

Размер фона, шаг 1

По умолчанию браузер не меняет размеры фонового изображения. Как быть, если нужно задать фону другие ширину и высоту или привязать его размеры к размерам родительского блока?

Для этого существует свойство `background-size`. Оно принимает в качестве значения два аргумента: первый — это ширина фонового изображения, второй — его высота. Второй аргумент необязательный. Если высота не указывается, то она определяется автоматически в зависимости от ширины и исходных пропорций. По умолчанию оба аргумента равны `auto`.

Пример:

```
background-size: auto auto; /* исходные ширина и высота изображения */  
background-size: 100px; /* ширина 100px, высота пропорциональная */  
background-size: 100px 200px; /* ширина 100px, высота 200px */
```

Размер фона, шаг 2

Ещё два значения свойства `background-size` — это `contain` и `cover`.

Значение `contain` работает так:

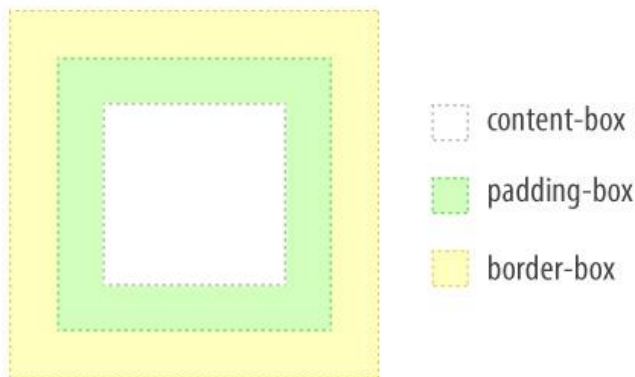
- пропорции изображения сохраняются;
- изображению задаются максимально возможные размеры, при которых оно и по ширине, и по высоте полностью помещается в границы фона;
- изображение может не закрывать всю фоновую область блока, если пропорции изображения и блока разные.

Значение `cover` работает иначе:

- пропорции изображения сохраняются;
- изображению задаются минимально возможные размеры, при которых оно закроет всю фоновую область блока;
- если пропорции изображения и блока разные, то часть изображения обрезается.

Границы фона

Ещё одно фоновое свойство — `background-origin`. Оно задаёт расположение и размеры области отображения фонового изображения и принимает три значения: `padding-box`, `border-box` и `content-box`.



Значение по умолчанию — `padding-box`. При этом значении область отображения фонового изображения соответствует внутренней области блока, не включая рамки.

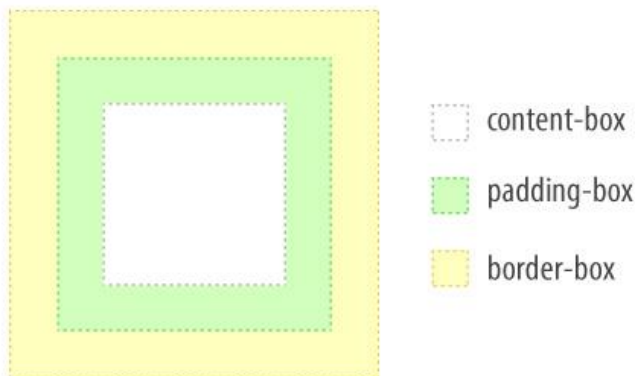
При `content-box` область отображения соответствует только области содержимого, не включая рамки и внутренние отступы.

При `border-box` область отображения соответствует полному размеру блока, включая и рамки. Фоновое изображение начинается от внешнего края блока и закрывается рамками, если они заданы.

Обрезка фона

Свойство `background-clip` управляет тем, как обрезается фон. Причём обрезаются не только фоновые изображения, но и фоновый цвет.

Значения свойства такие же, как у `background-origin`: `padding-box`, `border-box` и `content-box`.



Значение `border-box` задано по умолчанию, при этом фоновое изображение совсем не обрезается.

Значение `padding-box` обрежет фон по внутреннему краю области рамки.

Значение `content-box` обрежет фон по краю области содержимого.

Множественный фон

Можно задать блоку несколько фоновых изображений одновременно. Для этого пути к изображениям в свойстве `background-image` или `background` перечисляются через запятую:

```
background-image: url("image-1.png"), url("image-2.png");
```

При этом выше будет то изображение, которое находится раньше в списке: `image-1.png` в примере будет выше `image-2.png`.

Значения других свойств для множественных фоновых изображений тоже задаются через запятую. Порядок значений должен соответствовать порядку фоновых картинок. Например:

```
/* для первой картинки масштабирование contain, для второй — 100px */
background-size: contain, 100px;

/* для первой картинки позиционирование top, для второй — 100% */
background-position: top, 100%;
```

Позиция фона от разных сторон

Интересная возможность `background-position` — расположение фона можно задавать относительно любого угла блока, а не только от левого верхнего.

