SVG - მასშტაბირებადი ვექტორული გრაფიკები

ვიკიწიგნებიდან – სახელმძღვანელოების თავისუფალი ბიბლიოთებიდან
http://ka.wikibooks.org

სარჩევი

შესავალი

რა უნდა იცოდეთ?

სანამ SVG-ს სწავლას შეუდგებოდეთ უნდა იცოდეთ შემდეგი:

- HTML
- ბაზური XML

რა არის SVG?

- SVG ითარგმნება, როგორც მასშტაბირებადი ვექტორული გრაფიკები (Scalable Vector Graphics).
- SVG განსაზღვრავს ვექტორებს - ვებ გვერდებისთვის შექმნილ გრაფიკებს.
- SVG გრაფიკებს განსაზღვრავს XML ფორმატში.
- SVG გრაფიკების ხარისხი არ ეცემა მათი გადიდება/დაპატარავების შემთხვევაში.
- SVG ფაილებში ყოველ ელემენტსა და ატრიბუტს შეუძლია ანიმირება.
- SVG რეკომენდირებულია მსოფლიო ვებ ქსელის კონსორციუმის (W3C) მიერ .
- SVG ჩაშენებულია W3C სტანდარტებში, ისევე როგორც DOM და XLS.

SVG-ს ისტორია და უპირატესობები

Sun Microsystems, Adobe, Apple, IBM და Kodak არიან ზოგიერთი ის კომპანიები, რომლებიც ჩაერთვნენ SVG-ს განსაზღვრაში. SVG-ს უპირატესობები სხვა გაფართოვების სურათებთან მიმართებაში :

- SVG ფაილების წაკითხვა და მოდიფიკაცია შესაძლებელია ძალიან ბევრ პროგრამაში(მათ შორისაა Notepad).
- SVG ფაილები ბევრად პატარა და შეკუმშვადი ფაილებია, ვიდრე JPG და GIF ფაილები.
- SVG სურათები მასშტაბირებადი სურათებია.
- SVG სურათების ამობეჭდვა შესაძლებელია დიდ რეზოლუციებზე მაღალი ხარისხით.
- SVG სურათების ზომის შეცვლა შესაძლებელია. შესაძლებელია შეიცვალოს სურათის ნებისმიერი ნაწილი დამახინჯების გარეშე.
- SVG-ში ტექსტი მონიშვნადია, დაშესაძლებელია ტექსტის მოძებნა (საუკეთესოა რუკების მარკირებისთვის).
- SVG მუშაობს JAVA ტექნოლოგიებთან.
- SVG ფაილები არიან წმინდა XML ფაილები.

SVG-ს მთავარი კონკურენტია Flash.

SVG-ს უდიდესი უპირატესობაა სხვა სტანდარტებთან შეთანხმება(მათ შორის არიან XSL და DOM), რაც ფლეშს არ გააჩნია. Flash შეთანხმებულია კერძო ტექნოლოგიებთან, რომლებიც არ არიან ღია წყაროები.

SVG-ს ნაკლი არის ის, რომ მას ყველაბრაუზერი არ უზრუნველყოფს. Mozilla-ს ბრაუზერები, Firefox და Opera უზრუნველყოფენ SVG-ს. ასევე Microsoft-ის გეგმა არის SVG-ს უზრუნველყოფა.

SVG-ს მომხმარებელთა რიცხვი იზრდება და ამასთანავე Adobe GoLive უზრუნველყოფს მას.

SVG ფაილების დათვარიელება

თუ თქვენი ბრაუზერი არ უზრუნველყოფს SVG ფაილებს, მასინ თქვენ გჭირდებათ SVG viewer.

შენიშვნა : Firefox-ს და Opera-ს გააჩნიათ საკუთარი SVG Viewr-ები. ასე რომ, თუ თქვენ იყენებთ ამ ბრაუზერებს, მოგიწევთ ცალკე ცალკე Viewer-ების დაყენება.

