

Фильтры и преобразования. Создание мультимедийных эффектов

Фильтры и преобразования — это мультимедийные эффекты, которые можно применить к элементу Web-страницы. *Фильтрами* называют статические эффекты (изменение уровня прозрачности элемента, добавление тени, размытие и др.), а *преобразованиями* — динамические эффекты (всевозможные всплывания, выплывания и др.).

Основные понятия

Создать фильтр можно в определении стиля элемента с помощью атрибута `filter`:

```
filter:<Имя фильтра>(<Свойства фильтра>)
```

или

```
filter:progid:DXImageTransform.Microsoft.<Имя фильтра>(<Свойства фильтра>)
```

Создать фильтр можно из скрипта:

```
<id элемента>.style.filter =  
"progid:DXImageTransform.Microsoft.<Имя фильтра>(<Свойства фильтра>)"
```

Доступ к фильтру из скрипта осуществляется с помощью коллекции `filters`. Например, задать степень прозрачности элемента можно одним из способов:

```
img1.filters("alpha").opacity = 100;  
img1.filters(0).opacity = 100;  
img1.filters["alpha"].opacity = 100;  
img1.filters.alpha.opacity = 100;  
img1.filters.item("alpha").opacity = 100;  
img1.filters.item(0).opacity = 100;
```

Не все такие обращения работают со всеми фильтрами. Универсальным можно назвать только доступ по индексу в коллекции:

```
img1.filters(0).opacity = 100;  
img1.filters.item(0).opacity = 100;
```

Фильтр *Alpha*. Задание уровня прозрачности элемента

Фильтр Alpha задает уровень прозрачности элемента страницы.

Свойства фильтра:

- ☐ `opacity` задает начальный уровень градиентной прозрачности. Значение может быть в пределах от 0 (полная прозрачность) до 100 (полностью непрозрачен);
- ☐ `finishOpacity` устанавливает конечный уровень градиентной прозрачности. Значение может быть в пределах от 0 (полная прозрачность) до 100 (полностью непрозрачен). Используется в случае, если задан тип градиента;

- ❑ `style` определяет тип градиентной прозрачности:
 - 0 — нет градиента (значение по умолчанию);
 - 1 — линейный градиент;
 - 2 — круговой градиент;
 - 3 — прямоугольный градиент;
- ❑ `startX` и `startY` указывают горизонтальную и вертикальную координаты позиции, с которой начинается область градиентной прозрачности;
- ❑ `finishX` и `finishY` устанавливают горизонтальную и вертикальную координаты позиции, в которой заканчивается область градиентной прозрачности;
- ❑ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 1 приведен пример использования фильтра Alpha.

Листинг 1. Фильтр Alpha

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <script type="text/javascript">
    <!--
    function f_over() {
      i_img.filters.alpha.opacity=100;
    }
    function f_out() {
      i_img.filters.alpha.opacity=50;
    }
    //-->
  </script>
</head>
<body>
  
</body>
</html>
```

Фильтр *AlphaImageLoader*. Отображение картинок между фоном и содержимым элемента

Фильтр *AlphaImageLoader* отображает графическое изображение внутри элемента страницы между его фоном и содержимым.

Свойства фильтра:

- ☐ `src` — URL-адрес файла с изображением;
- ☐ `sizingMethod` — способ размещения изображения в пределах элемента страницы:
 - `crop` — обрезание изображения;
 - `image` — уменьшение или увеличение самого элемента страницы (значение по умолчанию);
 - `scale` — уменьшение или увеличение изображения;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

Если при разработке Web-ресурса используются 24-битовые графические файлы формата PNG с частично прозрачными областями, то в Internet Explorer они отображаются некорректно. Вместо прозрачных элементов графики Web-браузер отображает блекло-серый фон, закрывающий фон страницы. Решение этой проблемы заключается в использовании фильтра *AlphaImageLoader* (листинг 2).

Листинг 2. Фильтр *AlphaImageLoader*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>
<div id="div1" style="position:relative; height:240px; width:120px;
filter:progid:DXImageTransform.Microsoft.AlphaImageLoader(src='foto.png',
sizingMethod='image');">
</div>
</body>
</html>
```

Фильтр *BasicImage*. Черно-белые, "рентгеновские" и инвертированные элементы

Фильтр `BasicImage` делает элемент страницы черно-белым, имитирует просвечивание "рентгеновскими" лучами или отображает инвертированными цветами. Можно также задать угол поворота и прозрачность.

