

Привет, SVG!

SVG — это формат векторной графики. В отличие от растровой графики — PNG, GIF, JPEG — SVG может растягиваться и сжиматься без потери качества, то есть такие картинки будут одинаково чёткими и на обычных экранах, и на ретине.

Ещё одно из достоинств SVG — человекопонятный код: его можно не только прочитать, но и написать руками. Можно открыть файл и отредактировать его без использования графического редактора, можно самому написать простую картинку.

Также SVG-элементы можно оформить с помощью CSS и добавить им интерактивности с помощью JavaScript, а кроме того, SVG [достаточно хорошо поддерживается](#) всеми современными браузерами, и его уже можно активно использовать.

Давайте познакомимся с ним поближе. Вот простой пример кода:

```
<svg>

  <circle r="50" cx="50%" cy="50%" fill="yellowgreen"/>

</svg>
```

SVG-элемент вставляется с помощью тега `svg`, внутри которого уже находится остальное содержимое: фигуры, картинки или текст.

Содержимое в этом примере — это кружок (`circle`) зелёного цвета (`fill="yellowgreen"`).

SVG можно встраивать несколькими разными способами, мы рассмотрим их позже, а сейчас будем вставлять его непосредственно в код страницы.

Рисуем прямоугольник

В SVG есть несколько способов нарисовать фигуру, для простых фигур есть свои теги.

Например, прямоугольник рисуется с помощью тега `rect`. Код простого прямоугольника выглядит так:

```
<rect width="150" height="100"/>
```

Обратите внимание: все теги в SVG должны быть закрыты, то есть должно быть так: `<rect .../>` или так: `<rect ...></rect>`. Мы будем использовать первый способ.

Результат:

Атрибуты `width` и `height` управляют, соответственно, шириной и высотой фигуры. Значения можно задавать и в пикселях, и в процентах.

Для значений в пикселях после значения не нужно писать `px`, потому что пиксели — единица измерения, используемая в SVG по умолчанию. Проценты рассчитываются относительно размеров всего SVG-изображения: горизонтальные значения относительно ширины, вертикальные — относительно высоты.

В современных браузерах размерами и положением фигур нельзя управлять через CSS, но эта возможность появится в будущем.

Координаты прямоугольника

Чтобы задать координаты прямоугольника, используются атрибуты `x` и `y`:

```
<rect width="50%" height="100" x="20" y="50"/>
```

Координаты определяют положение верхнего левого угла фигуры.

Скругление углов

Скруглением углов прямоугольника управляют параметры `rx` и `ry`. Атрибут `rx` задаёт скругление по горизонтали, а `ry` — по вертикали. Если атрибут `ry` не задан, он будет равен `rx`.

Пример кода:

```
<rect width="50%" height="100" rx="50" ry="20"/>
```

В SVG можно рисовать не только четырёхугольники, но и многоугольники, это делается с помощью тега `polygon`. Пример кода:

```
<polygon points="70,5 90,41 136,48 103,80 111,126 70,105 29,126 36,80 5,48 48,41"/>
```

В атрибуте `points` задаются координаты вершин фигуры. Каждая координата задаётся по `x` и `y`. Координаты в `points` нельзя задавать в процентах.

Рисуем окружность

Окружность рисуется с помощью тега `circle`. Пример кода:

```
<circle r="50"/>
```

Атрибут `r` — радиус окружности.

В отличие от предыдущих фигур, положение окружности в пространстве определяется координатами центра фигуры: атрибут `cx` задаёт положение по горизонтальной оси, `cy` — по вертикальной.

По умолчанию координаты центра окружности равны `0,0`, поэтому она находится в верхнем левом углу. Подвинем фигуру:

```
<circle r="50" cx="100" cy="50%"/>
```

Значения можно задавать как в пикселях, так и в процентах. Процентные значения рассчитываются относительно размеров SVG-элемента.

Радиус и координаты можно задавать только атрибутами, с помощью CSS это сделать нельзя.

Рисуем эллипс

Эллипс рисуется почти так же, как круг, но у него два радиуса: по горизонтальной оси — `rx`, и по вертикальной — `ry`.

```
<ellipse rx="30" ry="40%"/>
```

Расположение эллипса, так же как и для `circle`, задаётся с помощью `cx` и `cy`.

```
<ellipse rx="30" ry="40%" cx="50%" cy="50%"/>
```

Рисуем линии

Линии рисуются с помощью тега `line`. Координаты начала линии задаются атрибутами `x1` и `y1`, координаты конца — атрибутами `x2` и `y2`. Координаты можно задавать в процентах.

Пример кода:

```
<line x1="220" y1="10" x2="20" y2="130"/>
```

Так как линия не образует фигуру с внутренним контуром, для отображения ей нужно задать не заливку, а обводку. Обводкой управляют два атрибута: `stroke` и `stroke-width`.

Атрибут `stroke` задаёт цвет обводки, `stroke-width` — толщину линии.

```
<line x1="220" y1="20" x2="20" y2="90" stroke="violet" stroke-width="5" />
```

Результат:

Можно задать только цвет линии, тогда толщина обводки по умолчанию будет равна одному пикселю.

Рисуем ломанные линии

Ломанные линии рисуются с помощью тега `polyline`. Координаты точек на линии задаются в атрибуте `points`, как для `polygon`.

Пример кода:

```
<polyline points="10,135 100,10 55,135 10,10 105,135"/>
```

Результат:

Разница между `polygon` и `polyline` заключается в поведении обводки: у многоугольника обводка замыкается сама по себе, а у ломаной линии — остаётся незамкнутой .