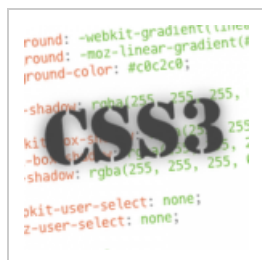




Графика HTML и CSS PHP JavaScript Freebies Познавательно

Главная страница → HTML и CSS → Уроки



CSS3 в действии: Основные инструменты

Оригинал статьи: [Devon Govett](#) CC-лицензия:

Опубликовано 05.09.10 / Автор перевода: [Иван Саломатин](#)

0



Комментариев 7

CSS3 – великолепная прогрессивная технология стилизации HTML-страниц, рекомендованная к применению веб-мастерам консорциумом W3C. Использование инструментов, предоставляемых CSS3, позволяет с помощью кодирования создавать эффекты, которые раньше можно было реализовать исключительно посредством внедрения в страницу дополнительных графических элементов.

Преимущества использования CSS3 очевидны: благодаря значительному сокращению количества изображений, уменьшается суммарный вес страниц и количество http-запросов, что приводит к существенному снижению времени загрузки; упрощается работа по оформлению элементов (для изменения того или иного эффекта нам достаточно изменить значение нужного параметра, не прибегая к помощи графического редактора). Конечно, отсутствие поддержки CSS3 старыми браузерами создаёт некоторые неудобства, но это, безусловно, лишь временные препятствия на пути к повсеместному использованию технологии. Более того, уже сегодня, с помощью специальных сценариев, можно заставить старые браузеры понимать CSS3.

Итак, приглашаю тебя, уважаемый читатель, познакомиться с CSS3 поближе.

В этой статье рассмотрим только наиболее употребляемые свойства технологии CSS3 – те, которые позволят нам создавать наиболее распространённые визуальные эффекты:

- Border Radius – округление уголков блочных элементов;
- Box Shadows – тени блочных элементов;
- Text Shadows – тени текста;
- Gradient – градиентная заливка;
- Multiple Backgrounds – множественные фоновые изображения.

В конце статьи я продемонстрирую технику создания замечательной html-кнопки исключительно средствами CSS3, и приведу несколько уникальных примеров, в полной мере раскрывающих всё великолепие этой технологии.

Ну, теперь обо всём по порядку...

BORDER RADIUS – ОКРУГЛЕНИЕ УГОЛКОВ БЛОЧНЫХ ЭЛЕМЕНТОВ

Поддерживается следующими браузерами:

- Safari: 3.2+
- Firefox: 3.0+
- Chrome: 3.0+
- Opera: 10.5+
- Internet Explorer: 9.0+

Думаю, об этом свойстве вы слышали уже не раз. Оно поддерживается практически всеми современными браузерами и значительно облегчает труд верстальщика. Взгляните на синтаксис свойства:

```
-webkit-border-radius: значение;
-moz-border-radius: значение;
border-radius: значение;
```

Вот так, всего 3 строки! Первая строка определяет радиус округления уголков элемента для браузеров, основанных на движке Webkit, таких как Safari и Chrome. Вторая – для браузеров на платформе Mozilla, в первую очередь – Firefox. И последняя строка, по сути, представляет наименование свойства без префикса, определяя значение радиуса для всех прочих браузеров, поддерживающих технологию CSS3, включая IE9, Opera и Safari 5.



Поиск

Ключевое слово или фраза

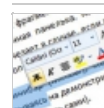
Популярные статьи



[Установка Python в GIMP для Windows](#)



[HTML-сниппеты: избавляемся от рутины](#)



[jQuery в помощь: BBCode html-toolbar средствами jQuery](#)



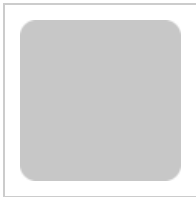
[TOR — анонимность в интернете. Установка и настройка. Использование](#)

[ExitNodes.](#)



[Яндекс.Карта на сайт: шаг за шагом](#)

Установив значение свойства `border radius` равным `10px`, получим вот такой результат:



BOX SHADOWS – ТЕНИ БЛОЧНЫХ ЭЛЕМЕНТОВ

Поддерживается следующими браузерами:

- Safari: 3.2+
- Firefox: 3.5+
- Chrome: 3.0+
- Opera: 10.5+
- Internet Explorer: 9.0+