Свойства фильтра:

- ☐ `grayScale` задает, будет ли элемент черно-белым или цветным. Он может принимать такие значения:
 - 0 — элемент страницы отображается цветным (значение по умолчанию);
 - 1 — элемент страницы отображается черно-белым;
- ☐ `invert` указывает, будет ли элемент отображаться с инвертированными цветами. Он может принимать следующие значения:
 - 0 — отображается как обычно (значение по умолчанию);
 - 1 — элемент страницы отображается с инвертированными цветами;
- ☐ `maskColor` задает цвет, на который будет заменен прозрачный цвет элемента страницы;
- ☐ `mask` определяет, будет ли прозрачный цвет элемента страницы заменен значением свойства `maskColor`. Он может принимать такие значения:
 - 0 — не будет (значение по умолчанию);
 - 1 — будет заменен;
- ☐ `mirror` указывает, будет ли элемент страницы отображен зеркально, может принимать значения:
 - 0 — как обычно (значение по умолчанию);
 - 1 — элемент страницы будет отображен зеркально;
- ☐ `opacity` задает уровень прозрачности элемента страницы. Значение задается в пределах от 0.0 до 1.0;
- ☐ `rotation` задает поворот элемента страницы, может принимать значения:
 - 0 — нет поворота (значение по умолчанию);
 - 1 — поворот на 90 градусов;
 - 2 — поворот на 180 градусов;
 - 3 — поворот на 270 градусов;
- ☐ `xRay` определяет, будет ли для элемента страницы имитировано просвечивание "рентгеновскими" лучами, может принимать значения:
 - 0 — как обычно (значение по умолчанию);

- 1 — элемент страницы будет просвечен;

☐ `enabled` — разрешает (`true`) или запрещает (`false`) применение фильтра.

Изменим код листинга 1 и вместо фильтра `Alpha` применим фильтр `BasicImage` (листинг 3). Фильтр создадим в скрипте, а первоначальное изображение сделаем черно-белым и полупрозрачным. При наведении курсора мыши изображение станет цветным и полностью непрозрачным.

Листинг 3. Фильтр `BasicImage`

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <script type="text/javascript">
    <!--
    function f_over() {
      i_img.filters(0).opacity=1.0;
      i_img.filters(0).grayScale=0;
    }
    function f_out() {
      i_img.filters(0).opacity=0.5;
      i_img.filters(0).grayScale=1;
    }
    function f_load() {
      i_img.style.filter=
"progid:DXImageTransform.Microsoft.BasicImage(opacity=0.5, grayScale=1)";
    }
    //-->
  </script>
</head>
<body onload="f_load();">

</body>
</html>
```

Фильтр *Blur*. Размытые элементы страницы

Фильтр `Blur` делает элемент страницы размытым.

Свойства фильтра:

- ☐ `pixelRadius` задает размер области размытия. Может принимать значения от 0.0 до 100.0;

- ❑ `shadowOpacity` устанавливает уровень прозрачности тени. Может принимать значения от 0.0 (полная прозрачность) до 1.0 (полная непрозрачность). Значение можно изменить, только если свойство `makeShadow` имеет значение `true`;
- ❑ `makeShadow` указывает, будет ли элемент страницы отображаться как тень:
 - `true` — будет отображаться как тень;
 - `false` — как обычно (значение по умолчанию);
- ❑ `enabled` — разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 4 приведен пример использования фильтра `Blur`.

Листинг 4. Фильтр `Blur`

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
  <script type="text/javascript">
  <!--
function f_load() {
  i_div.style.filter=
"progid:DXImageTransform.Microsoft.Blur(shadowOpacity=0.7, makeShadow=true,
pixelRadius=2.0) ";
}
function f_over() {
  i_div.filters.item(0).shadowOpacity=1.0;
  i_div.filters.item(0).pixelRadius=0.0;
}
function f_out() {
  i_div.filters.item(0).shadowOpacity=0.7;
  i_div.filters.item(0).pixelRadius=2.0;
}
  //-->
</script>
</head>
<body onload="f_load();" >
<div id="i_div" style="width:90%;" onmouseover="f_over();"
onmouseout="f_out();" >Размытый текст
</div>
Обычный текст
</body>
</html>
```

Фильтр *Chroma*. Отображение отдельного цвета изображения прозрачным

Фильтр *Chroma* делает прозрачным отдельный цвет элемента (полностью удаляет указанный цвет из изображения).

Свойства фильтра:

- ☐ `color` задает цвет, который будет прозрачным;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

Создайте картинку 120×240, залейте ее одним цветом #000080 (navy) и сохраните в формате gif (foto1.gif). Если сделать картинку другим цветом, то эффекта от примера не будет. Чтобы были видны контуры изображения, параметр `border` тега `` имеет значение 2. При загрузке видны только контуры изображения и текст за ним. Если навести курсор на изображение, то оно полностью появится и частично закроет текст. Если курсор убрать, то изображение опять исчезнет (листинг 5).

