

# Новые возможности CSS3

## Подключение сторонних шрифтов

Ранее приходилось работать с ограниченным набором шрифтов, пригодных для веб-страниц. В данном случае речь идет о шрифтах, о которых заведомо известно, что они работают на разных браузерах и в разных операционных системах. Данная ситуация не подходит для современных веб-страниц, где шрифты играют огромную роль в создании общего впечатления о документе.

В CSS3 поддержка сложных шрифтов обеспечивается посредством возможности **@font-face**, которая применяется следующим образом:

1. Требуемый шрифт (или, более вероятно, несколько версий шрифта для поддержки разных браузеров) загружается на сайт.
2. Каждый шрифт регистрируется в таблице стилей с помощью команды **@font-face**.
3. Зарегистрированный шрифт используется в правилах стиля указанием его названия, точно так же, как и обычные веб-шрифты.
4. Когда браузер обнаруживает таблицу стилей, в которой используется специальный веб-шрифт, он загружает этот шрифт с сервера в свой кэш для временного хранения страниц и изображений. После этого браузер использует этот шрифт только для данной страницы или сайта. Если этот же шрифт указывается в другой странице, он должен быть зарегистрирован на этой странице и загружен на ее сервер, откуда он может быть загружен браузером в свой кэш.

## Форматы веб-шрифтов

Хотя все современные браузеры поддерживают возможность **@font-face**, не все они поддерживают одинаковые типы файлов шрифтов. Например, Internet Explorer, который обеспечивает использование **@font-face** в течение многих лет, поддерживает только файлы типа EOT (Embedded OpenType). Этот формат предоставляет ряд преимуществ, например, в нем используется сжатие для уменьшения объема файла шрифтов, а также применяется строгое лицензирование для веб-сайтов, чтобы шрифт нельзя было украсть с одного сайта и использовать на другом.

Но формат EOT никогда ни пользовался большой популярностью и не используется никакими другими браузерами. Вместо него браузеры работают с более знакомыми стандартами шрифтов, применяемыми в компьютерных приложениях — TTF (TrueType) и OTF (OpenType PostScript). Кроме этого, существуют еще два типа отображения шрифтов — SVG и WOFF. В таблице ниже дано краткое описание всех этих форматов шрифтов:

Форматы внедряемых шрифтов

Формат	Описание	Используется в
TTF (TrueType), OTF (OpenType PostScript)	Распространенные форматы шрифтов настольных компьютеров	Firefox (до версии 3.6), Chrome (до версии 6), Safari и Opera
EOT (Embedded OpenType)	Формат, специфичный для продуктов корпорации Microsoft. Не завоевал популярности у браузеров, за исключением Internet Explorer	Internet Explorer (до версии IE 9)
SVG (Scalable Vector Graphics)	Универсальный графический формат, который можно использовать для создания шрифтов. Дает хорошие, но не отличные результаты — медленно отображается и демонстрирует текст пониженного качества	Safari Mobile (на iPhone и iPad до iOS 4.2) и мобильные устройства под управлением операционной системы Android
WOFF (Web Open Font Format)	Возможно, единый формат шрифтов будущего. Поддерживается новыми версиями браузеров	Любой поддерживающий браузер, начиная с Internet Explorer 9, Firefox 3.6 и Chrome 6

В итоге можно сказать следующее: если вы хотите использовать возможность @font-face и поддерживать широкий диапазон браузеров, вам нужно предоставлять ваш шрифт в нескольких разных форматах. Как минимум, шрифт нужно предоставить в формате TTF или OTF (без разницы), EOT и SVG. Хорошо (но не обязательно) также предоставить шрифт в перспективном формате WOFF, может стать более популярным

и лучше поддерживаемым в будущем. (Одним из достоинств этого формата является использование сжатых файлов, что сокращает время их загрузки.)