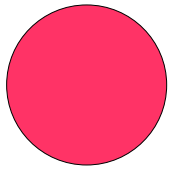
გადმოწერე SVG viewer უფასოდ Adobe-ის ოფიციალური ვებ გვერდიდან - http://www.adobe.com/svg/viewer/install/

SVG მაგალითები

მიმდინარე მაგალითი არის უბრალო SVG ფაილის მაგალითი. SVG ფაილი უნდა დამახსოვრდეს .svg გაფართოვებით :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
```

```
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>
</svg>
```



კოდის ახსნა :

პირველი ხაზი შეიცავს XML დეკლარაციას. ცნობას ავტონომიური ატრიბუტის შესახებ! ეს ატრიბუტი ზუსტად განსაზღვრავს რომ SVG ფაილი ავტონომიურია, თუ ის შეიცავს გარე ფაილების კონფიგურაციებს.

standalone="no" ნიშნავს, რომ SVG შეიცავს გარე ფაილის კონფიგურაციებს, ამ შემთხვევაში DTD - ს.

მეორე და მესამე ხაზები მიმართავენ გარე SVG DTD-ს. DTD-ს ადგილმდებარეობაა "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd". DTD მდებარეობს W3C-ზე და ის შეიცავს SVG-ს ყველა დასაშვებ ელემენტს.

SVG კოდი იწყება <svg> ელემენტით, რომელიც შედგება საწყისი <svg> და სასრული </svg> ტეგებისგან. ეს არის ძირეული ელემენტი. სიგრძის და სიგანის ატრიბუტები დასმულია SVG დოკუმენტის სიგრძესა და სიგანეში. ატრიბუტის ვერსია განსაზღვრავს SVG-ს ვერსიას და xmlns ატრიბუტი განსაზღვრავს SVG-ს სახელის ზოლს.

SVG-ს <circle> ელემენტი გამოყენებულია წრის შესაქმნელად. Cx და cy ატრიბუტები საზღვრავენ წრის ცენტრის x და y კოორდინატებს. თუ Cx და cy გამოტოვებულია, წრის ცენტრის კოორდინატები იქნება (0, 0). R ატრიბუტი საზღვრავს წრის რადიუსს.

Stroke და stroke-width ატრიბუტები აკონტროლებენ, თუ როგორ უნდა გამოისახოს ფიგურის კონტური. კოდში კონტურის სიფართე დასმულია 2 პიქსელზე და საზღვრის ფერი შავზე.

Fill ატრიბუტი მიმართავს ფიგურის შინაგან ფერს. კოდში ფიგურის ფერია - წითელი.

სასრული </svg> ტეგი ხურავს ძირეულ SVG ელემენტს და დოკუმენტს.

შენიშვნა: ყველა გახსნილი ტეგი აუცილებლად უნდა დაიხუროს.

SVG HTML გვერდებში

<EMBED> ტეგის გამოყენება

<embed> ტეგი უზრუნველყოფილია ყველა დიდი ბრაუზერის მიერ.

შენიშვნა: Adobe SVG Viewer იძლევა რეკომენდაციას იმის შესახებ, რომ EMBED ტეგი გამოიყენოთ, მაშინ, როცა SVG-ს ამაგრებთ HTML-ში. თუ გსურთ შექმნათ სწორი XHTML, თქვენ ვერ გამოიყენებთ <embed> ტეგს - <embed> ტეგი არ არის HTML-ის სპეციფიკაციების სიაში შეტანილი.

```
<embed src="rect.svg" width="300" height="100"
type="image/svg+xml"
pluginspage="http://www.adobe.com/svg/viewer/install/" />
```

შენიშვნა: pluginspage ატრიბუტი მიმართავს URL-ს პლაგინის გადმოსაწერად.

წვრილმანი: ინტერნეტ ექსპლორერი უზრუნველყოფს დამატებით ატრიბუტს, wmode="transparent", რომელიც HTML გვერდის ფონს ქმნის სრულიად კაშკაშას.

<OBJECT> ტეგის გამოყენება

<object> ტეგი არის HTML-ის სტანდარტული ტეგი და ის უზრუნველყოფილია ყველა ახალი ბრაუზერის მიერ. არახელსაყრელია ის, რომ ეს ტეგი არ უშვებს სკრიფტინგს.

შენიშვნა: თუ თქვენ დაყენებული გაქვთ Adobe SVG Viewer-ის ბოლო ვერსია, SVG ფაილები არ იმუშავებენ <object> ტეგის გამოყენების შემთხვევაში.

