

# Scalable Vector Graphics (SVG) - nowy standard grafiki w sieci WWW

**Format SVG** (ang. *Scalable Vector Graphics*) to stworzona w 1999 roku przez W3C aplikacja XML opisująca dwuwymiarową (2D) statyczną i animowaną grafikę wektorową na stronach WWW. W roku 2001 SVG uzyskał status rekomendacji W3C. Jest on zaimplementowany przez przeglądarkę **Amaya**, której rozwój wspiera W3C, **Mozilla Firefox**, od wersji 1.5, oraz przeglądarkę **Opera**. Pliki SVG mogą być edytowane i oglądane w szeregu nie związanych z przeglądarką programów. Istnieje szansa, że SVG zastąpi format grafiki wektorowej Macromedia Flash. Ma nad nim przewagę, gdyż jest tworzony jako standard przez jedną z najważniejszych organizacji zajmujących się standardami w Internecie (W3C). Prace nad nią są jawne i nie podlegają żadnym prawom patentowym. Do tworzenia grafiki w tym formacie można użyć wolnodostępnych programów, na przykład **Inkscape** i **Sodipodi** (SVG jest ich natywnym formatem).

**XML** (ang. *Extensible Markup Language*) - wzorowany na SGML-u sposób opisu znacznikami, umożliwiający wygodniejsze, szybsze i mniej sformalizowane przygotowywanie wszelkich dokumentów tekstowo-graficznych, które można bez problemów przenosić i adaptować do różnych form przekazu elektronicznego.

XML to język znaczników umożliwiający tworzenie swoich własnych znaczników formatujących. XML jest czymś znacznie więcej niż HTML, gdyż umożliwia pisanie dokumentów, które będą mogły być obsługiwane przez najróżniejsze urządzenia i programy. XML umożliwia też tworzenie tzw. aplikacji - czyli zestawów znaczników do konkretnych zastosowań, z których obecnie najbardziej znane są: XHTML, MathML (*Mathematical Markup Language*), SVG, CML (*Chemical Markup Language*).

Zaletą XML-a jest możliwość dowolnego mieszania tych aplikacji, co umożliwia np. włączanie fragmentów MathML czy SVG w dokumenty napisane w XHTML-u, tworząc jeden, poprawny składniowo dokument XML.

**GRAFIKA BITMAPOWA** (rastrowa) umieszcza dane na siatce pikseli. Składa się z wielu - kwadracików- (pikseli). Grafika -bitmapowa -rastrowa -pikselowa- powstaje na skutek nakładania na siebie plam barwnych. Zalety: dobrze oddaje półcienie, przejścia między kolorami, subtelne cieniowanie obrazów ciągłotonowych. Wady: źle wygląda po dużym powiększeniu (piksele ujawniają kwadracikową strukturę obrazka). Stosuje się do przedstawiania fotografii.

**GRAFIKA WEKTOROWA** przedstawia elementy tworzone wzorami matematycznymi. Grafika -wektorowa -rysunkowa -matematyczna- powstaje na bazie rysunku z linii wektorowych. Zalety: idealnie znosi powiększanie i pomniejszanie utrzymując gładkość linii. Wady: słabo (bez płynności) oddaje przejścia między kolorami. Stosuje się do tworzenia ilustracji, napisów, grafik typu logo, które będą zmieniane wymiarowo.

## Program Inkscape [www.inkscape.org](http://www.inkscape.org)

Inkscape jest programem do tworzenia grafiki wektorowej. Jest wolnym oprogramowaniem rozpo-  
wszechnianym na licencji GNU GPL. Program jest odpowiedzią na niezadowolenie części programi-  
stów z kierunku i postępu prac nad innym podobnym programem - Sodipodi. Obecnie Inkscape  
znacznie przewyższa możliwości swojego pierwowzoru, m. in. obsługuje tworzenie napisów wzdłuż  
linii, poprawiono funkcje eksportu do innych formatów - **EPS** (Encapsulated PostScript), **PS**  
(PostScript), **EPSI** (Encapsulated PostScript Interchange), **AI** (Adobe Illustrator), zwykły **SVG**, **POV**  
(POV-Ray, tylko spline'y), **SVGZ** (skompresowany SVG) - dodano możliwość wektoryzacji  
(trasowanie) rysunków "płaskich" grafiki rastrowej, za pomocą zintegrowanej wtyczki wykorzystują-  
cej program potrace. Zmieniono także interfejs z przypominającego wielookienkowe podejście wzo-