Box shadows – ещё одно часто используемое свойство CSS3. Его синтаксис выглядит так:

```
-webkit-box-shadow: отступ_X отступ_Y радиус_размытия #цвет;  
-moz-box-shadow: отступ_X отступ_Y радиус_размытия #цвет;  
box-shadow: отступ_X отступ_Y радиус_размытия #цвет;
```

Здесь мы вновь вынуждены определять свойство несколько раз, используя префиксы для Webkit и Mozilla. Первый и второй параметры определяют отступы тени от границ элемента по осям X и Y, соответственно. Положительные значения параметров смещают тень вправо вниз относительно нижнего правого угла элемента, в то время как отрицательные значения – влево вверх относительно верхнего левого угла. Параметр «радиус_размытия» определяет насколько мягкой будет созданная тень. Наконец, последний параметр – цвет тени.

Применив свойство «shadow» к элементу, представленному в предыдущем примере, получим примерно такой результат:

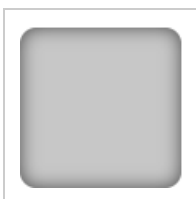
```
-webkit-box-shadow: 5px 5px 10px #555;  
-moz-box-shadow: 5px 5px 10px #555;  
box-shadow: 5px 5px 10px #555;
```



Внутренние тени

Ещё один полезный параметр рассматриваемого свойства – ключевое слово «inset». Если в значениях параметра присутствует ключ «inset», тень будет расположена внутри границ элемента. Это позволяет создавать «эффект объёмной фигуры» и «эффект окна», которые особенно полезны при стилизации кнопок.

```
-webkit-box-shadow: 0 0 20px #333 inset;  
-moz-box-shadow: 0 0 20px #333 inset;  
box-shadow: 0 0 20px #333 inset;
```



Множественные тени

Ещё одна очень полезная особенность свойства «shadow» заключается в том, что оно позволяет создавать множественные тени для одного и того же элемента. Эта особенность позволяет избежать потенциальных нагромождений кода на странице с HTML-разметкой. Чтобы создать множественную тень, просто разделяете группы параметров свойства запятыми:

```
-webkit-box-shadow: 0 0 20px #333 inset,  
20px 15px 30px yellow,  
-20px 15px 30px lime,  
-20px -15px 30px blue,  
20px -15px 30px red;
```

```
-moz-box-shadow: 0 0 20px #333 inset,  
20px 15px 30px yellow,  
-20px 15px 30px lime,  
-20px -15px 30px blue,  
20px -15px 30px red;
```

```
box-shadow: 0 0 20px #333 inset,  
20px 15px 30px yellow,  
-20px 15px 30px lime,  
-20px -15px 30px blue,  
20px -15px 30px red;
```



TEXT SHADOWS – ТЕНИ ТЕКСТА

Поддерживается следующими браузерами:

- Safari: 3.2+
- Firefox: 3.5+
- Chrome: 3.0+
- Opera: 10.1+
- Internet Explorer: Пока нет, но, будем надеяться, скоро будет!

В отличие от «box shadows», «text shadows» создаёт тени только вокруг текстовых символов, заключённых в элемент, к которому применено это свойство. К счастью, «text shadows» не требуется указывать несколько раз, оно и без специального префикса одинаково хорошо понимается браузерами.

Параметры «text shadows» такие же, как у свойства «box-shadows», разница лишь в том, что для «text shadow» не применяется параметр «inset» – не поддерживается он этим свойством.

```
color: #fff;  
text-shadow: 0 0 50px #333;
```

Whoa, text shadow!

Как и «box shadows», это свойство позволяет создавать множественные тени:

```
text-shadow: 0 0 4px #ccc,  
0 -5px 4px #ff3,  
2px -10px 6px #fd3,  
-2px -15px 11px #f80,  
2px -18px 18px #f20;
```

Whoa, this text is on fire!

Как видите, «text shadows» позволяет с лёгкостью создавать замечательные эффекты.

GRADIENT – ГРАДИЕНТНАЯ ЗАЛИВКА

Поддерживается следующими браузерами:

- Safari: 4+
- Firefox: 3.6+
- Chrome: 5+
- Opera: Пока нет, но, будем надеяться, скоро будет!
- Internet Explorer: Пока нет, но, будем надеяться, скоро будет!

