# Знакомство с 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:

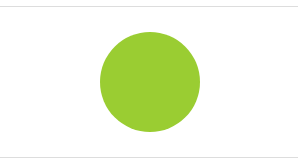
<svg>

<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"**). Вот так будет выглядеть действие этого кода в браузере:



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"/>

**Задание 1.**

1. Нарисуйте круг красного цвета и прямоугольник шириной **50%** и высотой **100** оранжевого цвета (**fill="orange"**).
2. Задайте прямоугольнику атрибут **x** равным 25%, а атрибут **y**— 25 пикселям.
3. Задайте значение **rx** равным **20**. Задайте значение **ry** равным **50**, обратите внимание как изменилась форма фигуры.

**Задание 2. «Починка телевизора».**

Мама заказала телевизор, но не знала, что доставкой займётся почта РФ. И телевизор пришёл немного «помятый». Почините картинку, подобрав размеры, координаты и радиусы скругления фигур. Все значения кратны пяти.

Исходный код

**CSS**

svg {

border: 1px solid #dddddd;

}

**HTML**

<svg height="250" width="330">

<rect class="leg" width="15" height="30" x="180" y="210" rx="2" fill="black"/>

<rect class="leg" width="15" height="30" x="5" y="190" rx="10" fill="black"/>

<rect class="case" width="300" height="200" x="20" y="25" rx="30" fill="dimgray"/>

<rect class="screen" width="200" height="160" x="30" rx="10" fill="deepskyblue"/>

<rect class="panel" width="80" height="110" x="200" y="50" fill="black"/>

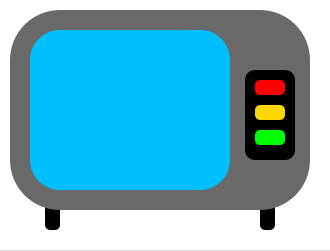
<rect class="button" width="20" height="10" x="245" y="70" rx="2" fill="red"/>

<rect class="button" width="30" height="30" x="265" y="95" rx="4" fill="gold"/>

<rect class="button" width="40" height="20" x="225" y="120" rx="7" fill="lime"/>

</svg>

Результат исправления должен быть таким



Ответ

<svg height="250" width="330">

<rect **class**="leg" width="15" height="30" x="260" y="200" rx="5" fill="black"/>

<rect **class**="leg" width="15" height="30" x="45" y="200" rx="5" fill="black"/>

<rect **class**="case" width="300" height="200" x="10" y="10" rx="50" fill="dimgray"/>

<rect **class**="screen" width="200" height="160" x="30" y="30" rx="30" fill="deepskyblue"/>

<rect **class**="panel" width="50" height="90" x="245" y="70" fill="black" rx="10"/>

<rect **class**="button" width="30" height="15" x="255" y="80" rx="5" fill="red"/>

<rect **class**="button" width="30" height="15" x="255" y="105" rx="5" fill="gold"/>

<rect **class**="button" width="30" height="15" x="255" y="130" rx="5" fill="lime"/>

</svg>

## **Многоугольники**

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"/>

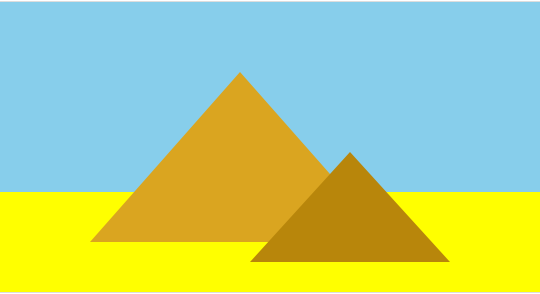
И результат:



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

**Задание 3. Пирамиды**

1. Нарисуйте многоугольник с координатами 5,135 115,5 225,135 и заливкой violet.
2. Нарисуйте пейзаж с пирамидами, используя rect и polygon. Цвета: skyblue, yellow, goldenrod, darkgoldenrod. Все значения кратны десяти.



Ответ:

<rect width="100%" height="190" fill="skyblue"/>

<rect width="100%" height="100" fill="yellow" y="190"/>

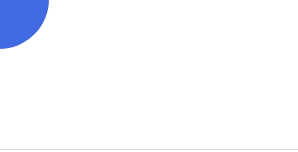
<polygon points="240,70 390,240 90,240" fill="goldenrod"/>

<polygon points="350,150 450,260 250,260" fill="darkgoldenrod"/>

## **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.

**Задание 4. Окружности**

1. Нарисуйте окружность с радиусом 10 и центром по горизонтали (cx) равным 50, по вертикали (cy) — 50%, цвет заливки — crimson.
2. Вторую окружность: радиус — 30, cx = 105, cy = 50%, цвет заливки — orangered.
3. третью: радиус — 50, cx = 200 и cy = 50%, цвет заливки — gold.