Листинг 5. Фильтр *Chroma*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
  <script type="text/javascript">
  <!--
function f_load() {
  i_img.style.filter=
    "progid:DXImageTransform.Microsoft.Chroma (Color=#000080)";
}
function f_over() {
  i_img.filters.item(0).enabled=false;
}
function f_out() {
  i_img.filters.item(0).enabled=true;
}
  //-->
</script>
</head>
<body onload="f_load();" >Текст за изображением

</body>
```

</html>

Фильтр *Compositor*. Выводим изображение на текст

Фильтр *Compositor* объединяет два элемента страницы и выводит результат.

Свойства фильтра:

- ☐ `enabled` — разрешает (`true`) или запрещает (`false`) применение фильтра;
- ☐ `Function` — функция преобразования. Может принимать следующие значения:
 - 0 — не выполняется никакой операции (значение по умолчанию);
 - 1 — показывается минимальный цвет от обоих наборов входа;
 - 2 — показывается максимальный цвет от обоих наборов входа;
 - 3 — отображается только вход А;
 - 4 — показывается вход А поверх входа В. Весь вход А видим, а вход В виден через прозрачные области входа А;
 - 5 — показываются все части входа А, которые содержатся во входе В. Те области входа А, где вход В прозрачен, также не показываются;
 - 6 — отображаются все части входа А, расположенные в местах, где вход В прозрачен. Никакая часть входа В не показывается;
 - 7 — вход А помещается поверх входа В, а его прозрачность рассчитывается исходя из прозрачности входа В;
 - 8 — вход А вычитается из входа В;
 - 9 — во вход А добавляется вход В;
 - 10 — показываются те пиксели каждого входа, где два изображения не накладываются;
 - 19 — отображается только вход В;
 - 20 — показывается вход В поверх входа А. Весь вход В видим, а вход А виден через прозрачные области входа В;
 - 21 — показываются все части входа В, в которых вход А непрозрачен;
 - 22 — отображаются все части входа В, в которых вход А прозрачен. Никакая часть входа А не показывается;
 - 23 — отображается вход В поверх входа А, а его прозрачность рассчитывается исходя из прозрачности входа А;
 - 24 — вход В вычитается из входа А;
 - 25 — во вход В добавляется вход А.

Чтобы создать статический фильтр с переходом, нужно выполнить следующие действия.

1. Выбрать значение функции.
2. Создать вход A.
3. Применить метод `apply()`.
4. Создать вход B. Изменить свойство элемента (например, `visibility`, `innerText`, `backgroundColor` или `border`).
5. Применить метод `play()`.

С помощью фильтра `Compositor` можно поменять однотонный цвет текста на текстуру (листинг 6). Для этого создадим изображение с текстурой в графическом редакторе и сохраним под названием `foto1.gif`. Создадим текстовый фрагмент и применим к нему фильтр `Compositor`.

Листинг 6. Фильтр `Compositor`

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
  <script type="text/javascript">
  <!--
function f_start() {
  i_div.filters.item("DXImageTransform.Microsoft.Compositor").Function=
    i_sel.options(i_sel.selectedIndex).value;
  i_div.innerHTML =
    "<div align='center' style='font-size: 32pt'>Пример</div>";
  i_div.filters.item(0).apply();
  i_div.innerHTML = i_img.innerHTML;
  i_div.filters.item(0).play();
}
  <!-->
</script>
</head>
<body>
<br><br><br>
<b>Function:</b>
<select id="i_sel" onchange="f_start();">
  <option value=0>0</option>
  <option value=1>1</option>
  <option value=2>2</option>
  <option value=3>3</option>
  <option value=4>4</option>
  <option value=5>5</option>
```

```

        <option value=6>6</option>
        <option value=7>7</option>
        <option value=8>8</option>
        <option value=9>9</option>
        <option value=10>10</option>
        <option value=19>19</option>
        <option value=20>20</option>
        <option value=21>21</option>
        <option value=22>22</option>
        <option value=23>23</option>
        <option value=24>24</option>
        <option value=25>25</option>
    </select>
    <div id="i_div" style="background-color: #FFFFFF;
    filter:progid:DXImageTransform.Microsoft.Compositor();
    position: absolute; top: 0px; left: 0px; width:240px; height:60px;">
    <div align="center" style="font-size: 32pt">Пример</div>
    </div>
    <div id="i_img" style="display: none;">
    
    </div>
</body>
</html>

```

Если в списке выбрать значение 2, 9 или 25, то мы получим текстовый фрагмент со структурой изображения.

Фильтр *DropShadow*. Тени

Фильтр DropShadow заставляет элемент страницы отбросить тень, которая отображается отдельно от самого элемента.

Свойства фильтра:

- ☐ color — цвет тени;
- ☐ offX и offY — горизонтальное и вертикальное смещения тени, которые могут принимать отрицательные значения;
- ☐ positive определяет режим работы фильтра и может принимать два значения:
 - true — тень создается из прозрачных пикселей элемента страницы (значение по умолчанию);
 - false — тень создается из непрозрачных пикселей элемента страницы;
- ☐ enabled разрешает (true) или запрещает (false) применение фильтра.

В листинге 7 приведен пример использования фильтра DropShadow.

