

## Лабораторная работа № 14

### Масштабируемая векторная графика

**Цель работы:** изучить способы вставки SVG-изображения на web-страницу, принципы создания svg-фигур и svg-контуров; познакомиться с правилами применения трансформации, градиентной заливки и анимации к svg-фигурам.

#### Теоретические сведения для выполнения работы

##### Использование SVG

Масштабируемая векторная графика (Scalable Vector Graphics, SVG) представляет собой вид графики, который создается с помощью математического описания геометрических примитивов (линий, кругов, эллипсов, прямоугольников, кривых), которые образуют изображение. Изображения SVG описываются тестовыми файлами с применением языка XML и предназначены для описания двухмерной векторной или смешанной векторно-растровой графики.

К преимуществам SVG-изображений относится:

1. Отсутствие потери качества при масштабировании.
2. Могут создаваться и редактироваться в любом текстовом редакторе
3. Совместимость со стандартами консорциума W3C: DOM и XSL.
4. Размеры их файлов являются небольшими по сравнению с любым другим типом файлов изображений.
5. Можно добавлять несколько гиперссылок.
6. Поддержка скриптов и анимации в SVG позволяют создавать динамичную и интерактивную графику.

Преимущественно .svg используют в дизайне иконок, логотипов и элементов пользовательского интерфейса для веб-сайтов, а также можно создавать графики и диаграммы, простую инфографику, масштабируемые дорожные карты, легкие игры вроде sudoku или кроссвордов.

Существуют следующие способы использования svg в веб-бразерах:

1. Подключение SVG-файла в HTML-документ с помощью тегов `img`, `embed`, `object` и `iframe`.

```

<embed src="example.svg" type="image/svg+xml">
<object data="example.svg" type="image/svg+xml"></object>
<iframe src="example.svg" width="200" height="300" style="border:
none"></iframe>
```

2. Вставка кода в HTML-документ в элементе `<svg>...</svg>`:

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1">
<!-- SVG-код -->
</svg>
```

4. подключение в PHP-документ с помощью функции `include`:

```
<? include("example.svg"); ?>
```

### 3. Использование SVG-файла в качестве фонового изображения:

```
<style>
  .home {
    display: block;
    width: 200px;
    height: 300px;
    background: url(example.svg) 0 0;
    background-size: contain;
  }
</style>
<a href="/" class="home"></a>;
```

Контейнер SVG документа имеет бесконечные размеры. Viewport и viewBox – это две прямоугольные области просмотра, которые ограничены конечными значениями высоты и ширины, указанными в параметрах viewport и viewBox. При изменении параметров viewport и viewBox появляется возможность отобразить без искажений или трансформировать любой конкретный фрагмент холста SVG. Например, viewBox="0 0 400 400" – определяет пользовательскую область просмотра –User Space с пользовательской системой координат – User Coordinate system. Так как viewport является предком для viewBox, то начало системы координат viewBox, по умолчанию, также как и системы координат viewport находится в левом верхнем углу (0,0) и положительное направление оси “X” – будет также вправо, а оси “Y” – вниз.