Даже если вы последуете всем приведенным ранее правилам и предоставите все требуемые форматы шрифтов, ожидайте некоторые загвоздки. В частности, с веб-шрифтами иногда возникают следующие проблемы:

1. Многие шрифты выглядят плохо в операционной системе Windows XP, т.к. в настройках многих компьютеров с этой операционной системой отключено сглаживание (anti-aliasing). (А без применения сглаживания шрифты выглядят очень непривлекательно.)
2. От пользователей поступали жалобы о проблемах с печатью определенных внедренных шрифтов из некоторых браузеров или в операционных системах.
3. Некоторые браузеры страдают проблемой FOUT (Flash of Unstyled Text, вспышка нестилизованного текста). Это явление происходит, когда внедренный шрифт не успевает загрузиться вовремя, и страница отображается сначала в резервном шрифте, а потом воспроизводится повторно на встроенном шрифте. Эта проблема особенно заметна в старых версиях браузера Firefox. Если вас это сильно беспокоит, можете воспользоваться библиотекой JavaScript от Google, позволяющей разработчику определить резервные стили, которые используются вместо незагруженных внедренных шрифтов, таким образом, предоставляя ему полный контроль над воспроизведением текста в любое время.

Хотя эти небольшие проблемы иногда и возникают, большинство из них постепенно решается с выпуском новых версий браузеров. Например, браузер Firefox теперь сводит эффект FOUT к минимуму, ожидая до трех секунд, пока не загрузится внедренный шрифт, прежде чем использовать резервный шрифт.

## Наборы шрифтов

На рисунке ниже показано несколько шрифтов из предоставляемых на сайте Font Squirrel: [www.fontsquirrel.com](http://www.fontsquirrel.com)