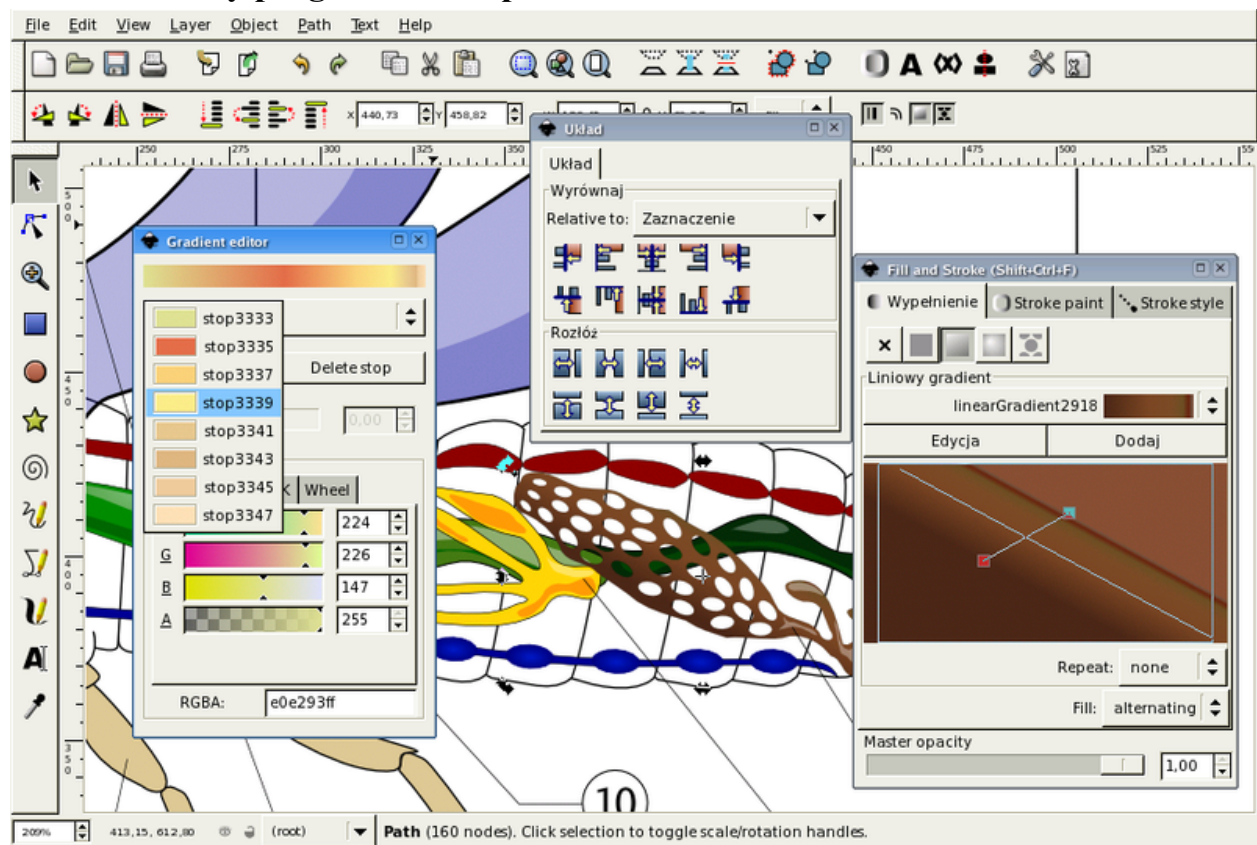
rowane na programie grafiki rastrowej GIMP, na bardziej typowe, integrujące okno narzędzi i pole rysowania.

Macierzystym formatem zapisu plików w Inkscape jest SVG. Program potrafi wyświetlać grafikę używając wygładzania, korzysta z kanału alfa dla uzyskania przezroczystości, pozwala na tworzenie gradientów wypełniających, ma wbudowany zbiór zakończeń linii (np. strzałki), obsługuje czcionki wektorowe. Jego zaletą jest zbiór przykładowych plansz, dostępnych z menu pomocy, prezentujących efekty, jakie można uzyskać w programie wraz z opisami jak tego dokonać.

### Polecane URL's (tutoriale):

- Inkscape Basic Tutorials <http://www.inkscape.org/doc/basic/tutorial-basic.html>
- Inkscape Shapes <http://www.inkscape.org/doc/shapes/tutorial-shapes.html>
- Inkscape Advanced Tutorial <http://www.inkscape.org/doc/advanced/tutorial-advanced.html>
- Adobe SVG Viewer <http://www.adobe.com/svg/>
- XML.com <http://www.xml.com/pub/a/2001/03/21/svg.html>
- W3C SVG (specyfikacja formatu) <http://www.w3.org/Graphics/SVG/>
- Mozilla SVG Project Frequently Asked Questions <http://www.mozilla.org/projects/svg/faq.html>

### Obszar roboczy programu Inkscape



# Zadanie 1. Zalety wektorowego zapisu grafiki

Proszę odwiedzić galerię prac w formacie svg.

