



House Made Sausage from Prairie Grass Cafe, Northbrook,
Kurman Communications, Inc

<http://flickr.com/kurmanphotos/7847424816>



Cats that Webchick Is Herding, Kathleen Murtagh

<http://flickr.com/ceardach/4549876293>



Stone Art, Josef Stuefer

<http://flickr.com/josefstuefer/5982121>



A Field of Tulips, Roman Boed

<http://flickr.com/romanboed/867231576>



Resting in the Sunshine, Steve Wilson

<http://flickr.com/pokerbrit/10780890983>



Naxos Island, Greece, Chris Hutchison

<http://flickr.com/employtheskinnyboy/3904743709>

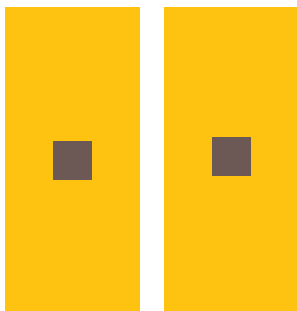


Рис. 1.9. На первом прямоугольнике коричневый квадратик математически выровнен по вертикали, но нам так не кажется. На втором прямоугольнике квадратик находится чуть выше геометрического центра, но человеческий глаз видит его так, словно он находится точно в центре



Рис. 1.10. Кажется, что круг меньше, но ограничивающая его рамка по размеру совпадает с квадратом



Рис. 2.20. Наша отправная точка

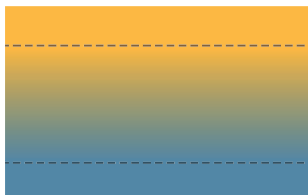


Рис. 2.21. Теперь градиент занимает 60% общей высоты элемента, а оставшаяся часть заполнена сплошными цветами; границы перехода цвета показаны пунктирными линиями

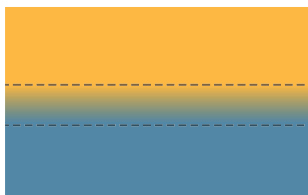


Рис. 2.22. Теперь градиент занимает только 20% общей высоты элемента, а оставшиеся части заполнены сплошными цветами; границы перехода цвета показаны пунктирными линиями



Рис. 2.23. Обе границы перехода цвета сейчас находятся на отметке 50%

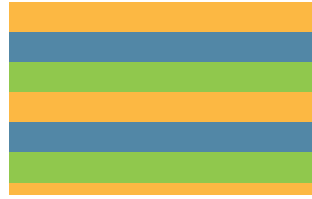


Рис. 2.27. Полосы трех цветов



Рис. 2.24. Наш сгенерированный фон без повторения



Рис. 2.28. Наши вертикальные полосы. *Вверху:* заполненные вертикальными полосами без повторения. *Внизу:* повторяющиеся полосы



Рис. 2.25. Горизонтальные полосы — итоговый результат



Рис. 2.26. Полосы неравной ширины



Рис. 2.38. Моя галерея узоров, сделанных с помощью CSS3 (вы можете найти ее по адресу <http://lea.verou.me/css3patterns>), показывает, какие возможности обеспечивали градиенты CSS еще в 2011 году. В 2011–2012 годах на эту страницу ссылались почти в каждой статье и книге, в которых заходила речь о градиентах CSS, а также упоминали в выступлениях на множестве тематических конференций. Несколько производителей браузеров использовали ее для тонкой регулировки своих реализаций градиентов CSS. Однако далеко не все показанные узоры уместно использовать на реальных веб-сайтах. Некоторые из них приведены только для того, чтобы продемонстрировать возможности CSS, а код, необходимый для их создания, чрезвычайно объемен и полон повторов. Для подобных ситуаций намного лучше подходит формат SVG. Несколько примеров SVG-узоров вы найдете на <http://philbit.com/svgpatterns>; этот веб-сайт был создан в качестве ответа галерее узоров CSS