Листинг 7. Фильтр DropShadow

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
<div style="background-color:silver;
filter:progid:DXImageTransform.Microsoft.DropShadow(color=gray, offX=4, offY=4);
position:absolute; top:10px; left:10px; width:240px; height:60px;">
</div>
</body>
</html>
```

Фильтр *Emboss*. Выпуклые элементы страницы

Фильтр Emboss делает элемент страницы выпуклым.

Свойства фильтра:

- ☐ bias — процентное значение, добавляемое к цвету элемента;
- ☐ enabled разрешает (true) или запрещает (false) применение фильтра.

В листинге 8 приведен пример использования фильтра Emboss.

Листинг 8. Фильтр Emboss

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>
<font style="background-color:silver; font-size:32pt;
filter:progid:DXImageTransform.Microsoft.Emboss(); width:240px;">
Пример
</font>
</body>
</html>
```

Фильтр *Engrave*. Вдавленные элементы страницы

Фильтр Engrave отображает элемент вдавленным в страницу.

Свойства фильтра:

- ☐ `bias` — процентное значение, добавляемое к цвету элемента;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 9 приведен пример использования фильтра `Engrave`.

Листинг 9. Фильтр `Engrave`

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>
<font style="background-color:silver; font-size:32pt;
filter:progid:DXImageTransform.Microsoft.Engrave(); width:240px;">
Пример
</font>
</body>
</html>
```

Фильтр *Glow*. Ореол вокруг элемента страницы

Фильтр `Glow` создает ореол вокруг элемента страницы.

Свойства фильтра:

- ☐ `color` — цвет ореола;
- ☐ `strength` — размер ореола;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 10 приведен пример использования фильтра `Glow`.

Листинг 10. Фильтр `Glow`

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>
<span style="position:absolute; font-size:32pt; top:0;
filter:progid:DXImageTransform.Microsoft.Glow(color=green, strength=5);">
Пример
```

```
</span>
</body>
</html>
```

Фильтр *Gradient*. Градиентная заливка элемента страницы

Фильтр `Gradient` градиентно закрашивает элемент страницы.

Свойства фильтра:

- ☐ `startColor` и `startColorStr` — начальный цвет градиентной заливки;
- ☐ `endColor` и `endColorStr` — конечный цвет градиентной заливки;
- ☐ `gradientType` задает направление градиентной заливки, может принимать значения:
 - 1 — градиентная заливка располагается по горизонтали (по умолчанию);
 - 0 — градиентная заливка располагается по вертикали;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 11 приведен пример использования фильтра `Gradient`.

Листинг 11. Фильтр *Gradient*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>
<div style="position:absolute; font-size:32pt; top:0;
filter:progid:DXImageTransform.Microsoft.Gradient(startColorStr=yellow,
endColorStr=red, gradientType=1);">Пример
</div>
</body>
</html>
```

Фильтр *Light*. Эффект освещенности элемента страницы

Фильтр `Light` создает эффект освещенности элемента страницы.

Методы фильтра:

- ☐ `AddAmbient(<Красный>, <Зеленый>, <Синий>, <Интенсивность>)`
добавляет источник рассеянного света с заданными цветовыми параметрами;

- ❑ `AddCone(<x1>, <y1>, <z1>, <x2>, <y2>, <Красный>, <Зеленый>, <Синий>, <Интенсивность>, <Угол>)` добавляет источник направленного света с заданными цветовыми параметрами. Параметры `x1`, `y1`, `z1` задают координаты источника света, а `x2`, `y2`, `0` — точки, куда направлен свет;
- ❑ `AddPoint(<x>, <y>, <z>, <Красный>, <Зеленый>, <Синий>, <Интенсивность>)` добавляет источник ненаправленного света с заданными цветовыми параметрами. `x`, `y`, `z` — координаты источника света;
- ❑ `ChangeColor(<№>, <Красный>, <Зеленый>, <Синий>, 1|0)` изменяет цвет источника света с заданным номером на указанный цвет. Последний параметр указывает, абсолютное (1) или относительное (0) изменение цвета производится;
- ❑ `ChangeStrength(<№>, <Интенсивность>, 1|0)` изменяет интенсивность света источника с заданным номером. Последний параметр указывает, имеется ли в виду абсолютное (1) или относительное (0) изменение интенсивности;
- ❑ `Clear()` удаляет все источники света;
- ❑ `MoveLight(<№>, <x>, <y>, <z>, 1|0)` перемещает источник света с заданным номером в место с заданными координатами. Последний параметр указывает, будет абсолютное (1) или относительное (0) перемещение.

Компоненты цвета задаются значениями от 0 до 255, а интенсивность от 0 до 100.

Свойство фильтра `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 12 приведен пример использования фильтра `Light`.

Листинг 12. Фильтр `Light`

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>


</body>
</html>
```

Фильтр *MaskFilter*. Отображение прозрачной области элемента заданным цветом

Фильтр `MaskFilter` отображает прозрачный цвет элемента страницы заданным цветом, а все непрозрачные цвета делает прозрачными.

