-linear-gradient(135deg, #999, #999 5px, #1f1f1f 5px, #1f1f1f 10px)`  
(повторяющийся линейный градиент)

`repeating-linear-gradient(135deg, #aaa, #999 5px, #777 5px, #888 10px)`  
(поигравшись с цветами получаем псевдо объём)

`repeating-radial-gradient(circle at 30% 20%, #bbb, #bbb 25px, #ddd 25px, #ddd 50px)`  
(повторяющийся радиальный градиент)





# transition

создаёт плавные эффекты переходов





# Возможности **transition**:

**transition-property** - указывает стили, к которым применяется анимация

**transition-duration** - указывает время проигрывания анимации в секундах

**transition-timing-function** - указывает тип анимации

**transition-delay** - указывает время задержки перед анимацией в секундах

**transition**: transition-property transition-duration transition-timing-function transition-delay;





# Возможности transition:

## transition-timing-function свойства:

**ease** - медленно - быстро - медленно (по умолчанию)

**linear** - действует одинаково на протяжении всей анимации

**ease-in** - медленно – быстро

**ease-out** - быстро – медленно

**ease-in-out** - медленно – медленно

**step-start** - без анимации сразу применяет конечные стили

**step-end** - без анимации, применяет стили в конце

**steps** - указывает конкретное количество шагов для анимации

**cubic-bezier** - задает анимацию по кривой Безье

<http://cubic-bezier.com/>



# animation и @keyframes

анатомия анимации



# Возможности animation:

animation-name: none – имя анимации

animation-duration: 0s – время анимации

animation-timing-function: ease – вид анимации

animation-delay: 0s – задержка перед началом анимации

animation-iteration-count: 1 – количество повторений

animation-direction: normal – направление движения анимации

animation-fill-mode: none – определение стилей после анимации

animation-play-state: running – рекламная пауза

**animation:** animation-name animation-duration animation-timing-function animationdelay  
animation-iteration-count animation-direction animation-fill-mode animation-playstate;





# Возможности @keyframes:

**from** – начало анимации

**to** – конец анимации

**проценты** – вся анимация разбивается на проценты

```
@keyframes name{  
  from{свойства}  
  to {свойства}  
}
```

```
@keyframes name{  
  0% {свойства}  
  25% {свойства}  
  50% {свойства}  
  75% {свойства}  
  100%{свойства}  
}
```





# Возможности свойств animation:

animation-iteration-count:

**цифры** - количество повторений анимации

**infinite** - бесконечная анимация





# Возможности свойств animation:

animation-direction:

**normal** - анимация начинается сначала до конца

**alternate** - анимация идёт с начала до конца и потом плавно обратно

**reverse** - анимация идет с конца в начало

**alternate-reverse** - анимация идет с конца в начало и потом плавно обратно





# Возможности свойств animation:

animation-fill-mode:

**none** - стили не применяются

**forwards** - применяются стили последнего кадра

**backwards** - применяются стили первого кадра и держится на протяжении свойства animation-delay

**both** - применяются стили и того и того кадра







# Возможности свойств animation:

animation-play-state:

`running` – пуск анимации

`paused` - рекламная пауза))





# transform

трансформации элементов



# Возможности transform:

`rotate(0deg)` – поворот в градусах

`scale(sx, sy)` – масштаб (может быть одно значение)

`scaleX(sx)` – масштаб по горизонтали

`scaleY(sy)` – масштаб по вертикали

`skewX(0deg)` – угловой наклон по вертикали

`skewY(0deg)` – угловой наклон по горизонтали

`translate(tx, ty)` – сдвиг по горизонтали и вертикали

`translateX(tx)` – сдвиг по горизонтали

`translateY(ty)` – сдвиг по вертикали





# Практика для закрепления материала:

<https://youtu.be/IEL20bF0i6U> – анимация фона

<https://youtu.be/WMu4HhAtrZI> – покадровая анимация

<https://youtu.be/W2OeiUuVAmo> – анимация кнопки

<https://youtu.be/LtydgLVmnkg> – 3D-трансформации на практике

<https://youtu.be/Baa-W3GZHms> – фильтры

<https://youtu.be/oLCgXG7E8zY> – Animate.CSS

<https://youtu.be/7gMYtFGjg3s> – Animate.CSS+Wow.JS

<https://youtu.be/eIIc41CjfxU> - анимация прелоадера на CSS3