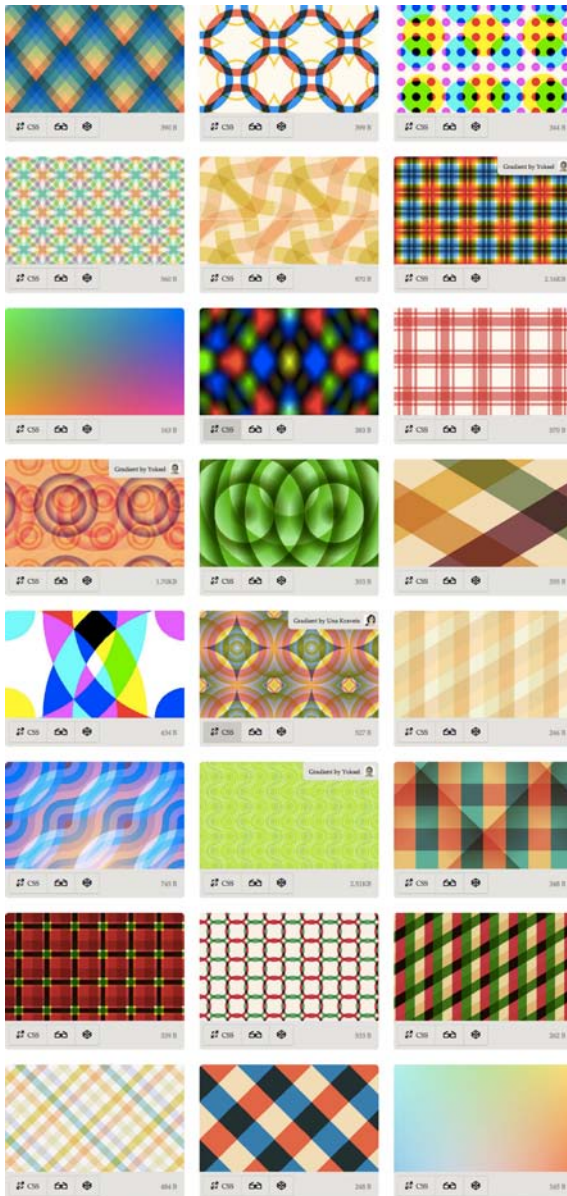


Рис. 2.51. Сочетание этих техник с **режимами смешивания** (<http://w3.org/TR/compositing-1>), когда значения `background-blend-mode`, отличные от `normal`, используются для некоторых (или даже всех) слоев, составляющих фоновое изображение, может дать очень интересные результаты, что подтверждает эта галерея узоров Беннетта Филя (<http://bennettfeely.com/gradients>). Для большинства из этих узоров используется только режим смешивания `multiply`, но и другие значения, такие как `overlay`, `screen` и `difference`, также могут оказаться очень полезными



Рис. 2.53. Наша первая попытка создать псевдослучайные полосы, где все цвета генерируются одним и тем же линейным градиентом

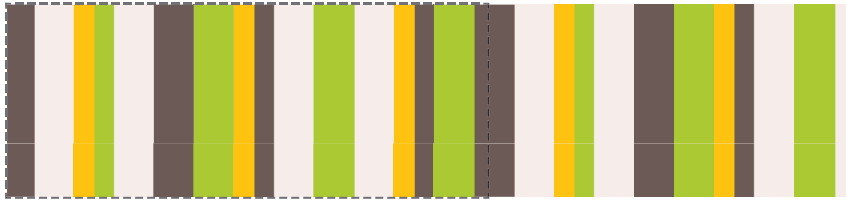


Рис. 2.54. Наша вторая попытка, включающая наложение друг на друга разных градиентов с разным размером фона; (воспринимаемая взглядом) повторяющаяся плитка обозначена пунктирными линиями



Рис. 2.55. Финальный вариант полосок, где для создания лучшего впечатления хаотичности используются простые числа

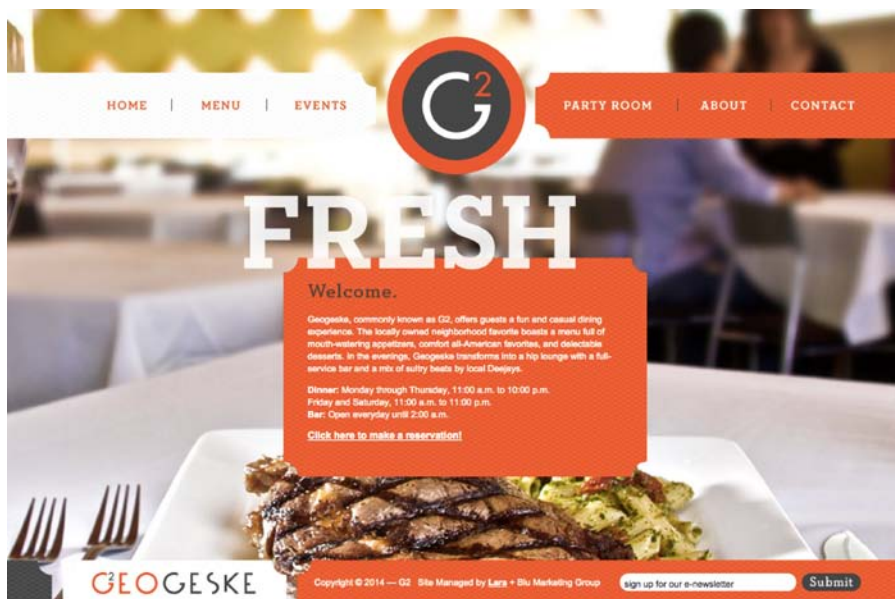


Рис. 3.30. Превосходный пример использования искривленных срезаемых углов на веб-сайте <http://g2geogeske.com>; дизайнер сделал их центральным элементом оформления: они присутствуют в навигации, в содержимом и даже в нижнем колонтитуле