Свойства фильтра:

- ☐ `color` задает цвет, на который изменится прозрачный цвет элемента страницы;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

Фильтр *Matrix*. Изменение размеров, повороты и инверсия при помощи матричных преобразований

Фильтр `Matrix` изменяет элемент страницы путем увеличения или уменьшения его размеров, а также поворачивает или инвертирует его при помощи матричных преобразований.

Свойства фильтра:

- ☐ `Dx` и `Dy` задают значения `fDx` и `fDy` матричных преобразований;
- ☐ `filterType` задает тип преобразования пикселей: `"bilinear"` (значение по умолчанию) или `"nearest neighbor"`;
- ☐ `m11`, `m12`, `m21`, `m22` указывают значения `fM11`, `fM12`, `fM21`, `fM22` матричных преобразований;
- ☐ `sizingMethod` определяет способ размещения нового изображения в пределах элемента страницы, к которому применяется фильтр. Может иметь два значения:
 - `"clip to original"` — обрезание изображения (значение по умолчанию);
 - `"auto expand"` — уменьшение или увеличение изображения;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 13 приведен пример использования фильтра `Matrix`.

Листинг 13. Фильтр *Matrix*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <script type="text/javascript">
<!--
var deg2 = Math.PI * 2 / 360;
var count = 400;
```

```

function f_SetRotation(obj, deg) {
    rad = deg * deg2;
    cos = Math.cos(rad);
    sin = Math.sin(rad);
    obj.filters.item(0).M11 = cos;
    obj.filters.item(0).M12 = -sin;
    obj.filters.item(0).M21 = sin;
    obj.filters.item(0).M22 = cos;
}
function f_Resize(obj, flMultiplier) {
    obj.filters.item(0).M11 *= flMultiplier;
    obj.filters.item(0).M12 *= flMultiplier;
    obj.filters.item(0).M21 *= flMultiplier;
    obj.filters.item(0).M22 *= flMultiplier;
}
function f_Spin(obj) {
    var flMultiple = count/720;
    count += 4;
    if (count>=360*3) {
        obj.onfilterchange = null;
    }
    f_SetRotation(obj, count);
    f_Resize(obj, flMultiple);
}
//-->
</script>
</head>
<body>
<div id="i_div" style="position:absolute; filter:progid:
DXImageTransform.Microsoft.Matrix(sizingMethod='auto expand') "
onfilterchange="f_Spin(this)">
    <div style="background-color: silver; padding:5;">
        Пример<br>матричных<br>преобразований<br>
    </div>
</div>
</body>
</html>

```

Фильтры *FlipH* и *FlipV*. Горизонтальный и вертикальный зеркальные образы

Фильтры *FlipH* и *FlipV* создают горизонтальный и вертикальный зеркальные образы (листинг 14).

Листинг 14. Фильтры FlipH и FlipV

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>
<span style="background-color: silver; position:absolute; left: 0;
top: 0; font-size:32pt; width:150px;">Пример
</span>
<span style="background-color: yellow; position:absolute; left: 0;
top: 60px; font-size:32pt; filter: FlipV; width:150px;">Пример
</span>
<span style="background-color: yellow; position:absolute; left: 190px;
top: 0; font-size:32pt; filter: FlipH; width:150px;">Пример
</span>
</body>
</html>
```

Фильтр *MotionBlur*. Эффект быстрого движения

Фильтр *MotionBlur* создает эффект быстрого движения (размытости).

Свойства фильтра:

- ☐ `add` определяет режим работы фильтра. Он может принимать два значения:
 - `true` — исходный элемент страницы перекрывает результат работы фильтра;
 - `false` — результат работы фильтра перекрывает исходный элемент страницы (значение по умолчанию);
- ☐ `direction` задает направление работы фильтра в градусах, которые должны быть кратны 45;
- ☐ `strength` указывает дистанцию, задаваемую в пикселах, на которую распространяется действие фильтра;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 15 приведен пример использования фильтра *MotionBlur*.

Листинг 15. Фильтр *MotionBlur*

```
<html>
<head>
```

```

<title>Фильтры</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>

</body>
</html>

```

Фильтр *Pixelate*. Отображение элемента страницы отдельными пикселями

Фильтр *Pixelate* отображает элемент страницы отдельными пикселями.

Свойства фильтра:

- ☐ `maxSquare` задает максимальный размер пикселя;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 16 приведен пример использования фильтра *Pixelate*.

Листинг 16. Фильтр *Pixelate*

```

<html>
<head>
<title>Фильтры</title>
<meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>

</body>
</html>

```

Фильтр *Shadow*. Снова о тени

Фильтр *Shadow* заставляет элемент страницы отбросить тень.