```
<object data="rect.svg" width="300" height="100"
type="image/svg+xml"
codebase="http://www.adobe.com/svg/viewer/install/" />
```

შენიშვნა: codebase ატრიბუტი მიმართავს URL-ს პლაგინის გადმოსაწერად.

<IFRAME> ტეგის გამოყენება

<iframe> ტეგი მუშაობს თითქმის ყველა ბრაუზერში.

```
<iframe src="rect.svg" width="300" height="100">
</iframe>
```

ვისურვებდი...

ვისურვედი SVG ელემენტების პირდაპირ HTML კოდში ჩამატებას, მხოლოდ SVG-ს სახელის ზომის მიმართვისგან, ამის მსგავსად :

```
<html
xmlns:svg="http://www.w3.org/2000/svg">
<body>
<p>This is an HTML paragraph</p>
<svg:svg width="300" height="100" version="1.1" >
<svg:circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red" />
</svg:svg>
</body>
</html>
```

SVG RECT

SVG ფიგურები

SVG-ს გააჩნია ზოგიერთი განსახვავებული ფიგურა, რომლის გამოყენება და მანიპულირება შესაძლებელია შემქმნელების მიერ :

- მართკუთხედი <rect>;
- წრე <circle>;
- ელიფსი <ellipse>;
- ხაზი <line>;
- ტეხილი ხაზი <polyline>;
- პოლიგონი <polygon>;
- ბილიკი <path>.

<RECT> ტეგი

<rect> ტეგი გამოიყენება მართკუთხედის და მართკუთხედის მაგვარი ფიგურების შესაქმნელად.

იმისათვის რომ გაუგოთ სამუშაოს, დააკოპირეთ მიმდინარე კოდი Notepad-ში და დაამახსოვრეთ ფაილი, როგორც "rect1.svg". განათავსეთ ფაილი თქვენს ბრაუზერის დირექტორიაში :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect width="300" height="100"
style="fill:rgb(0,0,255);stroke-width:1;
stroke:rgb(0,0,0)"/>
</svg>
```

კოდის ახსნა:

Rect ელემენტის width და height ატრიბუტები განსაზღვრებიან მართკუთხედის სიგრძე და სიგანეში.

- Style ატრიბუტი გამოიყენება CSS თვისებების განსაზღვრისთვის.

- CSS fill თვისება განსაზღვრავს მართკუთხედის ფერს.
- CSS stroke-width თვისება განსაზღვრავს მართკუთხედის საზღვრის ზომას.
- CSS stroke თვისება განსაზღვრავს მართკუთხედის საზღვრის ფერს.

განიხილეთ სხვა მაგალითი, რომელიც შეიცავს რამოდენიმე ახალ ატრიბუტს :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" width="250" height="250"
style="fill:blue;stroke:pink;stroke-width:5;
fill-opacity:0.1;stroke-opacity:0.9"/>
</svg>
```

კოდის ახსნა:

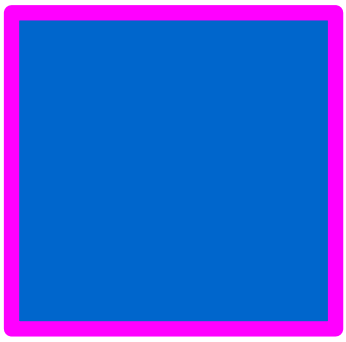
- X ატრიბუტი განსაზღვრავს მართკუთხედის მარცხენა პოზიციას.
- Y ატრიბუტი განსაზღვრავს მართკუთხედის მარჯვენა პოზიციას.
- CSS fill-opacity თვისება განსაზღვრავს ფერის გამჭირვალეობას.
- CSS stroke-opacity განსაზღვრავს შტრიხების ფერის გამჭირვალეობას.

მთლიანი ელემენტის გამჭირვალეობის განსაზღვრა :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" width="250" height="250"
style="fill:blue;stroke:pink;stroke-width:5;
opacity:0.9"/>
</svg>
```

კოდის ახსნა:

CSS opacity თვისება განსაზღვრავს მთლიანი ელემენტის გამჭირვალეობას.



