Эффект "матового стекла" посредством CSS-фильтров.

Данный пост любезно предоставлен Биром Тревисом (Bear Travis), занимающим должность Web Standards Engineerв компании Adobe. Мне нравится, что компания Adobe продвигает новые веб-технологии для дизайна, причем подходит к этому очень ответственно. CSS-фильтры являются тому хорошим примером.

КомпанияAdobeзнала, что фильтры будут востребованы, потому как проложили им путь посредством Photoshop. В web-инструментарий CSS-фильтры попали с уже осмысленным синтаксисом, к тому же "адобовцы" помогли разработчикам со спецификацией и внедрением в браузеры.

На данный момент мы можем видеть работу фильтров в стабильных версиях большинства браузеров. Аллилуя.

Ниже следует краткое руководство, в котором Бир поведает нам об их использовании.

Пока такие фильтры, как "контраст", "насыщенность" и "размытие" существовали только лишь в графических редакторах, в web-технологиях мы видели их в уже готовых изображениях, к которым эти фильтры применялись заранее. Как только же браузеры стали включать в себя возможности работы с фильтрами, мы получили возможность применять их к любым элементам веб-документа. В этой статье мы разберем один из таких эффектов на примере эффекта матового стекла, и рассмотрим, насколько более гибко и удобно можно использовать CSS-фильтры в сравнении с использованием статических изображений.

OldSchool: Эффект матового стекла из нескольких картинок

Использовать эффект матового стекла начали в web-дизайне довольно давно. Мы публиковали статью на CSS-Tricks в далеком 2008. Эффект реализовывался относительно просто: имелось два изображения – оригинальное и обработанное (матовое, "заблёренное" (blur) белым оттенком и размытое, с повышенной яркостью). Затем оригинал применялся фоном к тегу body, а обработанное изображение – задавалось фоном для контейнера div. После чего, для div’а задавались размеры, граница и позиционирование, так что в итоге мы получали нужный эффект.

В нашем примере картинка будет наезжать поверх содержимого, тем самым накладывая матовый эффект на фон.

<iframeid="cp_embed_40cd4258f2d72a60f37a5e2f47124b7e" src="//codepen.io/anon/embed/40cd4258f2d72a60f37a5e2f47124b7e?height=300&theme-id=1&slug-hash=40cd4258f2d72a60f37a5e2f47124b7e&default-tab=result" scrolling="no" frameborder="0" height="300" allowtransparency="true" class="cp_embed_iframe" style="width: 100%; overflow: hidden;"></iframe>

HTML

Разметка относительно проста. У нас всего один тегarticleвнутри которого находится все содержимое.

```
<article class="glass down">

<h1>Птица-Секретарь</h1>

<p>наводитсправки ...</p>

</article>
```

CSS

Для начала подгоняем все к размерам отображаемой области. Затем совмещаем матируемую версию фона с вершинойоригинального фона. В конце добавляем белый оттенок. Переполнение спрятано, дабы предотвратить появление скроллинга и чтобы закрепить эффект за элементом .glass

```
html, body, .glass {width: 100%;height: 100%;overflow: hidden;}

body{background-image:url('pelican.jpg');background-size: cover;}

.glass::before {display:block;width: 100%;height: 100%;background-image:url('pelican-blurry.jpg');background-size:cover;content:' ';opacity: 0.4;}

.glass{background-color: white;}
```

Этот CSS-код создает наше матированное и осветленное наложение. Нам также понадобится кнопка, которая будет сдвигать наложение к низу страницы, высвобождая пространство для отображения самого текста. Так как матированное изображение является зависимым от наложения, нам также нужно сдвинуть и его для выравнивания с фоном. Так как демонстрационный режим использует переходы, я предпочел CSS-

преобразования свойству background-attachment, так как CSS-преобразования могут работать с помощью аппаратного ускорения.

```
.glass.down{transform:translateY(100%) translateY(-7rem);}

.glass.down::before {transform:translateY(-100%) translateY(7rem);}

.glass.up, .glass.up::before {transform:translateY(0);}
```

На заметку

Если вам понадобятся дополнительные пояснения, я сделал версию-конструктор.

Эта техника проста и поддерживается большинством браузеров. Также я немного приукрасил демонстрационный пример с помощью переходов, сгенерированного контента, прозрачности, преобразований и свойства background-size - все они поддерживаются большинством браузеров вплоть до IE9, за исключением OperaMini.

NewSchool: Матовое стекло с помощью фильтров

Техника наложения двойных изображений требует одновременно иметь заблёренное изображение и оригинал, что может доставить кучу неудобств если понадобится применять эффект ко множеству картинок. К примеру, отзывчивые дизайны могут потребовать менять картинки на разных разрешениях экрана. Либо же это могут быть макеты, которые подбирают изображения динамически (например, различные изображения в заголовках к разным блогам).

Для этих случаев будет правильным использовать только исходные изображения. И далее матировать их. Вот здесь и приходят на помощь CSS-фильтры. Они позволяют применять эффект размытия посредством самого браузера, используя непосредственно CSS-фильтр.

CSS

Мы настраиваем CSSдля наложения матирования на оригинальное изображение с применением фильтра blur.

```
.glass::before {background-image:url('pelican-blurry.jpg');}

.glass::before {background-image:url('pelican.jpg');filter: blur(5px);}
```

Demo

Ложка дегтя

Проще пареной репы, верно? Увы, но на данный момент технология CSS-фильтровдовольно сыра. Это значит, что поддержка *может быть* встроена разработчиками в браузер, но не означает, что поддерживается всеми браузерами.

Однако, есть фильтры в SVG, которые появились довольно давно и применение SVG-фильтров в HTML посредством CSS поддерживается гораздо бОльшим числом браузеров. Вы можете добавить этот вариант как запасной, на тот случай, когда CSS-фильтры не будут работать. Демонстрационный режим ниже показывает как это сделать.

Для того, чтобы добавить SVG-фильтр, мы включаем тег <svg> в наш html-код и ссылаемся на фильтр с помощью тега url().

```
<svgxmlns="http://www.w3.org/2000/svg" version="1.1">

<defs>

<filter id="blur">

<feGaussianBlurstdDeviation="5" />

</filter>

</defs>

</svg>

.glass::before {background-image:url('pelican.jpg');

/* резервныйвариантсиспользованиемSVG-фильтров */

filter:url('#blur');

filter: blur(5px);}
```

Также не стоит исключать тот вариант, когда ни CSS, ни SVG-фильтры не поддерживаются.

В таком случае пользователь увидит подсвеченный текст на оттененном, но не матированном фоне, что, собственно, не так уж и плохо.

Заключение.

Фильтры позволяют нам использовать эффекты непосредственно силами браузера, что ранее было доступно только в графических редакторах. В сравнении со статическими

изображениями фильтры проще использовать и легче изменять. Фильтры прекрасно работают в текущих версиях Chrome, Safari, и Opera, а также в Firefox. Об IE пока что умолчим. CSS-фильтры стоит использовать уже сейчас, предварительно продумав резервные варианты на случаи, когда "сил" браузера не хватит на поддержку фильтров.