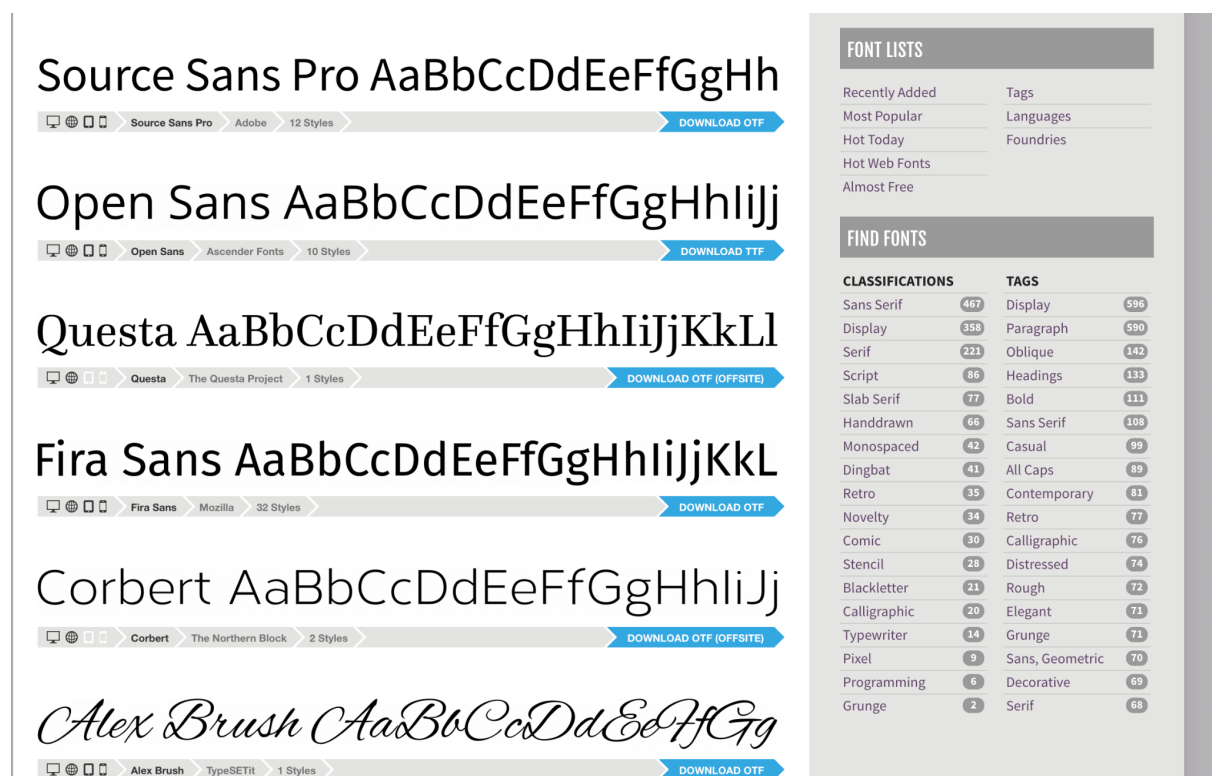


Рисунок 1 – пример выбора шрифтов с сайта Font Squirrel

Сайт Font Squirrel предоставляет для загрузки несколько сотен высококачественных шрифтов, организованных в разделы по категориям (такие как, например, Calligraphic, Grunge или Retro). Все эти шрифты бесплатные для любого использования, будь то на персональном компьютере для создания документов или на веб-странице в Интернете.

Набор шрифтов загрузится в виде файла, сжатого в формате ZIP, который содержит несколько файлов шрифта в разных форматах. Чтобы использовать шрифт в своей веб-странице, файлы разных форматов шрифта нужно загрузить на веб-сервер в папку этой веб-страницы. После этого шрифт нужно зарегистрировать, чтобы он был доступным для использования в таблице стилей. Регистрация выполняется с помощью правила @font-face в начале таблицы стилей, которое выглядит следующим образом:

```
@font-face {  
  
    font-family: 'MetrophobicRegular';  
  
    src: url('Metrophobic-webfont.eot');  
  
    src: local('Metrophobic'),  
  
        url('Metrophobic-webfont.eot?#iefix') format('embedded-opentype'),
```

```
url('Metrophobic-webfont.woff') format('woff'),  
url('Metrophobic-webfont.ttf') format('truetype'),  
url('Metrophobic-webfont.svg#MetrophobicRegular') format('svg');  
font-weight: normal;  
font-style: normal;  
}
```

Строки приведенного кода выполняют следующие функции:

1. Выражение `@font-face` регистрирует шрифт для его дальнейшего применения в таблице стилей.
2. Шрифту можно присвоить любое название. Это название будет употреблено позже, при использовании шрифта.
3. Первым должен быть указан формат ЕОТ, т.к. дальнейшая часть правила сбивает с толку Internet Explorer, и тот не обращает внимания на остальные форматы. Функция таблицы стилей `url()` указывает браузеру загрузить файл из обозначенного URL. Если шрифт размещен в одной папке с веб-страницей, то здесь можно просто указать название файла.
4. Функция `local()` указывает браузеру название шрифта, и если этот шрифт установлен на компьютере посетителя веб-страницы, браузер использует локальный шрифт. Но в редких случаях это может вызвать проблемы. Например, в зависимости от того, где установлен шрифт на компьютере посетителя, компьютеры Mac OS X могут вывести предупреждение о нарушении безопасности, или же может загрузиться другой шрифт с таким же названием, как и ваш. По этой причине веб-разработчики иногда указывают явно несуществующее имя файла, чтобы браузер не нашел локального шрифта. В качестве простого имени такого типа можно использовать какой-либо бессмысленный символ.
5. Последний шаг — это сообщить браузеру о других файлах шрифтов, которые он может использовать. Если имеется файл шрифта типа WOFF, укажите этот файл первым, т.к. данный формат предоставляет наилучшее качество шрифта. Следующим укажите файл шрифта формата TTF или OTF, а самым последним — файл формата SVG.