Używajmy tylko przeglądarek zgodnych ze zaleceniami W3C

(np. Opera, Mozilla, Mozilla Firefox)

<http://www.croczilla.com/svg/samples/>

Zapiszemy dowolny plik w formacie svg na dysku i określmy jego rozmiar.

np. <http://www.croczilla.com/svg/samples/tiger/tiger.svg>

Następnie otworzymy plik tiger.svg w programie Inkscape

i dokonamy eksportu do z formatu **svg** do mapy bitowej **png** o szerokości 800pix.

Porównajmy wielkości plików svg i png.

**Do jakich zastosowań format svg jest szczególnie predysponowany?**

## Zadanie 2. Samodzielnie opiszemy kodem XML prostą figurę - okrąg

Ponieważ format SVG jest aplikacją XML spóbowemy sami opisać kodem prosty obraz - okrąg.

W dowolnym edytorze tekstowym tworzymy plik: okrag.svg o zawartości jak niżej:

```
<svg>
```

```
<circle style="fill:yellow; stroke:red;" cx="300" cy="300" r="20"/>
```

```
</svg>
```

Nasza figura zaczyna się od znacznika <svg> i kończy </svg> a nasze koło to:

```
<circle style="fill:yellow; stroke:red;" cx="300" cy="300" r="20"/>
```

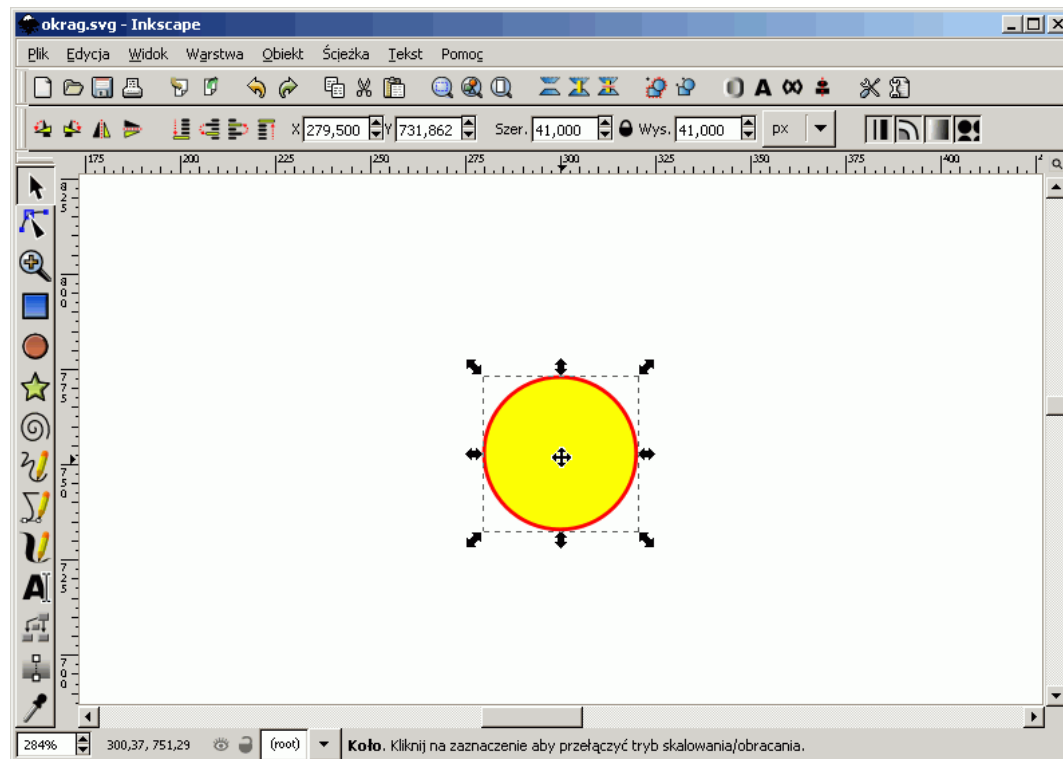
**fill** - wypełnienie (żółte)

**stroke** - kontur (krawędź) w kolorze czerwonym.

Za pomocą atrybutu **R** określa się promień figury. Domyślna wartość to piksele, a więc figura ma 30px średnicy. Atrybuty x i y określają koordynaty naszego obiektu (współrzędne: x - rzędne, y - odcięte).