**Задание 5. Мишень**

Нарисуйте мишень, используя circle и rect. Все размеры и координаты кратны пяти.

**Цвета:** yellowgreen, skyblue, white, crimson и black.



Ответ:

<svg width="300" height="290">  
<rect width="300" height="180" fill="skyblue"/>  
<rect width="300" height="110" y="180" fill="yellowgreen"/>  
<rect width="10" height="30" x="80" y="230" rx="5" fill="black"/>  
<rect width="10" height="30" x="210" y="230" rx="5" fill="black"/>  
<circle r="100" cx="50%" cy="160" fill="crimson"/>  
<circle r="80" cx="50%" cy="160" fill="white"/>  
<circle r="60" cx="50%" cy="160" fill="crimson"/>  
<circle r="40" cx="50%" cy="160" fill="white"/>  
<circle r="20" cx="50%" cy="160" fill="crimson"/>  
</svg>

## **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%"/>

**Задание 6. Эллипс. Удивлённый смайлик**

1. Нарисуйте эллипс с радиусом по горизонтали равным 40% и радиусом по вертикали — 30.
2. Переместите фигуру, задав cx значение 50% и cy значение 80%.
3. Добавьте зелёную заливку (yellowgreen).
4. Используя круги и эллипсы нарисуйте удивленный смайл.

|  |  |
| --- | --- |
| **Цвета:** lightgray, gold, white, black, crimson. Все значения кратны пяти. |  |

Решение:

<svg width="310" height="290">

<ellipse rx="75" ry="20" cx="150" cy="270" fill="lightgray"/>

<circle r="120" cx="150" cy="150" fill="gold"/>

<circle r="50" cx="100" cy="120" fill="white"/>

<circle r="50" cx="170" cy="120" fill="white"/>

<circle r="10" cx="100" cy="120" fill="black"/>

<circle r="10" cx="170" cy="120" fill="black"/>

<ellipse rx="10" ry="20" cx="135" cy="210" fill="crimson"/>

</svg>

## **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.

**Задание 7.** Нарисуйте линию оранжевого цвета (orange) толщиной 10 пикселей с началом в точке (20,20) и концом в точке (200,100).

**Задание 8. арифметические знаки**

|  |
| --- |
| Используя линии, нарисуйте арифметические знаки.  Цвета: teal, orangered, royalblue, purple и orange.  Все значения кратны пяти. |
|  |

Решение

<svg width="550" height="120">

<line x1="60" y1="10" x2="60" y2="110" stroke="teal" stroke-width="20"/>

<line x1="10" y1="60" x2="110" y2="60" stroke="teal" stroke-width="20"/>

<line x1="120" y1="60" x2="220" y2="60" stroke="orangered" stroke-width="20"/>

<line x1="230" y1="15" x2="320" y2="105" stroke="royalblue" stroke-width="20"/>

<line x1="320" y1="15" x2="230" y2="105" stroke="royalblue" stroke-width="20"/>

<line x1="350" y1="105" x2="440" y2="15" stroke="purple" stroke-width="20"/>

<line x1="450" y1="45" x2="540" y2="45" stroke="orange" stroke-width="20"/>

<line x1="450" y1="75" x2="540" y2="75" stroke="orange" stroke-width="20"/>

</svg>

## **Рисуем ломаные линии**

Ломаные 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 (левая фигура), а у ломаной линии — остаётся незамкнутой (фигура справа):



**Задание 8.**

1. Нарисуйте линию с координатами 50,180 100,20 150,180 20,80 180,80 50,180 100,20.
2. Добавьте ей обводку цвета red толщиной 5 пикселей. Чтобы убрать чёрную заливку, задайте атрибуту fill значение none.



**Задание 9. Воздушные шары**

|  |  |
| --- | --- |
| На  Нарисуйте открытку с воздушными шарами, используя rect, circle, polygon, ellipse и polyline.  **Цвета:** lemonchiffon, crimson, gold, brown.  Все значения кратны пяти. |  |

Решение

<svg width="360" height="500">

<rect width="320px" height="460px" x="20" y="20" rx= "20" fill="lemonchiffon"/>

<polyline points="250,290 260,320 245,350 260,380 245,410 260,440 " stroke="brown" stroke-width="1px" fill="none"/>

<polyline points="140,270 150,300 135,330 150,360 135,390 150,420 " stroke="brown" stroke-width="1px" fill="none"/>

<circle r="100" cx="140" cy="170" fill="crimson"/>

<polygon points="140 270 130 280 150 280" fill="crimson"/>

<ellipse rx="60" ry="80" cx="250" cy="210" fill="gold"/>

<polygon points="250 290 240 300 260 300" fill="gold"/>

</svg>