Зарегистрировав веб-шрифт с помощью функции `@font-face`, вы можете использовать его в любой таблице стилей. Для этого используется уже знакомое нам свойство `font-family`, которому присваивается значение в виде названия семейства шрифтов, зарегистрированного с помощью функции `@font-face` (в строке 2). Далее приведен пример использования этого шрифта в правиле таблицы стилей:

```
body {  
  
    font-family: 'Open Sans';  
  
}
```

Это правило применяет веб-шрифт ко всей странице, хотя область его применения, конечно же, можно было бы ограничить определенными элементами или применить классы. Но шрифт нужно в обязательном порядке зарегистрировать до того, как использовать его в правиле таблицы стилей. Если выполнить эти шаги в обратном порядке, шрифт не будет работать должным образом.

## Добавление теней к тексту

С помощью нового CSS3 свойства `text-shadow` Вы можете добавлять к тексту элементов тени (к тексту одного элемента может быть добавлено одновременно несколько теней).

При задании тени для текста необходимо указать: величину смещения тени от текста по горизонтали и вертикали (может быть отрицательной), а также радиус размытия и цвет тени.

```
.shadow1 {  
  
    text-shadow: 3px 2px #FFAE00;  
  
}  
  
.shadow2 {  
  
    text-shadow: 1px 1px 10px #FFAE00;  
  
}  
  
.shadow3 {  
  
    text-shadow: 2px 2px 2px #FFAE00, 2px 2px 15px #1435AD;  
  
}
```

## Свойство text-overflow

В CSS3 было добавлено новое свойство text-overflow, которое позволяет указать, что должно случиться с текстом вышедшем за пределы границ элемента.

```
.wrap1 {  
    text-overflow: ellipsis;  
    overflow: hidden;  
}
```

```
.wrap2 {  
    text-overflow: clip;  
    overflow: hidden;  
}
```

## Свойство word-wrap

С помощью нового CSS3 свойства word-wrap Вы можете указать, что длинные слова выходящие за пределы границ элемента должны разделяться и переноситься на новую строку.

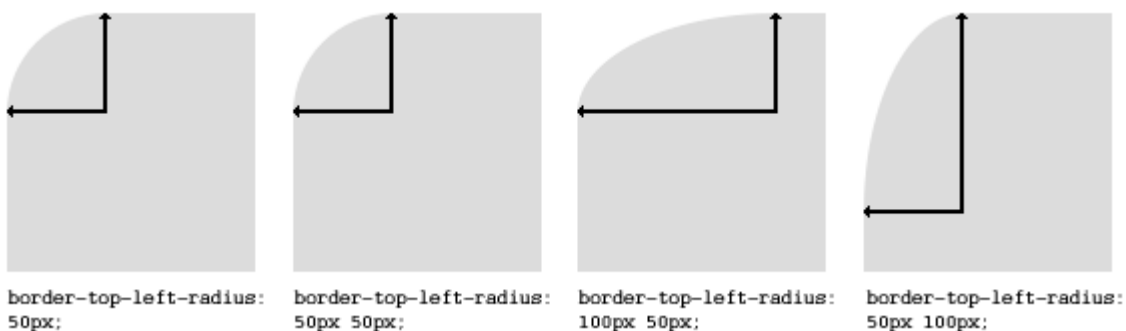
```
.wrap2 {  
    word-wrap: break-word;  
}
```

## CSS3 Рамки

CSS3 рамка дополняет возможности форматирования границ элементов с помощью свойств, позволяющих закруглить углы элемента, а также использовать изображения для оформления границ элемента.

1.

### Закругление углов с помощью border-radius



Свойство позволяет закруглить углы строчных и блочных элементов. Кривая для каждого угла определяется с помощью одного или двух радиусов, определяющих его форму — круга или эллипса. Радиус распространяется на весь фон, даже если элемент не имеет границ, точное положение секущей определяется с помощью свойства background-clip.