Свойства фильтра:

- ☐ `color` — цвет тени;
- ☐ `direction` задает направление тени в градусах, которые должны быть кратны 45;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 17 приведен пример использования фильтра *Shadow*.

Листинг 17. Фильтр Shadow

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>
<div style="background-color:silver;
filter:progid:DXImageTransform.Microsoft.Shadow(color=gray,
direction=130); position:absolute; top:10px; left:10px;
width:240px; height:60px;">
</div>
</body>
</html>
```

Фильтр *Wave*. Волнистое искажение элемента страницы

Фильтр *Wave* создает волнистое искажение элемента страницы.

Свойства фильтра:

- ☐ `add` — режим работы, который может принимать одно из двух значений:
 - `true` — исходный элемент страницы перекрывает результат работы фильтра;
 - `false` — результат работы фильтра перекрывает исходный элемент страницы (значение по умолчанию);
- ☐ `freq` — количество волн;
- ☐ `lightStrength` — окраска волн. Может принимать значение от 0 до 100;
- ☐ `phase` — фаза волн. Может принимать значение от 0 до 100;
- ☐ `strength` — дистанция в пикселах, на которую организуется фильтрация;
- ☐ `enabled` разрешает (`true`) или запрещает (`false`) применение фильтра.

В листинге 18 приведен пример использования фильтра *Wave*.

Листинг 18. Фильтр Wave

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
```

```
<body>
<span style="position:absolute; top: 80px; font-size: 80pt; color: navy;
filter:Wave(add=true, lightStrength=100, phase=10, strength=10,
freq=5);">
Пример</span>
</div>
</body>
</html>
```

Фильтр *Gray*. Отображение изображений черно-белыми

Фильтр *Gray* делает изображение черно-белым (листинг 19).

Листинг 19. Фильтр *Gray*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>

</body>
</html>
```

Фильтр *Invert*. Инверсия цветов

Фильтр *Invert* инвертирует компоненты и интенсивность света (листинг 20).

Листинг 20. Фильтр *Invert*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-
1251">
</head>
<body>


</body>
</html>
```

Фильтр *XRay*. "Рентгеновский" образ элемента

Фильтр *Xray* создает черно-белый "рентгеновский" образ элемента (листинг 21).

Листинг 21. Фильтр *XRay*

```
<html>
<head>
  <title>Фильтры</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
</head>
<body>


</body>
</html>
```

Преобразования. Эффекты перехода

Преобразования позволяют создавать визуальные эффекты при изменении свойств элементов или замене их содержимого. Перечислим и кратко опишем все доступные преобразования:

- ☐ **Barn** создает эффект "открывающейся и закрывающейся двери". Преобразование имеет следующие свойства:
 - **motion** задает направление движения. Может принимать два значения:
 - **out** — движение из центра к границам (значение по умолчанию);
 - **in** — движение от границ к центру;
 - **orientation** указывает направление преобразования. Может принимать два значения:
 - **horizontal** — по горизонтали;
 - **vertical** — по вертикали;
- ☐ **BlendTrans** плавно заменяет старое содержимое новым.
- ☐ **Blinds** имитирует эффект "открывающихся и закрывающихся жалюзи". Преобразование имеет следующие свойства:
 - **bands** задает количество полос, на которые разделяется область;
 - **direction** указывает направление. Может принимать следующие значения:
 - **up** — открытие вверх;
 - **down** — вниз;

- right — вправо;
 - left — влево;
- ❑ CheckerBoard создает эффект "шахматной доски". Преобразование имеет следующие свойства:
- direction — задает направление. Может принимать следующие значения:
 - up — открытие вверх;
 - down — вниз;
 - right — вправо;
 - left — влево;
 - squaresX и squaresY задают количество рядов по горизонтали и по вертикали;
- ❑ Fade имитирует эффект наплыва, то есть старое содержимое элемента страницы плавно пропадает, а новое одновременно плавно появляется. Преобразование имеет одно свойство:
- overlap задает время одновременного показа старого содержимого элемента страницы и нового относительно общей продолжительности работы фильтра. Может принимать значения от 0.0 до 1.0;
- ❑ GradientWipe отображает наплывание нового содержимого элемента страницы на старое, причем граница выглядит как градиентная полоса. Преобразование имеет следующие свойства:
- gradientSize задает часть площади элемента страницы, которая будет покрыта градиентной полосой. Может принимать значение от 0.0 до 1.0;
 - motion указывает направление преобразования. Может принимать два значения:
 - forward — движение происходит согласно значению свойства WipeStyle (значение по умолчанию);
 - reverse — в обратном направлении;
 - wipeStyle определяет направление преобразования:
 - 0 — по горизонтали (значение по умолчанию);
 - 1 — по вертикали;
- ❑ Inset — эффект, при котором новое содержимое элемента страницы диагонально заменяет старое;
- ❑ Iris имитирует открытие объектива камеры. Можно задавать такие свойства:
- irisStyle — форма лепестков, которая может принимать следующие значения:
 - circle — круг;

