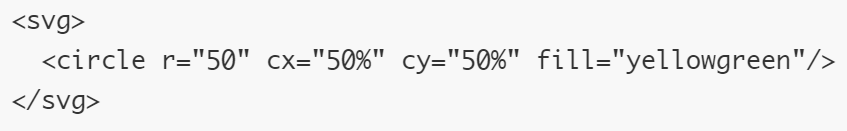
**SVG**

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 (вид дисплея).

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.

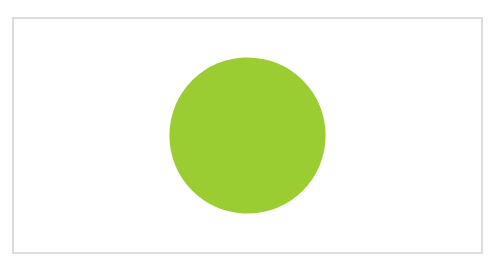
Также SVG-элементы можно оформить с помощью CSS и добавить им интерактивности с помощью JavaScript, а кроме того, SVG [достаточно хорошо поддерживается](http://caniuse.com/#search=svg) всеми современными браузерами, и его уже можно активно использовать.

Давайте познакомимся с ним поближе. Вот простой пример кода:



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

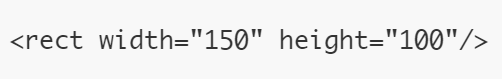
Содержимое в этом примере — это кружок (circle) зелёного цвета (fill="yellowgreen"). Вот так будет выглядеть действие этого кода в браузере:



**Рисуем прямоугольник**

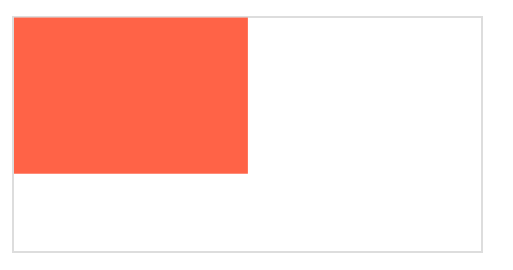
В SVG есть несколько способов нарисовать фигуру, для простых фигур есть свои теги.

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



​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:



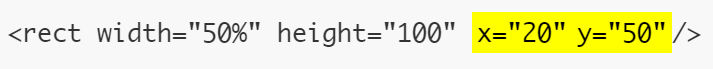
Атрибуты width и height управляют, соответственно, шириной и высотой фигуры. Значения можно задавать и в пикселях, и в процентах.

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:

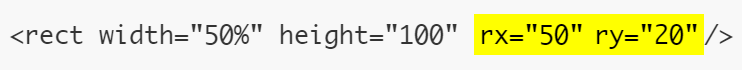


Координаты определяют положение верхнего левого угла фигуры.

**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.

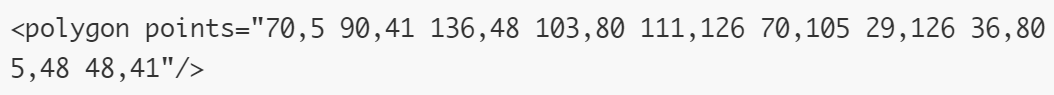
Пример кода:



**Выполните задание 1 – починку телевизора**

**Многоугольники**

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



И результат:

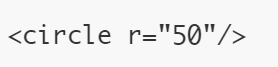


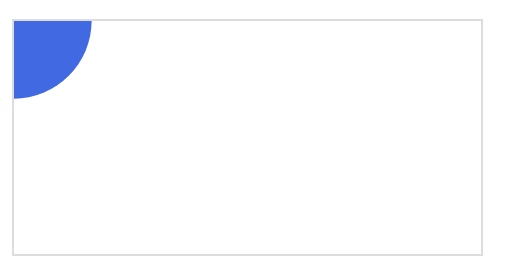
В атрибуте points задаются координаты вершин фигуры. Каждая координата задаётся по x и y. Координаты в points нельзя задавать в процентах.

**Выполните задание 2 – пирамиды**

**Рисуем окружность**

Окружность рисуется с помощью тега circle. Пример кода:

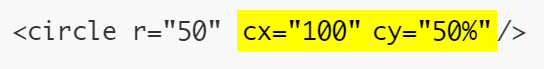
****

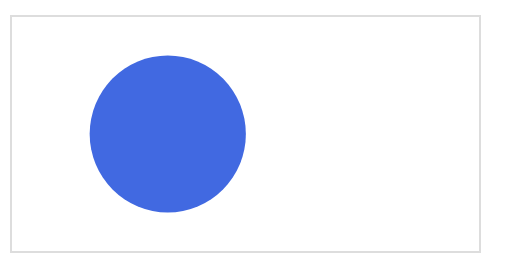


Атрибут r — радиус окружности.

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:





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

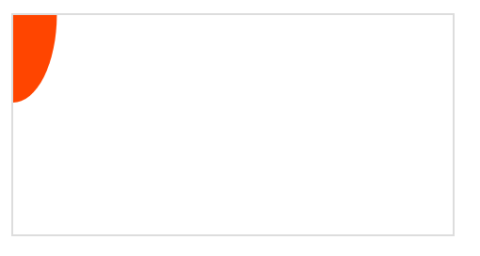
Радиус и координаты можно задавать только атрибутами, с помощью CSS это сделать нельзя.

**Выполните задание 3 – мишень**

**Рисуем эллипс**

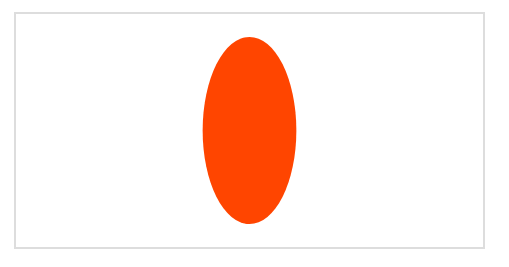
Эллипс рисуется почти так же, как круг, но у него два радиуса: по горизонтальной оси — rx, и по вертикальной — ry.

****

****

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

****

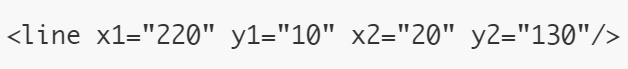
****

**Выполните задание 4 – удивлённый смайлик**

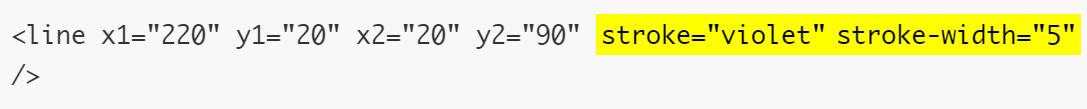
**Рисуем линии**

Линии рисуются с помощью тега line. Координаты начала линии задаются атрибутами x1и y1, координаты конца — атрибутами x2 и y2. Координаты можно задавать в процентах.

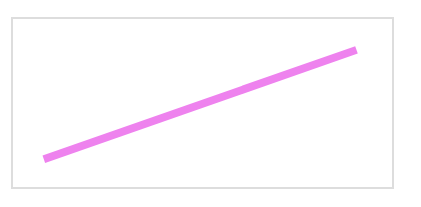
Пример кода:



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.



Результат:



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

**Выполните задание 5 – арифметические знаки**

**Рисуем ломаные линии**

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

Пример кода:



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):

****

**Выполните задание 6 – воздушные шары**