Свойство border-radius позволяет закруглить все углы одновременно, а с помощью свойств border-top-left-radius, border-top-right-radius, border-bottom-right-radius, border-bottom-left-radius можно закруглить каждый угол отдельно.

Если задать два значения для свойства border-radius, то первое значение закруглит верхний левый и нижний правый угол, а второе — верхний правый и нижний левый.

Значения, заданные через /, определяют горизонтальные и вертикальные радиусы. Свойство не наследуется.

<b>border-radius</b> <b>(border-top-left-radius, border-top-right-radius, border-bottom-right-radius, border-bottom-left-radius)</b>	
Значения:	
длина	Позволяет закруглить углы блока с помощью значений единиц длины.
%	Значения, закругляющие углы, задаются в процентах от длины и ширины сторон элемента.
initial	Устанавливает значение свойства в значение по умолчанию.
inherit	Наследует значение свойства от родительского элемента.

```
.r1 {  
    border-radius: 0 0 20px 20px;  
}  
  
.r2 {  
    border-radius: 0 10px 20px;  
}  
  
.r3 {  
    border-radius: 10px 20px;
```



```
}

.r4 {

    border-radius: 10px/20px;

}

.r5 {

    border-radius: 5px 10px 15px 30px/30px 15px 10px 5px;

}

.r6 {

    border-radius: 10px 20px 30px 40px/30px;

}

.r7 {

    border-radius: 50%;

}

.r8 {

    border-top: none;

    border-bottom: none;

    border-radius: 30px/90px;

}

.r9 {

    border-bottom-left-radius: 100px;

}

.r10 {

    border-radius: 0 100%;

}

.r11 {

    border-radius: 0 50% 50% 50%;

}

.r12 {

    border-top-left-radius: 100% 20px;
```

```
border-bottom-right-radius: 100% 20px;  
}
```

Пример закругления углов представлен на рисунке 1.

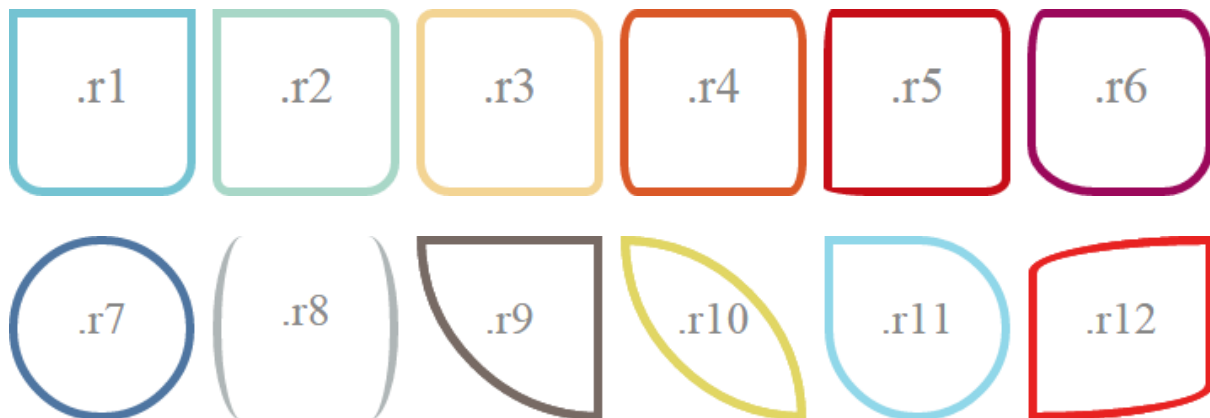
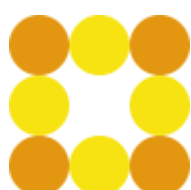


Рисунок 1 – Примеры закруглений углов блока

## Рамки-изображения border-image

Свойство позволяет устанавливать изображение в качестве рамки элемента. Основное требование, предъявляемое к изображению — оно должно быть симметричным. Свойство включает в себя следующие значения: {border-image: width source slice repeat outset;}

border-image	
Значения:	описание
краткая запись	Устанавливает рамку-изображение с помощью одного свойства, являющегося краткой записью свойств border-image-source, border-image-slice, border-image-width, border-image-outset и border-image-repeat. Значения по умолчанию: {border-image: 1 none 100% stretch 0; }.
initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.