```
<svg width="400" height="400" version="1.1" viewBox="0 0 400 400"
xmlns="http://www.w3.org/2000/svg">
```

Первые два числа ViewBox = “min-x” и “min-y” задают начало пользовательской системы координат, “width” и “height” – определяют ширину и высоту “пользовательской области просмотра” и одновременно отвечают за масштабирование изображения.

#### Основные элементы SVG

К основным элементам, которые могут быть созданы являются прямая линия, ломанная линия, многоугольник, прямоугольник, круг, эллипс, сложная траектория. Соответствующие им теги представлены в таблице 14.1

Таблица 14.1

Элементы SVG	Атрибуты
line (прямая линия)	x1 — координата начальной точки линии по оси X; y1 — координата начальной точки линии по оси Y; x2 — координата конечной точки линии по оси X; y2 — координата конечной точки линии по оси Y
polyline (ломанная линия)	points — координаты ломанной линии парами x,y через пробел
polygon (многоугольник)	points — координаты ломанной линии парами x,y через пробел
rect (прямоугольник)	x — координата левой верхней точки прямоугольника по оси X;

	y — Координата левой верхней точки прямоугольника по оси Y; width — ширина прямоугольника; height — высота прямоугольника; rx — радиус закругления углов прямоугольника по оси X; ry — радиус закругления углов прямоугольника по оси Y;
circle (круг)	cx — координата центра круга по оси X; cy — координата центра круга по оси Y; r — радиус круга;
ellipse (эллипс)	cx — координата центра эллипса по оси X; cy — координата центра эллипса по оси Y; rx — радиус эллипса по оси X; ry — радиус эллипса по оси Y;

Создание сложной траектории задается тегом `<path>`, который позволяет создавать произвольные фигуры. Форма фигуры задается атрибутом `d`, значение которого — это набор специальных команд. Эти команды могут быть и в верхнем, и в нижнем регистре. Верхний регистр указывает на то, что применяется абсолютное позиционирование, а нижний — относительное. Список команд и их значений представлены в таблице 14.2

Таблица 14.2

Команды тега <code>&lt;path&gt;</code>	Значение команды
M, m — начальная точка	mx, my — координаты точки
L, l — отрезок прямой	lx, ly — координаты от текущей точки линии к указанной
H, h — горизонтальная линия	hx — координата до которой создается линия по оси X
V, v — вертикальная линия	vy — координата до которой создается линия по оси Y
A, a — дуга эллипса	rx,ry — радиусы дуги эллипса; x-axis-rotation — угол поворота дуги относительно оси X; large-arc-flag — если (=1), то строится большая части дуги, если (=0) — меньшая; sweep-flag — если (=1), то дуга строится по часовой стрелке, если (=0) — против часовой стрелке; x,y — координаты конечной точки дуги
C, c — кубическая кривая Безье	x1,y1 — координаты первой контрольной точки; x2,y2 — координаты второй контрольной точки; x,y — координаты конечной точки кривой.
S, s — гладкая кубическая кривая Безье	x2,y2 — координаты второй контрольной точки; x,y — координаты конечной точки кривой. Первая контрольная точка является зеркальным отражением второй контрольной точки
Q, q — квадратичная кривая Безье	x1,y1 — координаты контрольной точки; x,y — координаты конечной точки кривой.
T, t — гладкая квадратичная кривая Безье	x,y — координаты конечной точки кривой. Контрольная точка этой команды является зеркальным отражением контрольной точки предыдущей команды.
Z, z — замыкание траектории	не имеет значений

#### Пример

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="600" height="100">
```

```
<path d="M10,15 h50 v60 L110,55 A25,35 -30 0,1 150,30
M160,50 C160,110 260,110 260,50 S360,-10 360,50
M370,50 Q420,100 470,50 T570,50 z"
stroke="#b4241b" stroke-width="3" fill="none" />
</svg>
```

Сложные SVG фигуры можно нарисовать в векторных редакторах Adobe Illustrator, CorelDRAW, Inkscape (рекомендуемый свободный редактор SVG-графики) и сохранить в формате svg. Далее полученный документ открывается в Блокноте, FrontPage или любом другом редакторе, в окне которого будет представлен автоматически корректно созданный код. Данный код можно скопировать и вставить в HTML.

К общим атрибутам используемым во всех элементах относятся:

1. stroke — цвет линии;
2. stroke-width — толщина линии;
3. stroke-linecap — стиль концов линии. Возможные значения атрибута: round – по форме круга; square – по форме квадрата;
4. stroke-dasharray — Чередование штрихов и пробелов в пунктирной линии;
5. fill — цвет заливки (none – без заливки);
6. fill-opacity — прозрачность заливки (от 0 до 1);
7. fill-rule — правило заливки. Возможные значения атрибута: nonzero — сплошная заливка; evenodd – внутренняя часть фигуры не заливается.
8. style — стиль элемента;
9. class — класс элемента.