ბოლო მაგალითი, მომრგვალებული კუთხეების მქონე მართხკუთხედის შექმნა :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<rect x="20" y="20" rx="20" ry="20" width="250"
height="100" style="fill:red;stroke:black;
stroke-width:5;opacity:0.5"/>
</svg>
```

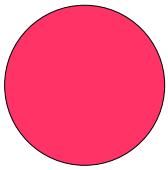
კოდის ახსნა:

rx და ry ატრტიბუტები განსაზღვრავენ მართკუთხედის კუთხეების მომრგვალებას.

CIRCLE ტეგი

<circle> ტეგი გამოიყენება წრის შესაქმნელად.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<circle cx="100" cy="50" r="40" stroke="black"
stroke-width="2" fill="red"/>
</svg>
```



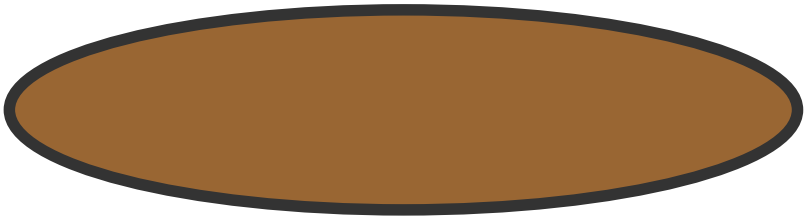
კოდის ახსნა:

- cx და cy ატრიბუტები განსაზღვრავენ წრის ცენტრის x და y კოორდინატებს. თუ cx და cy ატრიბუტები გამოტოვებულნი არიან,მაშინ წრის ცენტრის კოორდინატებია (0, 0).
- R ატრიბუტი განსაზღვრავს წრის რადიუსს.

ELLIPSE ტეგი

<ellipse> ტეგი გამოიყენება ელიფსის შესაქმნელად. ელიფსი და წრე გვანან ერთმანეთს. განსხვავება ისაა, რომ ელიფსის x და y რადიუსები განსხვავდებიან ერთმანეთისაგან.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<ellipse cx="300" cy="150" rx="200" ry="80"
style="fill:rgb(200,100,50);
stroke:rgb(0,0,100);stroke-width:2"/>
</svg>
```



კოდის ახსნა:

- Cx ატრიბუტი განსაზღვრავს ელიფსის ცენტრის x კოორდინატს.
- Cy ატრიბუტი განსაზღვრავს ელიფსის ცენტრის y კოორდინატს.
- Rx განსაზღვრავს ჰორიზონტალურ რადიუსს.
- Ry განსაზღვრავს ვერტიკალურ რადიუსს.

მიმდინარე მაგალითი ქმნის ერთმანეთზე დალაგებულ სამ ელიფსს :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<ellipse cx="240" cy="100" rx="220" ry="30"
style="fill:purple"/>
<ellipse cx="220" cy="70" rx="190" ry="20"
style="fill:lime"/>
<ellipse cx="210" cy="45" rx="170" ry="15"
```

```
style="fill:yellow"/>
</svg>
```

მიმდინარე მაგალითი აერთიანებს ორ ელიფსს :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<ellipse cx="240" cy="100" rx="220" ry="30"
style="fill:yellow"/>
<ellipse cx="220" cy="100" rx="190" ry="20"
style="fill:white"/>
</svg>
```

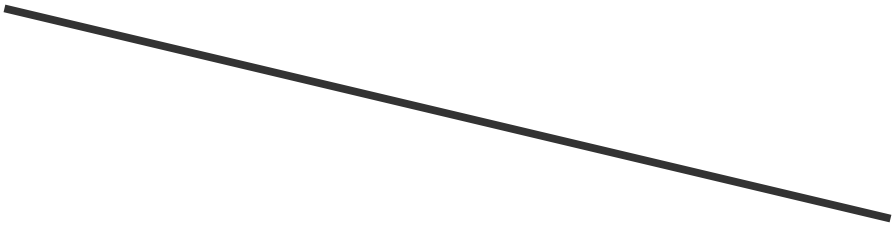

LINE ტეგი

<line> ტეგი გამოიყენება ხაზის შესაქმნელად.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<line x1="0" y1="0" x2="300" y2="300"
style="stroke:rgb(99,99,99);stroke-width:2"/>
</svg>
```

კოდის ახსნა:

- x1 განსაზღვრავს ხაზის დაწყებას x-ღერძზე.
- y1 განსაზღვრავს ხაზის დაწყებას y-ღერძზე.
- X2 განსაზღვრავს ხაზის დაწყებას x2-ღერძზე.
- Y2 განსაზღვრავს ხაზის დაწყებას y2-ღერძზე.