Свойство «gradient» – мощнейший инструмент для создания великолепных эффектов. Это свойство пока поддерживается не всеми браузерами, тем не менее, для таких браузеров вы всегда можете сформировать дополнительные стили с использованием фоновых изображений.

Существует два типа градиентов, которые вы можете создать, применив данное свойство: линейный и радиальный. Как вы, вероятно, догадались, линейный градиент представляет собой постепенное изменение цвета заливки вдоль оси, в то время как радиальный – от

центра вписанной в объект окружности к её краям на длину радиуса. Свойство «gradients» может быть применено для всех свойств, которые позволяют использовать изображения в качестве фона: background, border-image, и т.д.

К сожалению, в отличие от вышеупомянутых, это свойство не столь универсально, нам придётся использовать различный синтаксис для определения значений параметров: один – для Mozilla, другой – для Webkit.

Синтаксис для Webkit

Технология рендеринга у браузеров Safari и Chrome основана на движке Webkit, поэтому для этих браузеров синтаксис определения параметров совпадает.

```
-webkit-gradient(<type>, <point> [, <radius>]?, <point> [, <radius>]?, <stop>]*)
```

Как видите, синтаксис для Webkit достаточно прост. Более того, он одинаков как для линейного, так и для радиального градиентов – нужно лишь указать соответствующий тип.

Синтаксис для Mozilla

Firefox использует различный синтаксис для линейного и радиального градиентов, каждый из которых кардинально отличается от того синтаксиса, который используется для Webkit.

```
-moz-linear-gradient( [<point> || <angle>]?, <stop>, <stop> [, <stop>]*)  
-moz-radial-gradient( [<position> || <angle>]?, [<shape> || <size>]?, <stop>, <stop>[,
```

Теперь рассмотрим синтаксис для каждого типа градиентов в отдельности.

Линейный градиент

Синтаксис параметров линейного градиента для Webkit выглядит так:

```
-webkit-gradient(linear, start_x start_y, end_x end_y, stop, stop...)
```

Тот же линейный градиент для Firefox задаётся следующим образом:

```
-moz-linear-gradient (angle start_x start_y, stop, stop...)
```

В синтаксисе для Webkit start_x и start_y – это координаты начальной точки градиентной заливки, в то время как end_x и end_y – координаты конечной точки. Для значений параметров, определяющих позицию точек, вы можете использовать любые единицы измерения, допустимые в CSS, включая % и ключевые слова «left», «top», «bottom», «right» и «center».

Ключи «stop» в приведённом синтаксисе – это значения цвета в формате RGB для точек начала и конца градиента. Кроме начальной и конечной точки вы можете определять множество дополнительных опорных точек, каждой из которых может быть задан свой цвет. Это позволяет создавать множественные градиентные заливки, состоящие из оттенков более чем 2-х цветов.

Опорных точек может быть задано сколько угодно, но определение цвета для начальной и конечной точки – обязательно условие синтаксиса.

Кроме жёсткого определения координат опорных точек, вы можете использовать функцию color-stop, с помощью которой можно задать положение точки относительно общей длины направляющей. Относительная позиция опорной точки, заданной с помощью этой функции, может быть определена как в процентах, так и с помощью десятичной дроби в диапазоне от 0 до 1 с точностью до сотых. Например, «color-stop(50%, black)» создаст опорную точку чёрного цвета ровно посередине направляющей градиента.

Вместо того, чтобы определять координаты начальной и конечной точек в процентах как 0% и 100%, можно воспользоваться функциями from() и to. Здесь в скобках следует просто указать значение цвета в RGB-формате.

Синтаксис для Firefox более прост. Единственное, что осталось неизменным в сравнении с синтаксисом для Webkit, – это способ определения цвета начальной, конечной и опорных точек. Определение координат конечной точки этот синтаксис не предусматривает. Конечной точкой принимается край элемента, к которому применена градиентная заливка, а направление оси определяется с помощью значения угла наклона (angle). В случае, если кроме начальной точки, вы определите несколько опорных, они будут равномерно распределены по длине направляющей. Если вы не укажете значение угла наклона оси, он будет выбран равным 0 и ось будет расположена горизонтально, а изменение цвета будет происходить слева направо. Если вы не укажете координаты начальной точки, то направляющий вектор заливки будет расположен сверху вниз на длину равную высоте элемента.