Преобразования задаются в атрибуте **transform** SVG-элемента. Можно указать несколько преобразований через пробел. Виды трансформации:

- rotate(rotate-angle [cx cy]) – поворот;
- scale(sx [sy]) – масштабирование;
- translate(tx [ty]) – перенос;
- skewX(skew-angle) – наклон по оси X;
- skewY(skew-angle) – наклон по оси Y.

```
<svg xmlns="http://www.w3.org/2000/svg" version="1.1" width="600"
height="70">
  <rect x="100" y="10" width="50" height="50" stroke="#ccc" stroke-
width="3" fill="#fff" />
  <rect x="100" y="10" width="50" height="50" stroke="#b4241b" stroke-
width="3" fill="#ffff00" transform="scale(0.5)" />
</svg>
```

В теге <defs> (библиотека элементов и эффектов) можно задать градиентную заливку (linearGradient, radialGradient) и применить ее к отдельным фигурам. Также в этой библиотеке можно хранить любые элементы SVG: pattern, marker, path, gradient, а так же любую из основных

фигур SVG. Для использования этого элемента в этом же файле или в другом файле, элементу необходимо присвоить уникальное имя id. Например:

```
<defs>
  <linearGradient id = "MyGradient">
    <stop offset = "30%" stop-color = "red"/>
    <stop offset = "70%" stop-color = "yellow"/>
  </linearGradient>
</defs>
<rect x = "0" y = "0" width = "150" height="150" fill =
"url(#MyGradient)"/>
```

Часто используется парный тег: <g> для объединения нескольких фигур в группу для последующих действий над ней, как над одним целым, например для: перемещения, поворота, масштабирования и т.д. Группе так же может быть присвоен уникальный id для повторного использования. В свою очередь несколько групп могут быть объединены в одну.

```
<g id="gr1+gr2" transform="translate(50,70)">
  <g id = "gr1">
    <rect x="30" y="50" width="120" height="50" style="fill-opacity: 0.7; fill:
red;" />
    <rect x="30" y="140" width="120" height="50" style="fill:yellow; stroke-
width:3; stroke: blue;"/>
  </g>
  <g id="gr2">
    <rect x="30" y="230" width="120" height="50" style="fill:none; stroke-
width:3; stroke: blue;"/>
    <rect x="30" y="320" width="120" height="50" style="fill:none; stroke-
width:8; stroke: red; stroke-opacity:0.4;"/>
  </g>
</g>
```

Для создания копий svg-фигур и их размещения на странице, а также добавления различных преобразований используется тег <use>, указывается id контура и прописываются его координаты.

```
<svg viewBox="0 0 30 10" xmlns="http://www.w3.org/2000/svg">
  <circle id="myCircle" cx="5" cy="5" r="4" stroke="blue"/>
  <use xlink:href="#myCircle" x="10" fill="blue"/>
  <use xlink:href="#myCircle" x="20" fill="white" stroke="red"/>
</svg>
```

## Анимация SVG

SMIL (the Synchronized Multimedia Integration Language) – язык разметки, с помощью которого осуществляется анимация в SVG. В SVG каждой отдельной геометрической фигуре можно присвоить свои инструкции по анимации.

Тег <animate> анимирует отдельные свойства, который прописывается непосредственно в теге фигуры с указанием анимированного свойства в атрибуте attributeName. В примере анимируется свойство cx, расположение по оси x изменяется от 100 до 300px:

```
<circle cy="70" r="50" fill="red">
  <animate attributeName="cx" from="100" to="300" dur="5s"/>
</circle>
```

Анимируемыми свойствами могут быть также толщина обводки stroke-width; радиус в круге или размеры в других фигурах; заливка (при этом изменение заливки можно задавать от цвета к цвету); прозрачность opacity от 0 до 1, пунктирная заливка stroke-dasharray и др.

Ниже приведены различные примеры:

```
<animate attributeName="fill" from="blue" to="red" dur="6s"
repeatCount="indefinite"/>
<animate attributeName="fill" values="red; yellow; green; #0000ff <!--
несколько промежуточных значений цвета--> dur="15s" fill="freeze"/>
```

Можно задавать сразу несколько анимаций, и они будут выполняться одновременно, для последовательного выполнения можно задать атрибут begin.

```
<animate attributeName="fill" from="red" to="blue" dur="6s" begin="0s"
repeatCount="indefinite" />
<animate attributeName="fill" from="blue" to="red" dur="6s" begin="6s"
repeatCount="indefinite" />
```