Чтобы указать от какой стороны отсчитывать расположение фона, нужно перед значением координат задать ключевые слова: `top`, `right`, `bottom` или `left`. Например:

```
/* по умолчанию координаты задаются для левого верхнего угла */
background-position: 10px 50px; /* слева 10px, сверху 50px */

background-position: right 30px bottom 60px; /* справа 30px, снизу 60px */
background-position: left 50px bottom 10px; /* слева 50px, снизу 10px */
background-position: right 40px top 30px; /* справа 40px, сверху 30px */
```

Поддержка данных значений свойства `background-position` в современных браузерах [практически полная](#).

Повторение фона: background-repeat: round

Два интересных, но малоизвестных значения привычного свойства `background-repeat`, которое задаёт повторение фона, — это `round` и `space`.

Значение свойства по умолчанию `background-repeat: repeat` просто повторяет фоновую картинку по всей ширине и высоте блока. Если части повторяющейся картинки не помещаются в ширину блока, то они просто обрезаются.

Если задать значение `background-repeat: round`, то повторяющиеся картинки по краям блока обрезать не будут, а равномерно растянутся или сожмутся по всей ширине и высоте, чтобы занять оставшееся пространство.

Кстати, `background-repeat` принимает в качестве значения два аргумента: режим повторения по горизонтали и по вертикали. Если передать один параметр, то он применится к обоим направлениям. Например:

```
/* повторение фона repeat по горизонтали и вертикали */  
  
background-repeat: repeat;  
  
  
/* повторение фона round по горизонтали и repeat по вертикали */  
  
background-repeat: round repeat;
```

Повторение фона: background-repeat: space

Режим повторения фона `background-repeat: space` похож на `round`, при нём части картинки тоже не обрезаются. Отличие в том, что повторяющиеся фоновые картинки не сжимаются или растягиваются, а для компенсации ширины блока между ними добавляется пустое пространство.

Внешняя рамка

Полезное, но редко используемое свойство — внешняя рамка или `outline`.

Синтаксис `outline` схож с `border`. Но внешней рамке нельзя задать параметры отдельных сторон.

Пример:

```
/* сплошная чёрная рамка толщиной 10px */  
  
outline: 10px solid black;
```

Это сокращённый синтаксис свойства `outline`. Есть и отдельные свойства:

```
outline-width: 10px; /* толщина */
```

```
outline-style: solid; /* стиль */  
  
outline-color: black; /* цвет */
```

Внешняя рамка всегда отображается снаружи элемента, не влияет на его размер и не занимает места, то есть отображается над другими элементами.

С помощью свойства `outline-offset` можно изменять расположение рамки. Положительное значение отодвигает рамку от внешнего края элемента, а отрицательное «втягивает» внутрь.

У `outline-style` те же значения, что и у `border-style`. Кстати, есть интересные типы рамок `ridge` и `groove`, они создают объёмные рамки.

Скругление углов, часть 1

Закруглять углы элементов можно с помощью свойства `border-radius`.

Свойство задаёт радиус скругления углов в пикселях или процентах. Если у блока есть рамка, то скругляется и она.



Свойство `border-radius` принимает от одного до четырёх аргументов.

`border-radius: 10px`



`border-radius: 10px 20px`



`border-radius: 10px 20px 30px`



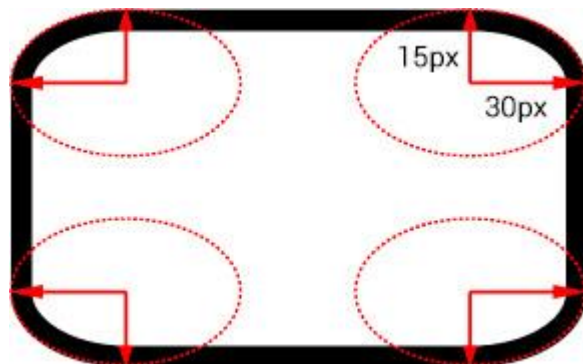
`border-radius: 10px 20px 30px 40px`



Скругление углов, часть 2

Вы можете закруглять отдельные углы с помощью свойств: `border-top-left-radius`, `border-top-right-radius`, `border-bottom-right-radius` и `border-bottom-left-radius`.