Otworzymy nasz plik okrag.svg w aplikacji Inkscape.

Przekonajmy się czy wygląd renderowanej w środowisku Inkscape grafiki zgadza się z naszym zamysłem.



Dokonamy korekty kodu okrag.svg. Ponownie edytujemy plik okrag.svg w edytorze tekstowym. Do linii kodu opisującej okrag:

```
<circle style="fill:yellow; stroke:red;" cx="300" cy="300" r="20"/>
```

dołączamy parametr określający grubość linii:

```
<circle style="fill:yellow; stroke:red; stroke-width:5.0" cx="300" cy="300" r="20"/>
```

Ponownie otworzymy nasz plik okrag.svg w aplikacji Inkscape. Co się zmieniło?

## Zadanie 3. Umieszczanie grafiki SVG w dokumencie HTML

Standardy sieciowe umożliwiają mieszanie aplikacji języka XML (np. XHTML, SVG, MathML) w jednym dokumencie.

Można stosować odwołanie do zewnętrznego dokumentu SVG:

```
<a href="plik.svg">kliknij tutaj</a>
```

Wykonajmy prostą stronę w języku HTML, która zawierać będzie kod jak poniżej:

```
<iframe src="okrag.svg" width="400" height="500"
frameborder="0" marginwidth="0" marginheight="0">
```

```
<object data="okrag.svg" type="image/svg+xml">Rysunek SVG
```

```
<embed src="okrag.svg" name="SVGtest" width="400" height="500" type="image/svg+xml" />
```

```
</object>
```

```
</iframe>
```

Zapiszmy naszą stronę jako plik test SVG.htm

Korzystając ze znacznika <IFRAME> - elementu, który definiuje ramkę zawierającą inną stronę WWW, umieścimy nasz obiekt okrag.svg.

Ponieważ format **SVG** należy do rodziny aplikacji **XML** wymagany odpowiedni nagłówek:

```
<?xml version="1.0" standalone="no"?>
```

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

```
<svg xmlns="http://www.w3.org/2000/svg">
```

.... tutaj treść dokumentu SVG

```
</svg>
```

Konieczna jest modyfikacja naszego pliku okrag.svg. Poprawiamy go, dodajemy nagłówek

```
<DOCTYPE>.
```

```
<?xml version="1.0" standalone="no"?>
```

```
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.0//EN" "http://www.w3.org/TR/2001/REC-SVG-20010904/DTD/svg10.dtd">
```

```
<svg xmlns="http://www.w3.org/2000/svg">
```

```
<circle cx="300" cy="300" r="20" style="fill:yellow; stroke:red; " />
```

```
</svg>
```

Otwieramy stronę test SVG.htm . Weryfikujemy wygląd strony w różnych przeglądarkach.

**Mozilla Firefox 1.5** (lub starszy)

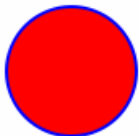
**Internet Explorer 6.0**

**Opera 8.01**(lub starszy)

Umieszczanie grafiki SVG w dokumencie HTML - rozwiązania alternatywne

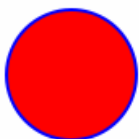
Proszę umieścić rysunek [circle.svg](#) w kodzie HTML używając poniższych rozwiązań.

Użyj znacznika <object>



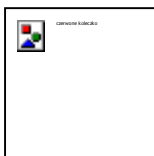
[Sorry, your browser does not support Scalable Vector Graphics.]  
<object data="svg/circle.svg" height="100" width="100" type="image/svg+xml">[Twoja przeglądarka nie obsługuje SVG]</object>

Użyj znacznika <embed>



<embed src="svg/circle.svg" height="100" width="100" alt="czerwone kółeczko" plugin="http://www.adobe.com/svg/viewer/install/">  
Uwaga: Znacznik <embed> nie jest zalecany przez W3C!

Użyj znacznika <img>



 //Poprawne tylko w przeglądarce [Amaya](#)

## Zadanie 4. Tworzenie grafiki SVG z poziomu kodu XML

Spróbujmy samodzielnie - "ręcznie" - stworzyć nieco bardziej złożony obrazek w formacie SVG.