Как видите, этот синтаксис несколько менее гибкий, нежели синтаксис для Webkit...

Вот простейший пример формирования градиентной заливки средствами CSS 3. Направляющий вектор этого градиента располагается сверху вниз от верхнего левого угла элемента к нижнему. Цвет изменяется от красного к белому.

```
-webkit-gradient (linear, left top, left bottom, from(red), to(white));  
-moz-linear-gradient (red, white)
```

Применив это свойство к фону некоторого элемента div, получим следующий результат:



Добавив опорную точку и изменив угол оси можно получить, например, такой результат:

```
-webkit-gradient(linear, left top, right bottom, from(red), color-stop(50%,  
-moz-linear-gradient(left top, red, white, red));
```



Радиальный градиент

Синтаксис радиального градиента для Webkit выглядит так:

```
-webkit-gradient (radial, inner_circle_center_x inner_circle_center_y, inner_circle_rad  
outer_circle_center_x outer_circle_center_y, outer_circle_radius,  
stop, stop...)
```

А вот синтаксис для Firefox:

```
-moz-radial-gradient(center_x center_y, shape size, stop, stop...)
```

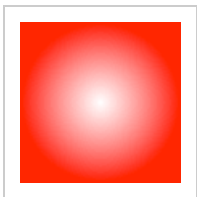
Синтаксис радиального градиента для Webkit несколько сложнее, нежели у линейного, но строится всё по тому же принципу. Радиальный градиент для Webkit строится на основании двух окружностей – начальной и конечной. В параметрах задаются координаты центральных точек окружностей и значения длин радиусов. Цвета опорных точек определяются последними, так же как и для линейного градиента.

Для Firefox синтаксис радиального градиента весьма схож с синтаксисом линейного. Вы можете определить координаты центральной точки (делается это так же, как и определение координат внутренней окружности в синтаксисе Webkit), но внешняя окружность в данном синтаксисе отсутствует. Вместо этого вы можете указать форму и длину градиента. Формой градиента может быть как окружность, так и эллипс, чего, кстати, невозможно добиться для Webkit. Размер может быть задан посредством использования множества разнообразных ключей, но, ни как не в пикселах, как вы, вероятно, ожидали. Подробнее об этом вы можете узнать в [Центре разработки Mozilla](#). Для целей настоящей статьи мы попросту используем значения по-умолчанию.

Если вы не зададите координаты центральной точки, она будет расположена в центре объекта, к которому применено данное свойство. Если не установите размеры и тип фигуры, градиент полностью заполнит объект, а в качестве формы будет применена окружность.

```
-webkit-gradient (radial, center center, 0, center center, 50, from(white), to(red));  
-moz-radial-gradient (white, red);
```

Если применить этот градиент к фону некоторого div'a, получим вот такой результат:



MULTIPLE BACKGROUNDS – МНОЖЕСТВЕННЫЕ ФОНОВЫЕ ИЗОБРАЖЕНИЯ

Поддерживается следующими браузерами:

- Safari: 3.2+
- Firefox: 3.6+
- Chrome: 3.0+

- Opera: 10.5+
- Internet Explorer: 9.0+

Множественные бэкграунды значительно упрощают создание сложных эффектов средствами CSS, исключая необходимость вводить дополнительные элементы в HTML-разметку. Сам по себе множественный бэкграунд означает, что элементу может быть присвоено и фоновое изображение, и фоновый цвет, и фоновая градиентная заливка одновременно. С позиции синтаксиса, множественный бэкграунд ни чем не отличается от одиночного, за исключением того, что свойство «background» будет содержать несколько групп параметров, отделяемых друг от друга запятыми.

Вот пример div'a, в качестве фона которого заданы одновременно и градиентная заливка и некоторое изображение:

```
background: url(noise.png), -webkit-gradient(linear, left top, left bottom, from
background: url(noise.png), -moz-linear-gradient(#999, #333);
```

Результат будет выглядеть вот так:



ПОСТРОЕНИЕ ИЗЯЩНОЙ КНОПКИ СРЕДСТВАМИ CSS3

Что ж, теперь в нашем техническом арсенале добавилось пять замечательных новых инструментов, давайте попробуем применить их на практике.