- `cross` — крест;
- `plus` — плюс;
- `square` — квадрат;
- `star` — звезда;
- `motion` — задает направление движения. Может принимать два значения:
 - `out` — движение из центра к границам (значение по умолчанию);
 - `in` — движение от границ к центру;
- ❑ `Pixelate` — старое содержимое элемента страницы рассыпается на отдельные пиксели, а новое содержимое собирается из отдельных пикселей. Можно указать следующее свойство:
 - `maxSquare` — максимальный размер пиксела;
- ❑ `RadialWipe` — старое содержание элемента радиально стирается, а новое появляется. Можно задать одно свойство:
 - `wipeStyle` — способ замещения содержимого элемента страницы, одно из следующих значений:
 - `clock` — вращение вокруг центра элемента страницы по часовой стрелке (значение по умолчанию);
 - `wedge` — вращение сразу в обе стороны вокруг центра элемента;
 - `radial` — радиальное вращение относительно верхнего левого угла;
- ❑ `RandomBars` — старое содержимое элемента страницы рассыпается на отдельные линии, а новое содержимое собирается из отдельных линий. Имеет одно свойство:
 - `orientation` — направление преобразования, которое может принимать два значения:
 - `horizontal` — по горизонтали;
 - `vertical` — по вертикали;
- ❑ `RandomDissolve` создает эффект поточечного появления нового содержимого на месте старого;
- ❑ `RevealTrans` плавно заменяет старое содержимое новым, используя эффект, определяемый единственным свойством:
 - `transition` — эффект преобразования, одно из следующих значений:
 - `0` — прямоугольник внутрь;
 - `1` — прямоугольник наружу;
 - `2` — круг внутрь;
 - `3` — круг наружу;

- 4 — "шторка" снизу вверх;
- 5 — "шторка" сверху вниз;
- 6 — "шторка" слева направо;
- 7 — "шторка" справа налево;
- 8 — вертикальные "жалюзи";
- 9 — горизонтальные "жалюзи";
- 10 — "шахматная доска" слева направо;
- 11 — "шахматная доска" сверху вниз;
- 12 — поточечное появление;
- 13 — эффект "открывающейся и закрывающейся двери" от границ справа и слева к центру;
- 14 — эффект "открывающейся и закрывающейся двери" от центра к правой и левой границам;
- 15 — эффект "открывающейся и закрывающейся двери" от границ сверху и снизу к центру;
- 16 — эффект "открывающейся и закрывающейся двери" от центра к верхней и нижней границам;
- 17 — диагональное наплзание от верхнего правого угла к нижнему левому;
- 18 — диагональное наплзание от нижнего правого угла к верхнему левому;
- 19 — диагональное наплзание от верхнего левого угла к нижнему правому;
- 20 — диагональное наплзание от нижнего левого угла к верхнему правому;
- 21 — "рассыпание" на отдельные линии по горизонтали;
- 22 — "рассыпание" на отдельные линии по вертикали;
- 23 — случайный эффект (один из описанных выше);

□ `Slide` — старое содержимое элемента страницы сдвигается в сторону, открывая под собой новое содержимое (эффект слайдов). Преобразование настраивается следующими свойствами:

- `slideStyle` задает способ замещения старого содержимого элемента страницы новым. Может принимать следующие значения:
 - `hide` — скрытие (значение по умолчанию);
 - `push` — выталкивание;

- `swap` — замена;
- `bands` определяет количество полос, на которые разделяется область;
- ☐ `Spiral` — старое содержимое элемента страницы спирально закрашивается новым содержимым. Можно задать следующие свойства:
 - `gridSizeX` и `gridSizeY` — количество полосок по горизонтали и по вертикали (от 1 до 100);
- ☐ `Stretch` растягивает новое содержимое элемента страницы, тем самым закрывая старое содержимое. Режим определяется одним свойством:
 - `stretchStyle` — способ замещения старого содержимого элемента страницы новым, принимает одно из следующих значений:
 - `hide` — скрытие;
 - `spin` — замена;
 - `push` — выталкивание;
- ☐ `Strips` — новое содержимое элемента страницы диагонально наплзает на старое содержимое. Можно задать свойство:
 - `motion` — угол, в направлении которого будет появляться новое содержимое элемента, принимает одно из следующих значений:
 - `leftdown` — нижний левый;
 - `leftup` — верхний левый;
 - `rightdown` — нижний правый;
 - `rightup` — верхний правый;
- ☐ `Wheel` создает эффект посекторного наплзания нового элемента страницы на старый. Для настройки применяется свойство:
 - `spokes` — количество секторов (от 2 до 20);
- ☐ `ZigZag` — новое содержимое элемента страницы зигзагообразно отдельными полосками заполняет старое. Можно указать следующие свойства:
 - `gridSizeX` и `gridSizeY` — количество полосок по горизонтали и по вертикали (от 1 до 100).

