SVG vs Canvas

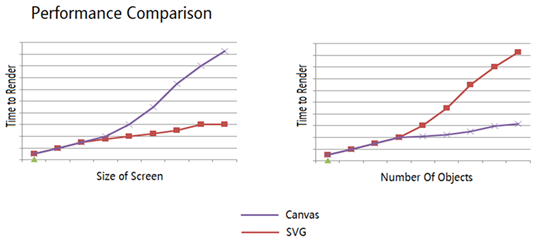
|  |  |
| --- | --- |
| **Canvas** | **SVG** |
| На основе пикселей (динамический .PNG) | На основе фигур |
| Один элемент HTML | Несколько графических элементов, которые входят в состав DOM |
| Изменяется только с помощью сценария | Изменяется с помощью сценария и CSS |
| Модель события/взаимодействие с пользователем детальны (x,y) | Модель события/взаимодействие с пользователем абстрактны (прямоугольник, путь) |
| Производительность улучшается при уменьшении поверхности, увеличении числа объектов (>10 000) или наличии обоих условий | Производительность улучшается при уменьшении числа объектов (<10 000), увеличении поверхности или наличии обоих условий |

|  |  |  |
| --- | --- | --- |
|  | **Canvas** | **SVG** |
| **Формат** | Растровый | Векторный |
| **Масштабирование** | Zoom | Scale |
| **Доступ** | Доступ к отдельным пикселям (RGBA) | Доступ к отдельным элементам (DOM) |
| **Индексируемость и Accessibility** | Виден только конечный растр (нельзя выделить фигуры, текст и т.п.)  — плохо для Accessibility | Можно посмотреть структуру (например, вытащить весь текст) |
| **Стилизация** | Визуальные стили задаются при отрисовке через API | Визуальные стили задаются атрибутами, можно подключать CSS |
| **Программирование** | JS API для работы с примитивами | DOM для работы с элементами |
| **Обновление** | Для обновления — рисование поверх или полная перерисовка | Возможно изменение отдельных элементов |
| **События** | Нет легкого способа для обработки событий мыши. Объекты под курсором надо определять вручную. | Легко вешаются события от мыши через DOM, обрабатываются автоматически. |
| **Интеграция кода** | Код на JS отдельно от Canvas | Внутрь можно включать JS |

HTML5 Canvas - просто поверхность для рисования, Canvas элемент не имеет представления о том, что в нем нарисовано и где именно, изображение - набор пикселей.

SVG – хранит «ссылку» на каждый отрисованный объект, каждый элемент SVG является элементом DOM, что позволяет навешивать различные обрасобытия на эти элементы

Производительность отрисовки при использовании Canvas и SVG:

[](http://blogs.msdn.com/cfs-file.ashx/__key/communityserver-blogs-components-weblogfiles/00-00-01-11-84-metablogapi/5428.image_5F00_3B76AA11.png)

Сanvas лучше работает при небольших размерах области отрисовки и на большом числе объектов, в SVG лучше подходит при необходимости масштабирования или вывода на большой экран и на не слишком большом количестве выводимых за раз объектов.

В случаях, где нужен попиксельный вывод, очевидно, лучше подходит Canvas.

Там, где диаграмма бьется на отдельные объекты, в которых нужно поддерживать интерактивность, лучше подходит SVG.

### Преимущества

|  |  |
| --- | --- |
| **Canvas** | **SVG** |
| * Высоко производительная 2D поверхность для рисования * Постоянная производительность – все является пикселем. * Производительность ухудшается лишь в случае увеличения разрешения изображения. * Есть возможность сохранять полученное изображение в формате .png или .jpg. * Лучше всего подходит для создания растровой графики (например в играх, фракталов и т.п.), редактирования изображений и операций, требующих манипулирования на уровне пикселей. | * Качество изображения не зависит от разрешения - это позволяет масштабировать SVG под любое разрешение экрана. * SVG имеет очень хорошую поддержку анимации. Элементы могут быть анимированы с использованием декларативного синтаксиса или с помощью JavaScript, также есть много библиотек (Raphael.js) * Разработчик имеет полный контроль над каждым элементом с помощью API SVG DOM в JavaScript. * SVG является файлом формата XML, что означает, что в зависимости от реализации каждого веб-браузера, доступность SVG документов может быть намного лучше, чем canvas элементов. * SVG обеспечивает в основном презентационную разметку, кроме того, семантика пользовательского интерфейса может быть улучшена ARIA атрибутами. |

* Недостатки

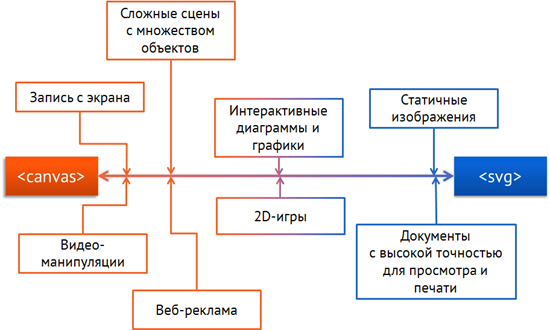
|  |  |
| --- | --- |
| * Canvas | * SVG |
| * Все, что нарисовано – пиксели, для элементов рисунка отсутствуют dom-узлы * отсутствует api для анимации * слабые возможности отрисовки текста * слабая accessibility * каждый элемент нужно отдельно перерисовывать | * медленная отрисовка документа, если много svg объектов, т.к. каждый объект – элемент DOM - дерева |

##### Лучше подходит Canvas

* Редактирование растровой графики
* Наложение эффектов на графику/видео
* Генерирование растровой графики (визуализация данных, фракталы, графики функций)
* Анализ изображенией
* Игровая графика (спрайты, фон и т.п.)

##### Лучше подходит SVG

* Масштабируемые интерфейсы
* Интерактивные интерфейсы
* Диаграммы, схемы
* Векторное редактирование изображений



Links:

<http://www.w3schools.com/svg/>

<http://www.w3schools.com/html/html5_canvas.asp>

<https://developer.mozilla.org/ru/docs/Web/SVG>

<https://msdn.microsoft.com/ru-ru/library/ie/gg193983(v=vs.85).aspx>

<http://html5ru.com/canvas-vs-svg.html>

<https://dev.opera.com/articles/svg-or-canvas-choose/>

<http://stackoverflow.com/questions/5882716/html5-canvas-vs-svg-vs-div>

<http://stackoverflow.com/questions/4996374/what-is-the-difference-between-svg-and-html5-canvas>

<http://smus.com/canvas-vs-svg-performance/>

<http://blogs.msdn.com/b/kichinsky/archive/2011/05/23/html5-canvas-svg.aspx>

canvas tutorials

http://codetheory.in/20-best-canvas-tutorials-and-examples-that-will-make-you-a-canvas-master/