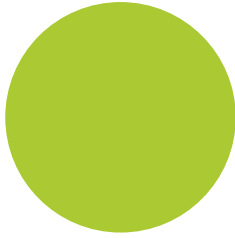


Рис. 3.50. Наша отправная точка (или секторная диаграмма, показывающая 0%)



Рис. 3.51. Окрашивание правой части нашего круга в коричневый цвет с помощью простого линейного градиента

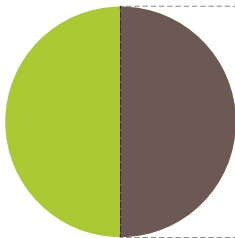


Рис. 3.52. Псевдоэлемент, который будет выполнять функцию маски, обозначен здесь пунктирными линиями

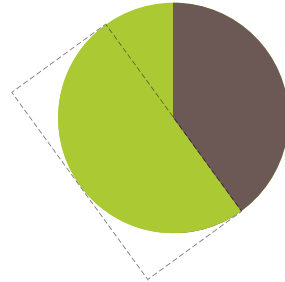
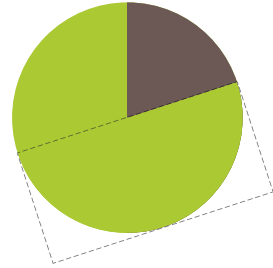
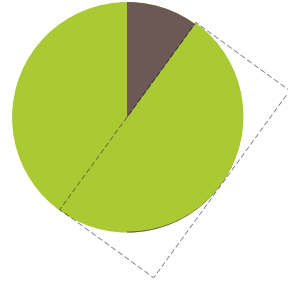


Рис. 3.53. Наша простая секторная диаграмма, отображающая разные процентные значения. Сверху вниз: 10% (36deg или .1turn), 20% (72deg или .2turn), 40% (144deg или .4turn) 🤖

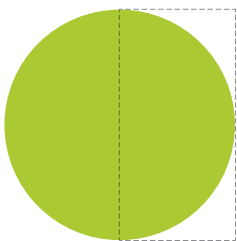
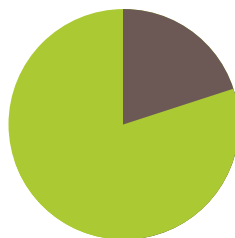
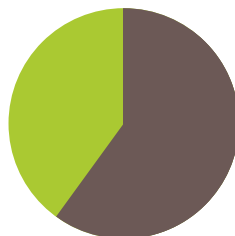


Рис. 3.54. Наш псевдоэлемент (обозначенный здесь пунктирным контуром) после того, как мы закончили определение его стилей



20%



60%

Рис. 3.57. Наш текст до того, как мы его спрячем

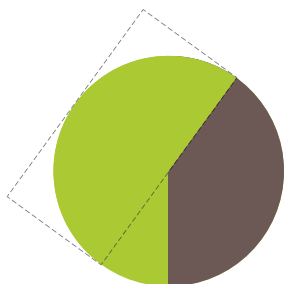


Рис. 3.55. Наша секторная диаграмма ломается на значениях, превышающих 50% (показанное здесь соответствует значению 60%)

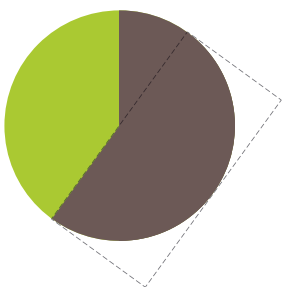


Рис. 3.56. Наша исправленная диаграмма и значение 60%



Рис. 3.58. Наша отправная точка: определенный с помощью SVG зеленый круг с жирной обводкой цвета #655