А ещё можно задавать разные горизонтальные и вертикальные радиусы скругления. Для этого нужно написать два значения через пробел в свойстве скругления угла. Первое значение задаёт радиус по горизонтали, второе — по вертикали:



```
/* горизонтальный радиус 30px, вертикальный 15px */
```

```
border-top-right-radius: 30px 15px;
```

Разные горизонтальные и вертикальные радиусы можно задавать и в свойстве `border-radius`. Для этого нужно использовать `/`, например:

```
/* горизонтальный радиус всех углов 10px, вертикальный 5px */
```

```
border-radius: 10px / 5px;
```

```
/* разные горизонтальные и вертикальные радиусы у каждого угла */
```

```
border-radius: 10px 20px 30px 40px / 5px 15px 25px 35px;
```

Изображение рамки: `border-image-source`

Мы подошли к обширной и интересной теме, в которой рамки и фоновые изображения встречаются.

Итак, семейство свойств `border-image` задаёт фоновое изображение для рамки блока. Поддержка данного семейства свойств в современных браузерах [довольно неплохая](#).

Свойство `border-image-source` задаёт путь к изображению рамки. По умолчанию картинкой заполняются только углы рамки. В следующих заданиях мы разберём, как можно управлять отображением рамки.

В качестве изображения для рамки используем вот такую картинку:



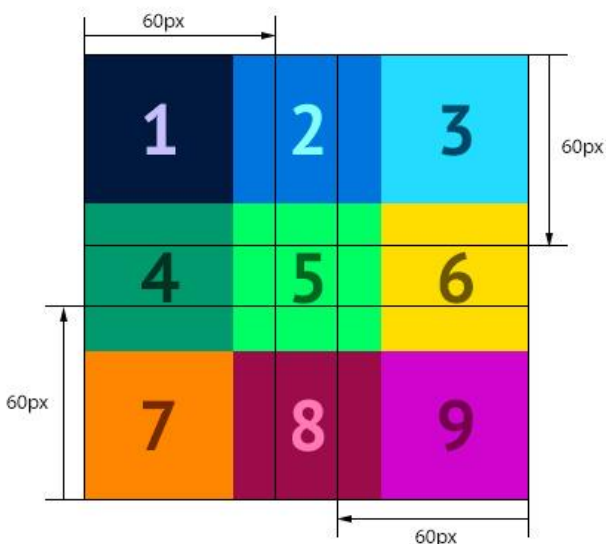
Синтаксис свойства такой же, как у `background-image`, то есть:

```
border-image-source: url("image.jpg");
```

Давайте же зададим фоновое изображение рамки и начн

Изображение рамки: `border-image-slice`

Давайте разберёмся, как работает механизм «нарезки» фонового изображения для рамки.



Каждая рамка имеет `9` областей: `4` угла, `4` стороны и центральную область. Для заполнения этих областей браузер должен нарезать картинку для рамки на `9` частей. Когда браузер не знает, как это сделать, он просто размещает картинку по углам — мы видели это в предыдущем задании.

Свойство `border-image-slice` задаёт отступы от краёв картинки до четырёх линий, которые «разрезают» её на части, как на схеме справа. Если эти отступы небольшие, то получается «нарезка» из `9` частей, которые затем размещаются в соответствующих областях рамки.

Но если отступы слишком большие (больше половины картинки), то браузер не может получить `9` частей и располагает то, что отрезалось по углам.

Значение свойства можно задавать числом без единицы измерения (оно обычно обозначает пиксели) или в процентах (относительно размера самой картинки). Пример:

```
border-image-slice: 60;
```

```
border-image-slice: 10%;
```

Нарезка несимметричных картинок

Сравните две картинки:



Для нарезки первой из них можно было задать одинаковые отступы линий разреза — `50px`. Для второй картинки этого явно недостаточно — она менее симметрична.

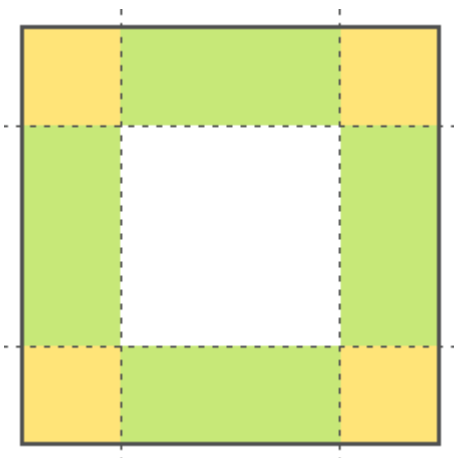
С помощью `border-image-slice` можно задавать разные отступы линий разреза. Для этого нужно задавать значения через пробел в порядке: верх, право, низ, лево. Пример:

```
border-image-slice: 10 20 30 40;
```

Средняя часть картинки обычно не используется. Но если в значение свойства добавить ключевое слово `fill`, то средняя часть картинки будет отображаться в средней области рамки: она закроет собой фон блока, но не закроет содержимое. Пример:

```
border-image-slice: 10 20 30 40 fill;
```

Изображение рамки: `border-image-repeat`, часть 1



Свойство `border-image-repeat` задаёт способ заполнения фоном боковых сторон рамки (зелёные области на рисунке).