В теге <animate> можно ссылаться на анимируемый объект через его id:

```
<circle id="myelement" r="50" cx="100" cy="70" fill="red"/>
<animate xlink:href="#myelement" attributeName="fill" from="red"
to="blue" dur="5s"/>
```

Тег <animateTransform> предназначен для создания анимации трансформаций, вид трансформации указывается в атрибуте type.

```
<rect x="5" y="10" width="50" height="25" fill="grey">
  <animateTransform attributeName="transform" attributeType="XML"
type="scale" from="1" to="4" dur="5s" fill="freeze"/></rect>
<animateTransform xlink:href="#mygroup" attributeType="XML" type="rotate"
from="0, 60 50" to="45,60,50" dur="5s"
additive="sum" fill="freeze"/>
```

Для обработки событий можно воспользоваться тегам анимации с атрибутами begin и end: begin="mousedown", end="mouseup", begin="mouseover".

Например

```
<svg width="500" height="100">
  <circle id="orange-circle" r="30" cx="50" cy="50" fill="orange" />
  <animate
    xlink:href="#orange-circle"
```

```
attributeName="cx"  
from="50"  
to="450"  
dur="1s"  
begin="click"  
fill="freeze" />  
</svg>
```

### Работа с текстом

Текст в элементе SVG определяется с помощью тега `<text>`. К специфическим атрибутам, используемым в `text` относятся:

1. `x` и `y` – базовая линия текста `<text x="0" y="20">Text</text>`
2. `dx` и `dy` – размещение текстовых областей относительно друг друга
3. `text-anchor` – выравнивание текстовой строки относительно точки (`x`, `y`). Может принимать значения `start`, `middle`, `end`
4. `rotate` – поворот текста `transform="rotate(30 20, 40)"`
5. `textLength` – ширина текстовой области
6. `lengthAdjust` – сжатие и растягивание текста, используется вместе с атрибутом `textLength`. Может принимать значения `spacing` и `spacingAndGlyphs`.

Тег `tspan` в SVG аналогичен тегу `span`. Используется при необходимости применить стиль к отдельному элементу.

```
<tspan style="font-family: Arial;">Текст</tspan>
```

С помощью тега `textPath` осуществляется отображение текста вдоль направляющей линии. Шаблон использования `textPath`:

```
<defs>  
<path id="idname" fill="..." stroke="..." d="..." />  
</defs>  
<use xlink:href="#idname"/>  
<text x="..." y="...">  
<textPath xlink:href="#idname">  
Write your text here.  
</textPath>
```

Примеры направляющих линий:

```
d="M0.057,0.024c0,0,10.99,51.603,102.248,51.603c91.259,0,136.  
172,53.992,136.172,53.992"
```

```
d="M73.2,148.6c4-6.1,65.5-96.8,178.6-95.6c111.3,1.2,170.8,90.3,175.1,97"
```

```
d="M6,150C49.63,93,105.79,36.65,156.2,47.55,207.89,58.74,213,  
131.91,264,150c40.67,14.43,108.57-6.91,229-145"
```

Текст в SVG может быть стилизован с помощью CSS

### Задания к лабораторной работе № 14

**Задание 1** Создайте новый документ, в котором разместите текст, используя тег `<text>`, вдоль произвольной кривой, используя `<textPath>`. Данный текст поместить по центру, выделить его произвольным цветом.

Размер шрифта должен составлять 36px. Используя лекцию 9 создать документ style.scss, в котором должны быть представлены стили текста.

**Задание 2** Создайте элементы представленные на рис. 14.1

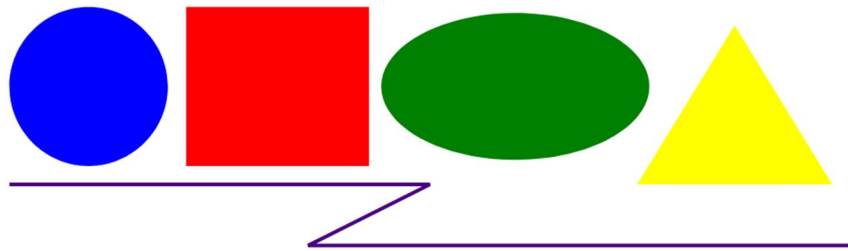


Рис. 14.1

**Задание 2.1** Измените для треугольника желтую заливку на линейную градиентную заливку от зеленого к оранжевому

**Задание 2.2** Для круга задайте произвольный радиальный градиент

**Задание 3** Сделайте элемент на рис. 14.2 используя графический редактор для работы с векторной графикой. Используя тег <path> создайте свою произвольную траекторию, добавив к элементу.