Kod naszej strony obrazek\_SVG.xml niech ma postać:

```
<svg version="1.1"
baseProfile="full"
xmlns="http://www.w3.org/2000/svg">
<rect width="100%" height="100%" fill="white" />
<rect x="30" y="50" width="240" height="100" fill="red" />
<circle cx="150" cy="100" r="80" fill="red" />
<text x="150" y="125" font-size="60" text-anchor="middle" fill="white">SVG</text>
</svg>
```

### inne przykłady:

Proszę sprawdzić samodzielnie poniższe przykłady kodując źródła SVG i werfikując uzyskany obraz w aplikacji Inkscape lub w przeglądarkach WWW zgodnych z W3C

#### koło

```
<circle style="fill: blue; fill-opacity: 1; stroke: green" cx="130" cy="120" r="45" />
```

#### prostokąt

```
<rect style="fill: blue; fill-opacity: .5; stroke: green" x="10" y="20" width="150"
height="70" />
```

#### elipsa

```
<ellipse style="fill: blue; fill-opacity: 1; stroke: green" cx="130" cy="120" rx="250"
ry="100" />
```

#### linia

```
<line style="stroke: red" x1="100" y1="300" x2="300" y2="100" />
```

#### linia łamana

```
<polyline style="fill: none; stroke: blue" points="50,375 150,375 150,325 250,325 250,375" />
```

gdzie: points: współrzędne x,y kolejnych wierzchołków oddzielone przecinkami

### wielokąt

```
<polygon style="stroke:blue; stroke-width:10" points="210,46 227,96 281,97 238,129 254,181 210,150 166,181 182,129 139,97 193,97" />
```

### tekst

```
<text x="10" y="20" style="font-family: serif; font-size: 30"> Hello, <tspan style="fill:red">world</tspan> </text>
```

podelement **tspan** zaznacza fragment tekstu, umożliwiając np. jego odrębne sformatowanie

**Łączenie obiektów w grupy** ( wspólna manipulacja połączonymi obiektami )

```
<g id="kolo_i_prostokat">
```

```
<circle style="fill: blue; fill-opacity: 1; stroke: green" cx="130" cy="120" r="45" />
```

```
<rect style="fill: blue; fill-opacity: .5; stroke: green" x="10" y="20" width="150" height="70" />
```

```
</g>
```

### Modyfikacja położenia i transformacje obiektów

```
<g transform="translate(-10,-20) scale(2) rotate(45)" >
```

```
<circle style="fill: blue; fill-opacity: 1; stroke: green" cx="130" cy="120" r="45" />
```

```
<rect style="fill: blue; fill-opacity: .5; stroke: green" x="10" y="20" width="150" height="70" />
```

```
</g>
```

## Wnioski

1. SVG Scalable Vector Graphics to **najpoważniejszy rywal Flasha** w dziedzinie grafiki wektorowej w sieci. Był on projektowany, tak by sprostać wymaganiom stawianym przez projektantów i programistów tworzących aplikacje sieciowe. Pozbawiony jest wielu wad, którymi obarczony jest format Macromedii.
2. **SVG to język znacznikowy zgodny ze specyfikacją XML**. Fakt ten ma duże znaczenie dla jego rozwoju: ponieważ dokumenty SVG mają postać zrozumiałą dla człowieka, łatwo jest je edytować ręcznie.
3. Stosunkowo łatwo (i zarazem tanio) można stworzyć **edytor czy walidator** składni dokumentów SVG.
4. **SVG współpracuje z istniejącymi standardami** wykorzystywanymi do konstruowania serwisów WWW, takimi jak HTML, języki skryptowe (JavaScript, VBScript), ASP czy XML. Stwarza to nadzieję na pojawienie się dynamicznych, multimedialnych serwisów sieciowych z prawdziwego zdarzenia. Obecnie ich egzystencja zależy od niedoskonałych jeszcze narzędzi, takich jak Flash Generator czy Swift.
5. SVG może wpłynąć na **usprawnienie indeksowania zasobów multimedialnych** przez wyszukiwarki internetowe. Dzisiaj szybko rosnąca liczba witryn wyposażonych we flashowe animacje skutecznie opiera się próbom indeksowania treści przez mechanizmy wyszukiujące. Trudno dziś ocenić, jakie możliwości niesie ten nowy standard. Są już przeglądarki SVG, które poprawnie renderują obrazy statyczne (z uwzględnieniem tekstur, gradientów i przezroczystości), zaś kolejne ich wersje powinny obsługiwać animację i interaktywne elementy sceny. Największe możliwości w tej dziedzinie stoją dzisiaj przed działającymi przeglądarkami internetowymi, mają one bowiem rozbudowane "zaplecze" programistyczne w postaci parsera XML, obsługi języków skryptowych i dostępu do źródeł danych (np. poprzez mechanizm databinding).