



SVG

SVG

SVG to skrót od Scalable Vector Graphics. Jest to format graficzny, w którym kształty są określone w języku XML. XML jest następnie renderowany przez przeglądarkę SVG. Obecnie większość przeglądarek internetowych może wyświetlać SVG, tak jak mogą wyświetlać PNG, GIF i JPG.

Użytkownicy przeglądarki Internet Explorer 8 (i wcześniejszych) mogą być zmuszeni do zainstalowania Adobe SVG Viewer, aby móc przeglądać SVG w przeglądarce. IE 9+, Chrome i Firefox obsługują natywnie SVG.

SVG służy do dwuwymiarowej grafiki wektorowej. Dla trójwymiarowego wyszukiwania formatu X3D.

SVG

SVG jest grafiką wektorową - szczególnie przydatną w przypadku diagramów typu wektorowego, takich jak:

Wykresy dwuwymiarowe w układzie współrzędnych X, Y.

Wykresy kolumnowe, wykresy kołowe itp.

Skalowalne ikony i logo dla aplikacji internetowych, tabletów i urządzeń mobilnych oraz aplikacji internetowych.

Diagramy architektury i projektowania

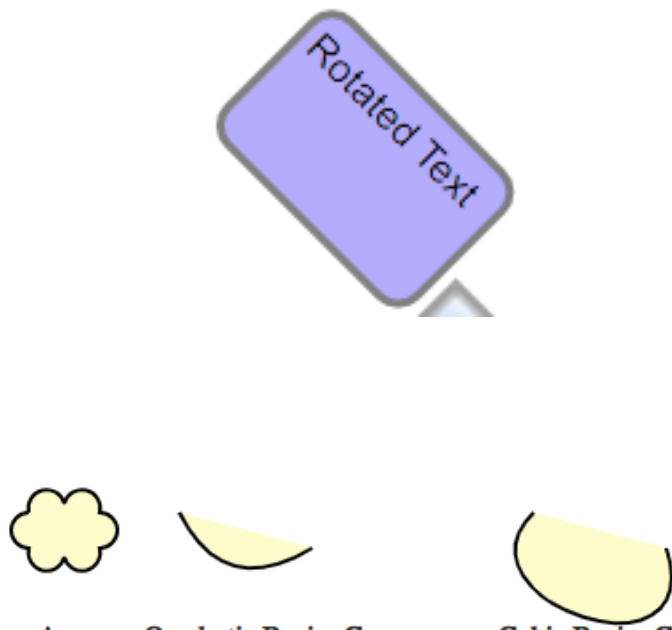
Będąc formatem grafiki wektorowej kształty, które mają być wyświetlane, są przechowywane jako wektory lub struktury wektorowe. Innymi słowy, jako liczby. Nie jako piksele.

Bycie "skalowalnym" oznacza, że widz może skalować obraz SVG w górę i w dół bez utraty jakości. Jest to możliwe, ponieważ grafiki są definiowane jako liczby zamiast pikseli. Skalowanie obrazu SVG w górę lub w dół oznacza tylko pomnożenie lub podzielenie liczb określających kształty SVG.

Format SVG nie jest idealny dla grafiki bitmapowej, takiej jak zdjęcia, filmy itp., Chociaż można osadzić grafikę bitmapową w obrazie SVG. Może to być przydatny sposób rysowania kształtów lub tekstu na szczycie obrazu bitmapowego.

SVG

Przykłady

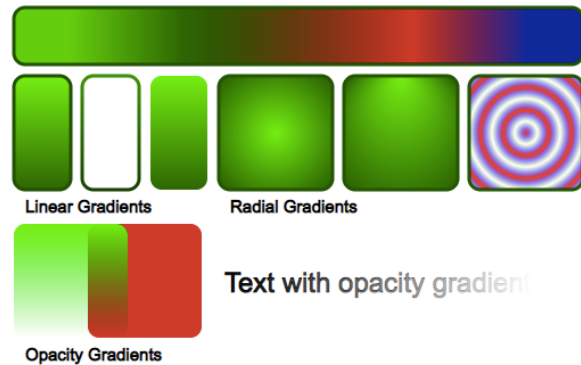


Text

Text

Text

Text along a curved pa



Text with opacity gradient

Przykłady

```
<svg>  
  
<rect x="10" y="10" height="100" width="100"  
  
      style="stroke:#ff0000; fill: #0000ff"/>  
  
</svg>
```



Przykład pokazuje prosty prostokąt o szerokości 100 pikseli i wysokości 100 pikseli. Kolor obrysu (kontur prostokąta) jest ustawiony na kolor HTML # ff0000 - czerwony. Kolor wypełnienia jest ustawiony na # 0000ff - niebieski.