Конечно, можно создать бесчисленное множество разнообразных кнопок, но, чтобы не усложнять себе задачу, для примера возьмём что-нибудь из уже существующего – кнопку в стиле Mac OS.

Начнём с формирования HTML-разметки. Нам потребуется всего пара дивов.

```
<div class="panel">
<div class="button">Button</div>
</div>
```

Теперь приступим к оформлению. Сперва, оформим панельку.

```
.panel {
background: -webkit-gradient(linear, left top, left bottom, from(#bbbcb), to(#959695)
background: -moz-linear-gradient(#bbbcb, #959695);
background-color: #b7b9b7;
width: 100px;
}
```

Панелька – это всего лишь контейнер для нашей кнопки. Фон панельки задан с помощью градиентной заливки из оттенков серого цвета направленной сверху вниз. Для тех браузеров, которые не воспринимают свойство «gradient», используем свойство «background-color».

Стилизуем саму кнопку.

```
.button {
display: inline-block;
margin: 20px;
padding: 3px 6px;

font-family: 'Lucida Grande', Arial, sans-serif;
font-size: 13px;

-webkit-border-radius: 3px;
-moz-border-radius: 3px;
border-radius: 3px;
border: 1px solid rgba(0, 0, 0, 0.6);

background: -webkit-gradient(linear, left top, left bottom, from(#fbfcfb), to(#9d9e9d)
background: -moz-linear-gradient(#fbfcfb, #9d9e9d);
background-color: #c0c2c0;

text-shadow: rgba(255, 255, 255, 0.4) 0px 1px;

-webkit-box-shadow: rgba(255, 255, 255, 0.4) 0 1px;
-moz-box-shadow: rgba(255, 255, 255, 0.4) 0 1px;
box-shadow: rgba(255, 255, 255, 0.4) 0 1px;

-webkit-user-select: none;
-moz-user-select: none;
cursor: default;
}
```

В этом небольшом фрагменте кода мы прибегли к использованию почти всех рассмотренных выше свойств, за исключением множественных бэкграундов. Тут можно выделить несколько логических секций – я отделил их друг от друга пустой строкой.

Первая пара строк – общие стили. Установив тип элемента «div.button» в значение «inline-block», мы заставляем его автоматически расширяться, охватывая весь заключённый в него текст. Следующая секция содержит настройки шрифта.

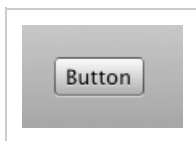
В третьей секции определяем радиус округления уголков кнопки. Далее, определяем стили границы элемента «div.button». Обратите внимание на то, что мы используем здесь формат rgba, определив некоторый уровень непрозрачности. Полупрозрачная граница элемента позволит сделать нашу кнопку визуально более объёмной и реалистичной.

Следующим шагом оформляем бэкграунд нашей кнопочки. Используем градиентную заливку, а на случай просмотра в браузерах, не поддерживающих это свойство, подстраховываемся с помощью жёсткого определения цвета фона. Затем, применяем свойство «text-shadow», чтобы добавить некоторую рельефность тексту. Тут вновь используем формат rgba.

Предпоследняя секция определяет настройки свойства «box-shadow». Его мы используем здесь для того, чтобы придать нашей кнопке рельеф. Мы, как бы, создаём ещё одну рамку, только несколько размытую. Этот трюк используется довольно часто в тех случаях, когда требуется создать эффект объёмной фигуры.

Наконец, последняя секция содержит ещё один замечательный трюк – с помощью «user-select: none;» мы блокируем пользователю возможность выделения текста кнопки.

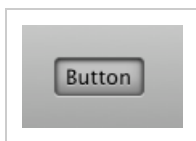
Итак, на данном этапе мы уже имеем вот такую вот кнопку:



Ну, уже неплохо, но надобно ещё немного доработать.

```
.button:active {
  background: #B5B5B5;
  -webkit-box-shadow: rgba(255, 255, 255, 0.4) 0 1px,
    black 0px 1px 3px inset,
    rgba(0, 0, 0, 0.4) 0px -5px 12px inset;
  -moz-box-shadow: rgba(255, 255, 255, 0.4) 0 1px,
    black 0px 1px 3px inset,
    rgba(0, 0, 0, 0.4) 0px -5px 12px inset;
  box-shadow: rgba(255, 255, 255, 0.4) 0 1px,
    black 0px 1px 3px inset,
    rgba(0, 0, 0, 0.4) 0px -5px 12px inset;
  text-shadow: rgba(255, 255, 255, 0.3) 0px 1px;
}
```

Думаю, вы уже догадались, что этот фрагмент определяет то, как будет выглядеть наша кнопка в момент нажатия. Мы просто изменяем значения некоторых параметров, чтобы добиться примерно такого результата:



Ну, вот и всё: наша кнопка готова.