При помощи данного изображения можно получить рамки для элемента.

```
div {  
    width: 260px;  
    height: 100px;  
    border-image-width: 15px;  
    border-image-source: url(border_round.png);  
    border-image-slice: 30;  
    border-image-repeat: stretch;  
}  
  
/* Пример 2 */  
div {  
    width: 260px;  
    height: 100px;  
    border-image-width: 15px;  
    border-image-source: url(border_round.png);  
    border-image-slice: 30;  
    border-image-repeat: round;  
}
```

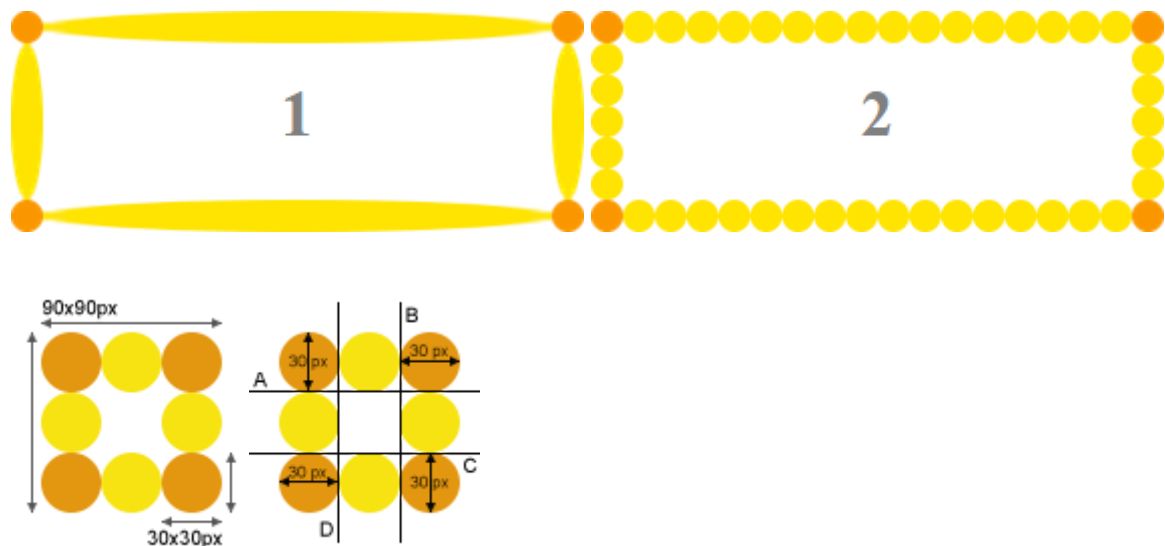


Рисунок 2 – Пример оформления блоков с помощью изображения

Срезы А — В — С — D образуют углы рамки, а часть рисунка, расположенная между ними, заполняет оставшееся пространство рамки в соответствии с заданным значением свойства `border-image-repeat`.

Размер угловой части (в данном примере это число 30), задается с помощью значения свойства `border-image-slice`.

## Ширина рамки-изображения `border-image-width`

Свойство задаёт ширину изображения для границы элемента. Если ширина не задана, то по умолчанию она равна 1.

border-image-width	
Значения:	Описание
длина	Устанавливает ширину рамки в единицах длины — px / em. Можно задавать от одной до четырех значений одновременно. Если задано одно значение, то ширина всех сторон рамки одинакова, два значения задают ширину верхней-нижней и правой-левой и т.д.
число	Числовое значение, на которое умножается значение border-width.
%	Ширина рамки элемента вычисляется относительно размера изображения. Горизонтальные относительно ширины, вертикальные — относительно высоты.
auto	Соответствует значению border-image-slice.
initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.

