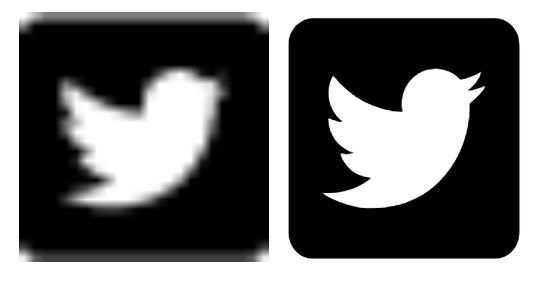
1. SVG in HTML and CSS

*Помимо привычных растровых форматов (JPG, PNG, GIF), в веб-разработке широко применяют векторный – SVG (сокращение от Scalable Vector Graphics – «масштабируемая векторная графика»).*

SVG – это вид графики, которую создают с помощью математического описания геометрических примитивов, образующих все детали будущего изображения. То есть в этом формате хранится не сама картинка, а инструкции для её построения по точкам и кривым, написаные на языке разметки SVG, расширяющем XML.



Прелесть SVG раскрывает масштабирование. Если увеличить растровое изображение – проявляется пикселизация: отдельные точки превращаются в одноцветные блоки пикселей, границы областей становятся ступенчатыми и картинка выглядит некрасиво.

Если же увеличить векторное изображение – все его детали останутся чёткими. Поэтому SVG-картинки отображаются на самых больших разрешениях без потери качества и просто идеальны для адаптивных сайтов.

У SVG -графики есть ещё одно достоинство. Её можно открывать и менять в любом редакторе кода и даже на лету. И наконец, при прочих равных SVG весит намного меньше растровых изображений.

Есть несколько вариантов добавления векторной графики в HTML-страницы:

**Через тег <img>.** Такой вариант подключения используется для контентных изображений вроде графиков, схем и логотипов.

**Через CSS-свойство background-image.** Этот случай подходит для декоративных изображений вроде фонов, иконок и прочей мелочи.

Минус всех этих способов в том, что взаимодействовать с внутренними элементами SVG не получится. Файл будет как за стеклом: смотреть можно, а трогать нельзя.

**Вставлять SVG -код прямо в HTML-файл – это третий способ.** И для нас самый интересный. Он позволяет не просто отображать готовую графику, но и менять её (толщину элементов, заливку, обводку и так далее).

Такая вставка выручает, когда при взаимодействии с картинкой нужно её визуально модифицировать – например, перекрасить SVG -иконку при наведении на неё курсора. Минус в том, что такие картинки не кэшируются отдельно от страницы.

2. CSS in SVG

*Существует четыре способа определения свойств элементов SVG: XML-атрибуты, встроенные стили, внутренние и внешние таблицы стилей.*

**XML-атрибуты**

Большинство свойств, используемых в SVG, можно задать в виде XML-атрибутов – эффект будет таким же, как при использовании CSS.

Мы можем убрать часть повторений, определив их внутри элемента группы <g>. Стили, примененные к группе, наследуются фигурами, входящими в нее, если только элемент фигуры специально не установит другое значение для того же свойства. Например, к группе окружностей в светофоре применен одинаковый стиль обводки.

Так же для предотвращения избыточной геометрии можно использовать элемент <use>. Он позволяет повторно использовать графику, определенную однажды в нашем файле, в других местах. Например, мы определяем базовый круг в светофоре один раз, а затем используем его повторно три раза, с разными вертикальными позициями и цветами заливки.

*Как мы видим, CSS не требуется для SVG, все эти задачи решаются с помощью XML-атрибутов. Однако использование CSS облегчает создание единого дизайна для всех страниц сайта и облегчает последующие изменения.*

**Inline-стили**

Все элементы SVG могут иметь атрибут style. Как и его эквивалент в HTML, он принимает строку «свойство: значение» CSS. Стили, заданные таким образом, заменяют как XML-атрибуты, так и значения из таблиц стилей.

**Блок <style>**

Мы можем включить внутреннюю таблицу стилей в документ SVG с помощью элемента <style>, аналогичную элементу <style> в HTML. Этот элемент может содержать любое допустимое содержимое таблицы стилей CSS, но основное содержимое – селекторы CSS, которые будет определять свойства элементов SVG.

Значения, заданные с помощью CSS, имеют приоритет над XML-атрибутами того же элемента. Например, толщина обводки 2px для кругов светофора заданная в блоке <style> имеет приоритет над толщиной в 3px, заданной в блоке <g>

**Внешние таблицы стилей**

Стили могут быть собраны во внешние файлы с расширением.css для использования в нескольких документах. Существует четыре способа подключения этих стилей для использования в SVG:

– правило импорта в блоке <style> или другом CSS-файле:

@import "style.css"

– инструкция по обработке стилей XML в прологе файла SVG:

<?xml-stylesheet href="style.css" type="text/css"?>

– элемент <link> в <head> документа HTML5, содержащий встроенный SVG:

<link href="style.css" rel="stylesheet" type="text/css">

– элемент HTML <link> в отдельном SVG-файле:

<html:link href="style.css" rel="stylesheet" type="text/css">

Внешние таблицы стилей никогда не загружаются из SVG-файлов, которые используются на веб-страницах (<img> или background-image)

3. CSS and SVG

*Объявление стилей CSS часто является просто удобством по сравнению с использованием XML-атрибутов, позволяя применять одни и те же правила стиля ко многим элементам (или многим файлам) без повторения кода (принцип DRY – Don’t repeat yourself). Однако с помощью CSS можно добиться гораздо большей гибкости, используя, например, псевдоклассы CSS.*

Для примера заставим зажигаться огни светофора при наведении мыши.

4. CSS vs SVG

До сих пор подчеркивалось, что CSS и SVG дополняют друг друга. Но нужно помнить, что они создавались для разных целей – один для форматирования документов, другой для отрисовки графики.

CSS предполагает, что большинство элементов в документе (блочных или строчных) содержит текст. Эти элементы рассматриваются как непрерывный поток, который может перестраиваться в зависимости от доступного пространства. Например, при изменении окна браузера или разрешения экрана на разных устройствах.

В отличии от этого, форма и положение элементов SVG неизменяемы. Здесь нет единого потока содержимого, который может быть перенесен на новую строку. Все элементы SVG расширяются или сжимаются вместе, сохраняя свое относительное положение.

Хотя CSS позволяет создавать простые геометрические фигуры, но для продвинутой графики лучше использовать SVG, а CSS использовать для стилизации SVG-контента