Рис. 3.59. Простая пунктирная обводка, созданная с помощью stroke-dasharray

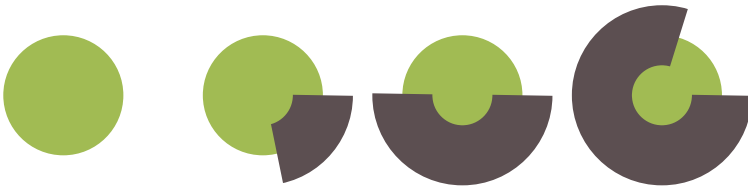


Рис. 3.60. Несколько значений stroke-dasharray и соответствующий визуальный результат.
Слева направо: 0 189 40 189 95 189 150 189



Рис. 3.61. Наша SVG-графика начинает напоминать секторную диаграмму

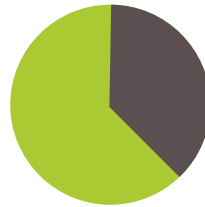


Рис. 3.62. Готовая секторная диаграмма, созданная с помощью SVG

Our awesome speakers



Angelina Fabbro
@angelinamagnum



Antoine Butler
@aebsr



Jenn Schiffer
@jenschiffer



Lea Verou
@leaverou



Nicole Sullivan
@stubbomella



Patrick Hamann
@patrickhamann

Рис. 4.9. На веб-сайте CSSConf 2014 этот эффект используется для оформления фотографий лекторов. Полноцветное изображение показывается при наведении указателя мыши и переводе фокуса на выбранный элемент



Рис. 4.10. *Наверху:* исходное изображение. *Внизу:* изображение после применения фильтра `sepia()`



Рис. 4.11. Наше изображение после добавления фильтра `saturate()`



Рис. 4.12. Наше изображение после добавления третьего фильтра, `hue-rotate()`



Рис. 4.13. Сравнение метода на основе фильтров (*наверху*) и метода, основанного на режимах смешивания (*внизу*)

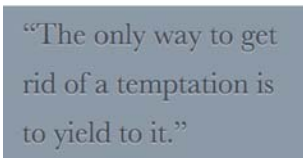
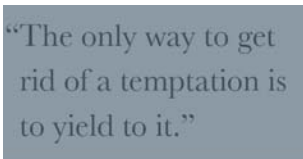


Рис. 5.28. Эффект вдавленного текста, реализованный для темного шрифта на более светлом фоне (*вверху*: до, *внизу*: после)

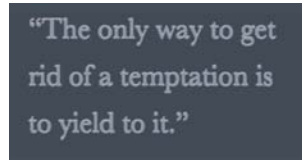


Рис. 5.29. Эффект вдавленного текста сломался: мы попробовали применить предыдущее решение к тексту, цвет которого светлее цвета фона

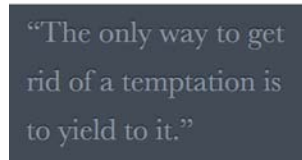
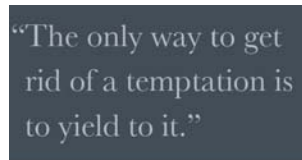


Рис. 5.30. Эффект вдавленного текста при использовании светлого цвета шрифта на темном фоне (*вверху*: до, *внизу*: после)



Рис. 7.7. Элементы управления становятся все меньше с увеличением количества цветов и сокращением количества свободного пространства. Обратите внимание на особое оформление в случае, когда цвет только один: кнопка удаления скрыта.
 Цвета взяты из палитр Adobe Color (<http://color.adobe.com>):
 Agave (<http://color.adobe.com/agave-color-theme-387108>)
 Sushi Maki (<http://color.adobe.com/Sushi-Maki-colortheme-350205>)

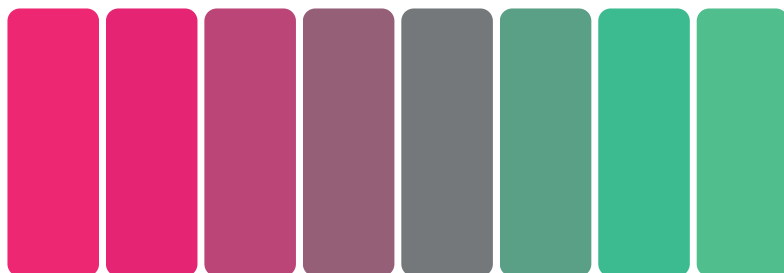


Рис. 8.12. Эластичный цветовой переход от $\text{rgb}(100\%, 0\%, 40\%)$ к цвету gray ($\text{rgb}(50\%, 50\%, 50\%)$) с функцией расчета времени cubic-bezier(.25,.1,.2,3). Каждая координата RGB интерполируется по отдельности, поэтому в результате в составе перехода оказываются странные цвета вроде $\text{rgb}(0\%, 100\%, 60\%)$. Проверьте на <http://play.csssecrets.io/elastic-color>