У свойства четыре значения: `stretch`, `repeat`, `space` и `round`.

Значение по умолчанию — `stretch`. При этом значении фоновые картинки растягиваются на всю длину боковых сторон.

Если задано значение `repeat`, то фоновые картинки будут повторяться. При этом они могут обрезаться.

Можно устанавливать режим заполнения отдельно для горизонтальных и вертикальных сторон рамки.

Например:

```
/* все стороны рамки заполняются в режиме stretch */  
  
border-image-repeat: stretch;  
  
  
  
  
  
  
  
  
  
/* горизонтальные стороны — режим repeat, вертикальные — stretch */  
  
border-image-repeat: repeat stretch;
```

Изображение рамки: `border-image-repeat`, часть 2

Значение `round` свойства `border-image-repeat` тоже устанавливает режим заполнения стороны рамки повторяющимися боковыми участками картинки. Но, в отличие от `repeat`, если в ширину стороны не вмещается целое число повторяющихся кусочков, крайние части не обрезаются. Кусочки при этом равномерно растягиваются так, чтобы все они стали одного размера и заняли оставшееся место стороны рамки.

Согласно спецификации при заданном значении `space` алгоритм похож на `round`, только для компенсации оставшегося места кусочки картинки не растягиваются, а остаются прежней ширины, при этом между кусочками появляется дополнительное свободное пространство.

Изображение рамки: `border-image-width`

Следующее свойство, которое мы рассмотрим — `border-image-width`.

У блока должна существовать рамка определённой толщины `border-width`, тогда ему можно задать и фоновую картинку для рамки. Область, в которой будет отображаться эта картинка по умолчанию равна ширине рамки.

Свойство `border-image-width` позволяет управлять шириной видимой области рамки-картинки, масштабировать её. Саму ширину рамки это свойство не меняет.

Если значение этого свойства больше `border-width`, картинка рамки заползёт под содержимое, даже если не задано свойство `fill`.

Ширина рамки-картинки задаётся в `%`, `px`, `em` или других единицах измерения. Также возможно значение `auto`, при котором ширина зависит от значения `border-image-slice`.

Можно задавать разную ширину сторон. В этом случае значения перечисляются аналогично `margin`, `padding` в последовательности: верхнее, правое, нижнее, левое. Например:

```
border-image-width: 10px 20px 30px 40px;
```

```
border-image-width: 10px 50px;
```

Изображение рамки: `border-image-outset`

Ещё одно свойство, относящееся к фоновому изображению рамки, `border-image-outset`.

Аналогично `outline-offset` это свойство позволяет отодвинуть рамку за пределы элемента, но при этом одновременно немного масштабируя картинку. Отрицательные значения `border-image-outset` не поддерживаются.

Отступы рамок-изображений тоже можно задавать разные для каждой из сторон. Синтаксис обычный:

```
border-image-outset: 10px;
```

```
border-image-outset: 10px 20px 30px 40px;
```

Рамки и треугольники, часть 1

У рамок есть одна особенность: если рамка элемента широкая, а сам элемент имеет нулевую ширину и высоту, то стороны рамки становятся треугольными.

```
<div class="square"></div>
```

```
.square {
```

```
margin: 100px auto;
```

```
width: 0px;
```

```
height: 0px;
```

```
border: 100px solid #ffffff;
```

```
border-right-color: #2ecc40;
```

```
border-left-color: #0074d9;
```

```
border-top-color: #ff4136;
```

```
border-bottom-color: #ffdc00;
```

```
background-color: #aaaaaa;
```

}

Рамки и треугольники, часть 2

Теперь чтобы сделать треугольник нужно:

- одной стороне рамки задать нулевую ширину,
- ещё две стороны сделать прозрачными,
- последней стороне задать нужный цвет.

Так с помощью обычного `border` можно создавать треугольные стрелки, направленные во все стороны.

```
<div class="arrow-bottom"></div>
```

```
.arrow-bottom {  
  border: 100px solid #ffffff;  
  border-top-color:#0074d9;  
  border-bottom-width:0;  
  border-left-color:transparent;  
  border-right-color:transparent;  
}
```

Рамки и треугольники, часть 3

Похожим способом можно создавать треугольные стрелки, направленные по диагонали.

Для этого:

- одной стороне рамки задаём нужный цвет,
- ещё двум сторонам задаём нулевую ширину,
- последнюю сторону делаем прозрачной.