Синтаксис:

```
div {
  border-image-source: url(border.png);
  border-image-width: 30px;
}
```

## Ресурс рамки-изображения border-image-source

Свойство задаёт путь к изображению, которое будет использоваться для границы блока.

border-image-source	
Значения:	Описание
none	Отсутствие изображения для рамки. Значение по умолчанию.
url(url)	Относительный или абсолютный путь к изображению.

initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.

Синтаксис:

```
div {
  border-image-source: url(border.png);
}
```

## Элементы рамки-изображения border-image-slice

Свойство определяет размер частей изображения, используемых для оформления границ элемента и делит изображение на девять частей: четыре угла, четыре края между углами и центральную часть.

border-image-slice	
Значения:	Описание
число	<p>Размер частей рамки можно задавать с помощью одного, двух, трех или четырех значений.</p> <p>Одно значение устанавливает границы одинакового размера для каждой стороны элемента.</p> <p>Два значения: первое определяет размер верхней и нижней границы, второе — правой и левой.</p> <p>Три значения: первое определяет размер верхней границы, второе — правой и левой, а третье — нижней границы.</p> <p>Четыре значения: определяет размеры верхней, правой, нижней и левой границы.</p> <p>Числовое значение представляет количество px.</p>
%	<p>Размеры границ рассчитываются относительно размера изображения.</p> <p>Горизонтальные относительно ширины, вертикальные — относительно высоты.</p>
fill	<p>Значение указывается вместе с числом или процентным значением. Если оно задано, изображение не обрезается внутренним краем рамки, а заполняет также область внутри рамки.</p>

initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.

Синтаксис:

```
div {border-image-slice: 50 20;}
```



border.png  
размер 200x200px

```
div {
width: 200px;
height: 200px;
border-image-width: 40px;
border-image-source:
url(border.png);
border-image-slice: 40;
border-image-repeat: round;
}
```



<div></div>

Рисунок 3 – Пример задания срезов изображения

## Повтор рамки-изображения border-image-repeat

Свойство управляет заполнением фоновым изображением пространства между углами рамки. Можно задавать как с помощью одного значения, так и с помощью пары значений.

border-image-repeat	
Значения:	Описание
stretch	Растягивает изображение на все пространство. Значение по умолчанию.
repeat	Повторяет заполняющую часть изображения, при этом видны места стыков с угловой частью, и если длины образца не хватает, то он растягивается.
round	Наиболее точно заполняет промежуток между углами рамки, дублируя заполняющую часть изображения, при этом может образовывать стыки по середине стороны рамки.
space	Действует аналогично с repeat.

initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.

Синтаксис:

```
div {
  border-image-source: url(border.png);
  border-image-repeat: repeat;
}
```



Рисунок 4 – Пример повтора центральной рамки изображения

## Смещение рамки-изображение border-image-outset

Свойство позволяет переместить изображение-рамку за пределы границ элемента на указанную длину. Может задаваться как с помощью одного, так и четырёх значений.

border-image-outset	
Значения:	Описание
длина	Отступ рамки-изображения задается с помощью любого положительного числа, указанного в px или em.
число	Числовое значение, на которое умножается border-width.
initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.

## Смещение внешней рамки outline-offset

Свойство задаёт расстояние между границей элемента border и внешней границей, созданной с помощью свойства outline.

outline-offset	
Значения:	Описание
длина	Задаёт расстояние с помощью единиц длины — px / em. Отрицательное значение отображает рамку внутри элемента, положительное — снаружи элемента.
initial	Устанавливает это свойство в значение по умолчанию.
inherit	Наследует значение этого свойства от родительского элемента.