Помимо описанных свойств, специфических для конкретного преобразования, можно задавать следующие общие свойства:

- ☐ `duration` — продолжительность преобразования в секундах;
- ☐ `enabled` — разрешает (`true`) или запрещает (`false`) применение фильтра;
- ☐ `percent` — процент выполнения преобразования. Может принимать значения от 0 (преобразование еще не началось) до 100 (преобразование закончено);
- ☐ `status` — возвращает состояние выполнения преобразования. Может принимать следующие значения:

- 0 — если преобразование было остановлено;
- 1 — если оно было применено;
- 2 — если преобразование выполняется.

Общие методы:

- ❑ `apply()` — "замораживает" элемент страницы, после чего можно с ним делать все, что хотим;
- ❑ `play([<Продолжительность>])` — запускает преобразование;
- ❑ `stop()` — останавливает преобразование.

Чтобы запустить преобразование, нужно выполнить следующие действия:

1. Применить метод `apply()`.
2. Изменить свойство элемента (например, `visibility`, `innerText`, `backgroundColor` или `border`).
3. Применить метод `play()`.

Пример, приведенный в листинге 22, наглядно демонстрирует применение преобразований для получения динамических эффектов.

Листинг 22. Преобразования

```
<html>
<head>
  <title>Преобразования</title>
  <meta http-equiv="Content-Type" content="text/html; charset=windows-1251">
  <script type="text/javascript">
    <!--
    var trans = false;
    function f_start() {
      var index = i_sel.options(i_sel.selectedIndex).value;
      i_div.filters[index].apply();
      if (trans) {
        trans = false;
        i_img1.style.visibility="visible";
        i_img2.style.visibility="hidden";
      }
      else {
        trans = true;
        i_img1.style.visibility="hidden";
        i_img2.style.visibility="visible";
      }
      i_div.filters[index].play();
    }
  </script>
</head>
<body>
```

```

}
//-->
</script>
</head>
<body>
<div id="i_div" style="position:absolute; top:40px; left:10px;
width:150px; height:200px; filter:
progid:DXImageTransform.Microsoft.Barn(orientation='vertical',
motion='in', duration=1)
progid:DXImageTransform.Microsoft.Blinds(direction='left', bands=6,
duration=1)
progid:DXImageTransform.Microsoft.CheckerBoard(squaresX=6, squaresY=4,
direction='right')
progid:DXImageTransform.Microsoft.Fade(duration=5, overlap=0.4)
progid:DXImageTransform.Microsoft.GradientWipe(duration=3,
gradientsize=0.5)
progid:DXImageTransform.Microsoft.Inset
progid:DXImageTransform.Microsoft.Iris(irisStyle='cross', motion='out')
progid:DXImageTransform.Microsoft.Pixelate(maxSquare=30, duration=2,
enabled=false)
progid:DXImageTransform.Microsoft.RadialWipe(wipeStyle='wedge')
progid:DXImageTransform.Microsoft.RandomBars(orientation='vertical')
progid:DXImageTransform.Microsoft.RandomDissolve(duration=0.75)
progid:DXImageTransform.Microsoft.Slide(slideStyle='swap', bands=4)
progid:DXImageTransform.Microsoft.Spiral(gridSizeX=65, gridSizeY=65,
duration=1)
progid:DXImageTransform.Microsoft.Stretch(stretchStyle='push')
progid:DXImageTransform.Microsoft.Strips(duration=5, motion='leftup')
progid:DXImageTransform.Microsoft.Wheel(spokes=8)
progid:DXImageTransform.Microsoft.Zigzag(gridSizeX=8, gridSizeY=8)
progid:DXImageTransform.Microsoft.RevealTrans(duration=1,
transition=12);">
    <div id="i_img1" style="position:absolute; top:40px; left:10px;">
        
    </div>
    <div id="i_img2"
        style="position:absolute; top:40px; left:10px; visibility:hidden;">
        
    </div>
</div>
<select id="i_sel">
    <option value=0>Barn</option>
    <option value=1>Blinds</option>

```

```
<option value=2>CheckerBoard</option>
<option value=3>Fade</option>
<option value=4>GradientWipe</option>
<option value=5>Inset</option>
<option value=6>Iris</option>
<option value=7>Pixelate</option>
<option value=8>RadialWipe</option>
<option value=9>RandomBars</option>
<option value=10>RandomDissolve</option>
<option value=11>Slide</option>
<option value=12>Spiral</option>
<option value=13>Stretch</option>
<option value=14>Strips</option>
<option value=15>Wheel</option>
<option value=16>ZigZag</option>
<option value=17>RevealTrans</option>
</select>
<input type="button" value="Преобразовать" onclick="f_start();" >
</body>
</html>
```