

Рис. 4.6. Элементы, стилизованные с помощью CSS, с которыми использование `box-shadow` теряет всякий смысл (здесь применяется значение свойства `box-shadow`, равное `2px 2px 10px rgba(0,0,0,.5)`)

Результаты тщетных попыток применить `box-shadow` в некоторых из перечисленных ситуаций показаны на рис. 4.6. Существует ли решение для подобных случаев или нам придется вообще отказаться от использования теней?

Решение

Спецификация **Filter Effects** (<http://w3.org/TR/filter-effects>) предлагает решение данной проблемы в форме нового свойства `filter`, позаимствованного из формата SVG. Однако несмотря на то что фильтры CSS — это, по сути, те же самые **фильтры SVG**, для их использования **никакого знания SVG не требуется**. Они определяются посредством нескольких удобных функций, таких как `blur()`, `grayscale()` и — барабанная дробь — `drop-shadow()`! Можно даже составить последовательность из нескольких фильтров, если того требует ситуация, разделив их пробелами, например так:

```
filter: blur() grayscale() drop-shadow();
```

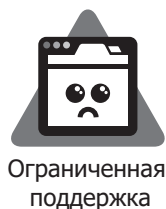
Фильтр `drop-shadow()` принимает те же параметры, что и базовое свойство `box-shadow`, то есть без радиуса размазывания, без ключевого слова `inset`, без разделенных запятыми определений нескольких теней. Например, вместо:

```
box-shadow: 2px 2px 10px rgba(0,0,0,.5);
```

мы бы написали:

```
filter: drop-shadow(2px 2px 10px rgba(0,0,0,.5));
```

Результат применения этого фильтра `drop-shadow()` к элементам с рис. 4.6 демонстрируется на рис. 4.7.



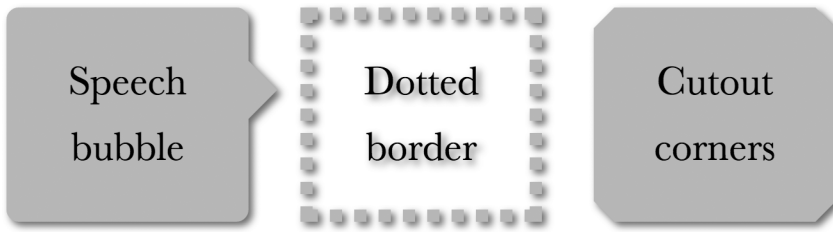


Рис. 4.7. Результат использования фильтра `drop-shadow()` с элементами с рис. 4.6

! Поскольку алгоритмы размытия могут быть разными, вам может потребоваться отрегулировать значение размытия в зависимости от потребностей в конкретной задаче.

Лучше всего в фильтрах CSS то, что они обеспечивают **изящные резервные решения**: когда фильтры не поддерживаются, ничего не ломается, просто никакой эффект не применяется. Вы можете добиться **чуть лучшей поддержки браузерами**, используя **заодно фильтр SVG**, если перед вами стоит задача любыми способами заставить этот эффект работать в как можно большем количестве браузеров. Соответствующие фильтры SVG для каждой функции

фильтрации вы найдете в спецификации **Filter Effects** (<http://w3.org/TR/filter-effects>). Можно одновременно использовать и фильтр SVG, и упрощенный аналог на CSS, позволяя каскадным стилям самим определять победителя:

```
filter: url(drop-shadow.svg#drop-shadow);
filter: drop-shadow(2px 2px 10px rgba(0,0,0,.5));
```

К сожалению, если SVG-фильтр содержится в отдельном файле, то он не поддается такой же простой настройке, как приятная, удобная в использовании функция прямо в CSS-коде, а строковая функция, в свою очередь, захламляет код. В файле параметры фиксированы, и иметь несколько файлов на случай, если нам понадобятся слегка различающиеся тени, непрактично. Мы могли бы использовать URI данных (что заодно сэкономило бы нам несколько запросов HTTP), но они также приводят к увеличению размера файла. Поскольку фильтр SVG используется для обеспечения обходного решения, имеет смысл использовать один или два варианта, даже для немного отличающихся фильтров `drop-shadow()`.

Кроме того, не следует забывать, что **отбрасывать тень будет любая непрозрачная область**, независимо от того, каким элементом она представлена, — даже текст (на прозрачном фоне), как вы уже видели на рис. 4.7. Возможно, вы думаете, что отменить этот эффект можно с помощью `text-shadow: none;`, но `text-shadow` — это отдельное свойство, не способное компенсировать влияние фильтра `drop-shadow()` на текст. А если вы используете `text-shadow` для создания

настоящей тени текста, то благодаря фильтру `drop-shadow()` у этой тени также появится своя тень, то есть **вы создадите тень тени!** Взгляните на следующий пример CSS-кода (и простите за безвкусный результат — я всего лишь пытаюсь продемонстрировать всю дикость этой проблемы):

```
color: deeppink;
border: 2px solid;
text-shadow: .1em .2em yellow;
filter: drop-shadow(.05em .05em .1em gray);
```

Пример визуализации этого кода вы видите на рис. 4.8: здесь показана как тень `text-shadow`, так и отбрасываемая ею тень `drop-shadow()`.

ПОПРОБУЙТЕ САМИ!

<http://play.csssecrets.io/drop-shadow>



Рис. 4.8. Тени `text-shadow` также отбрасывают тень сквозь фильтр `drop-shadow()`

СВЯЗАННЫЕ СПЕЦИФИКАЦИИ

Filter Effects: <http://w3.org/TR/filter-effects>