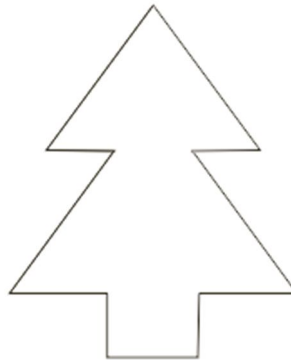


Рис. 14.2

**Задание 4** Откройте svgicon.html файл из папки labs. Используйте любой svg-код иконки из файла и поместите в раздел defs. Разместите этот контур не менее 5-ти раз на страничке при помощи тега use. Задайте различные SCSS-стили и примените их к контурам. Применить к элементам различные трансформации. Пример задания представлен на рис. 14.3

```
<svg width="600" height="600">
<rect fill="linen" width="100%" height="100%" />
<style>
.col-1 { fill: #F35C78; }
.col-2 { fill: #E77D6D; }
.col-3 { fill: #D99B64; }
</style>
<defs>
<path id="house" d="M32 18.451l-16-12.42-16 12.42v-5.064l16-12.42
16 12.42zM28 18v12h-8v-8h-8v8h-8v-12l12-9z"/>
</defs>
<use xlink:href="#house" class="col-1" x="20" y="10"
transform="rotate(35 36 26)"/>
<use xlink:href="#house" class="col-2" x="67" y="10"
transform="rotate(100 83 26)"/>
<use xlink:href="#house" class="col-3" x="114" y="10"
transform="rotate(140 130 26)"/>
</svg>
```

Рис. 14.3



**Задание 5** Сделав предварительно копию документа с элементами из задания 2 анимировать для них следующие свойства

**Задание 5.1** Для эллипса сделать анимацию изменения цвета при щелчке мыши на нем

**Задание 5.2** Для квадрата сделать анимацию контура

**Задание 5.3** Для треугольника при наведении мыши изменение цвета контура

С анимацией контура ознакомьтесь по ссылке в примечании

**Задание 6** В задании 1 лабораторной работы № 1 внизу страницы создать четыре svg-фигуры в виде кругов радиуса 45px. Каждый из них должен быть содержать гиперссылку на задание из лаб. раб. № 1. Для копии документа измените ранее созданные CSS-стили на SCSS.

**Примечание:** создание анимации <https://css-live.ru/articles/rukovodstvo-po-svg-animaciyam-smil.html>

<https://habr.com/ru/post/450924/>

Анимации контура <https://svg-art.ru/?p=1783>

### Контрольные вопросы

1. Дайте понятие SVG? Как расшифровывается аббревиатура?
2. Какие преимущества SVG перед остальными форматами?
3. Как использовать SVG в HTML?
4. Каким образом создать прямую линию и ломанную линию?
5. Каким образом создать прямоугольник и многоугольник?
6. Каким образом создать круг и эллипс?
7. Для чего предназначен тег <path>?
8. Что означают значения в теге <path>?
9. Какие атрибуты относятся к общим?
10. Как создать заливку svg-фигуры?
11. Как изменить цвет линии или контура svg-фигуры?
12. Как изменить цвет толщины контура svg-фигуры?
13. Каким образом преобразовать svg-фигуру?
14. Для чего используется тег <use>?
15. Каким образом использовать графические редакторы для создания svg?
16. Каким образом создать текст в svg?
17. Для чего используется тег <defs>?
18. Каким образом создать градиентную заливку?
19. Каким образом создать анимацию?
20. Какие атрибуты могут быть использованы при создании анимации?
21. Как использовать обработчики событий при анимации svg?
22. Каким образом используется Sass при создании таблиц стилей?
23. Для чего используется viewBox?
24. Для чего используется тег <g>?