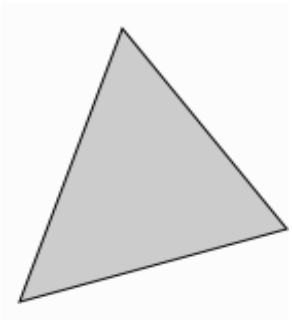
POLYGON ტეგი

<polygon> ტეგი გამოიყენება გრაფიკების შესაქმნელად, რომლებსაც გააჩნიათ სამი, ან მეტი მხარე.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polygon points="220,100 300,210 170,250"
style="fill:#cccccc;
stroke:#000000;stroke-width:1"/>
</svg>
```

კოდის ახსნა:

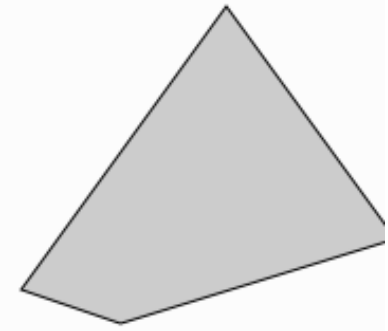
- Points ატრიბუტი განსაზღვრავს პოლიგონის თითო კითხის x და y კოორდინატებს.



მიმდინარე მაგალითი ქმნის პოლიგონს ოთხი მხრით :

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
```

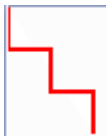
```
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polygon points="220,100 300,210 170,250 123,234"
style="fill:#cccccc;
stroke:#000000;stroke-width:1"/>
</svg>
```



POLYLINE ტეგი

<polyline> ტეგი გამოიყენება სწორი ხაზებისაგან შემდგარი ნებისმიერი ფიგურის შესაქმნელად.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<polyline points="0,0 0,20 20,20 20,40 40,40 40,60"
style="fill:white;stroke:red;stroke-width:2"/>
</svg>
```



PATH ტეგი

<path> ტეგი გამოიყენება ბილიკის განსაზღვრისთვის.

მიმდინარე ბრძანებები დასაშვებიაბილიკის ინფორმაციისთვის :

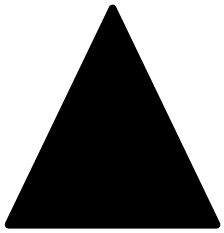
- M = გადაადგილება
- L = ხაზისკენ

- H = ჰორიზონტალური ხაზისკენ
- V = ვერტიკალური ხაზისკენ
- C = მრუდისკენ
- S = გლუვი კუთხისკენ
- Q = კვადრატული კუთხისკენ
- T = გლუვი კვადრატული კუთხისკენ
- A = ელიფსური თაღი
- Z = ბილიკის დახურვა

შენიშვნა: ყველა ბრძანებას შეუძლია აგრეთვე გამოძახოს პატარა ასოებით. მთავრული ასოები ნიშნავს აბსოლიტურ ადგილმდებარეობას, პატარა ასოები ნიშნავს შედარებით ადგილმდებარეობას.

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<path d="M250 150 L150 350 L350 350 Z" />
</svg>
```

ეს მაგალითი საზღვრავს ბილიკს, რომელიც იწყება 250 150 პოზიციიდან ხაზისკენ 150 350 პოზიციასთან ერთად, შემდეგ აქედანვე ხაზისკენ 350 350 და საბოლოოდ იხურება ბილიკი უკნისაკენ 250 150.



მიმდინარე მაგალითი ქმნის სპირალს:

```
<?xml version="1.0" standalone="no"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
"http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="100%" height="100%" version="1.1"
xmlns="http://www.w3.org/2000/svg">
<path d="M153 334
C153 334 151 334 151 334
C151 339 153 344 156 344
C164 344 171 339 171 334
C171 322 164 314 156 314
C142 314 131 322 131 334
```

```
C131 350 142 364 156 364
C175 364 191 350 191 334
C191 311 175 294 156 294
C131 294 111 311 111 334
C111 361 131 384 156 384
C186 384 211 361 211 334
C211 300 186 274 156 274"
style="fill:white;stroke:red;stroke-width:2"/>
</svg>
```