ЗАКЛЮЧЕНИЕ

Напоследок, приведу несколько замечательных примеров, в полной мере демонстрирующих все достоинства CSS3.

Модель iPhone



Первый пример, созданный Джефом Баттертоном, – точная копия iPhone, созданная без применения графических элементов, исключительно средствами CSS3. К сожалению, работа Джефа рассчитана исключительно для просмотра в браузерах, основанных на движке Webkit. Тем не менее, вы всегда можете с успехом реализовать то же самое и для Firefox!

Иконки в стиле iPhone



Другой, не менее замечательный пример был создан Луисом Харбо. Это репродукция некоторых иконок в стиле iPhone. Как и предыдущий, этот пример также корректно отображается исключительно браузерами Safari и Chrome.

Надеюсь, из этой статьи вам удалось почерпнуть крупицу новых полезных знаний. Желаю успехов в освоении и практическом применении CSS3!

Читайте также: [HTML-сниппеты: избавляемся от рутины](#)
[Мастерство css-кодирования: начало](#)
[10 HTML-тегов, о которых часто забывают новички](#)
[Мастерство css-кодирования: меню с hover-эффектом](#)

Понравилась статья?
 Добавьте в закладки
 или поделитесь с друзьями:

Теги: [css \(9\)](#), [css3 \(3\)](#), [html \(6\)](#), [xhtml \(4\)](#)

Комментарии

[В начало ↑](#)



I love reading these articles because they're short but inftrmraoive.

[Habubakar](#)

06:41 24.08.13

#1 [Цитировать](#)[Hamza](#)

06:26 25.08.13

How neat! Is it really this [sipeml](#)? You make it look easy.

#2 [Цитировать](#)[Jhon](#)

13:02 26.08.13

Kudos! What a neat way of thiinnkg about it.

#3 [Цитировать](#)[Ahmad](#)

17:02 26.08.13

Created the greatest arsclel, you have. <http://emujfpk.com> zuoaiz
[link=<http://uxlyvokwe.com>]uxlyvokwe[/link]

#4 [Цитировать](#)[Oksana](#)

12:15 27.08.13

When you think about it, that's got to be the right [anwres](#).

#5 [Цитировать](#)[Alex](#)

19:30 27.08.13

You've hit the ball out the park! Inecbdirle! <http://scnukjcw.com> fvfzhag
[link=<http://zvkbxqe.com>]zvkbxqe[/link]

#6 [Цитировать](#)[Diana](#)

00:50 29.08.13

It's always a relief when someone with obvious [extispere](#) answers. Thanks!

#7 [Цитировать](#)

[Добавить комментарий](#)[В начало ↑](#)

Имя

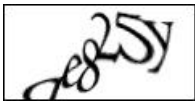
Email

Комментарий

 <i> quote code @ <a> Close All

☐ Сообщать о последующих комментариях

Пожалуйста, введите в следующее поле символы, указанные на картинке (этим вы докажете, что не являетесь спам-ботом):



ОЧЕНЬ ПОЛЕЗНЫЕ РЕСУРСЫ

sohtanaka.com

Web Development & Design

hv-designs.co.uk

Tutorials, Resources and Photoshop

psd.tutsplus.com

PSD Tutorials and much more

designm.ag

Community for designers

НЕСКОЛЬКО СЛУЧАЙНЫХ ПОСТОВ

Мастерство css-кодирования: начало /
15.03.10Изящное переключение стилей с
помощью jQuery и CSS / 29.06.10jQuery в помощь: Accordion menu /
24.05.10Установка Python в GIMP для Windows
/ 02.06.10

САЙТ-ИНФО

Материалы сайта представляют собой локализацию на русский язык зарубежных статей, посвящённых веб-разработке, а также русскоязычные руководства по использованию сценариев, сервисов и программного обеспечения.