Wyświetlanie SVG

Wyświetlanie SVG w przeglądarkach takich jak Chrome, Firefox i Internet Explorer można wyko-

Wskaż przeglądarce adres URL pliku SVG.

Umieść SVG na stronie HTML

Możesz osadzić obraz SVG w pliku HTML na kilka sposobów:

Używanie elementu iframe

Używanie elementu img

Używanie obrazu SVG jako obrazu tła.

Używanie elementu svg

Używanie elementu osadzania

SVG

iframe

Ponieważ przeglądarki mogą wyświetlać obrazy SVG po wpisaniu adresu URL do obrazu SVG, można również umieścić obraz SVG na stronie HTML za pomocą elementu `iframe`. Oto przykład:

```
<iframe src = „przyklad" width = "200" height = "200">
```

img

Osadzanie obrazu SVG za pomocą elementu `img` odbywa się tak jak każdy inny rodzaj obrazu. Na

```
<img src = "przyklad.jsp">
```

Obraz SVG jest następnie wyświetlany na stronie HTML, tak jak każdy inny obraz.

SVG

SVG jako obrazy tła

Ponieważ przeglądarki obsługują obrazy SVG, tak jak obrazy bitmapowe, możesz użyć obrazów SVG jako obrazów tła za pomocą CSS. Oto przykład:

```
div {  
    background-image: url ('przyklad.svg');  
    background-size: 100px 100px;  
}
```

Może być konieczne ustawienie rozmiaru tła dla obrazu SVG, aby poinformować przeglądarkę, jak je skalować.

SVG

svg wewnątrz elementu HTML

Osadzanie obrazu SVG za pomocą elementu svg może być wykonane przez osadzenie elementu SVG bezpośrednio na stronie HTML, na przykład:

```
<div>  
<svg>  
  <circle cx = "40" cy = "40" r = "24" styl = "stroke: # 006600; fill: # 00cc00" >  
  </ svg>  
</ div>
```

Element div jest właśnie tutaj, aby zilustrować, że element svg może być osadzony bezpośrednio w HTML. Element svg nie musi jednak być osadzony w elemencie div.

Za pomocą elementu SVG można osadzić SVG bezpośrednio na stronie HTML, zamiast umieszczać obraz SVG w osobnym pliku. Możesz ustawić szerokość i wysokość obrazu SVG, dodając atrybuty szerokości i wysokości do elementu `svg`.

Za pomocą elementu SVG można również wygenerować SVG bezpośrednio w przeglądarce przy użyciu JavaScript. Interfejs API JavaScript D3 jest w tym bardzo dobry. JQuery JavaScript API też może to zrobić.

Za pomocą elementu `svg` możesz także stylizować `svg` i jego elementy potomne za pomocą CSS, tak jak w przypadku każdego innego HTML. Zwróć uwagę, że elementy SVG czasami mają różne nazwy dla niektórych swoich właściwości CSS niż elementy HTML.

Używanie SVG jako obrazów tła elementów HTML

Możesz użyć obrazów SVG jako obrazów tła elementów HTML za pomocą właściwości CSS obrazu tła. Wystarczy wskazać plik obrazu SVG, tak jak w przypadku dowolnego innego pliku obrazu. Nie wszystkie przeglądarki mogą jeszcze w pełni to wspierać, aby przetestować je w przeglądarkach, które zamierzasz obsługiwać. Oto przykład:

```
.nazwa_klasy {  
    background: url (/przyklad.svg);  
}
```

Ikony SVG to obrazy SVG, które są używane jako ikony lub przyciski obrazów w aplikacji internetowej lub aplikacji mobilnej. Ikony SVG mogą być również używane w przypadku logo.

Zalety ikony SVG

Używanie ikon SVG dla ikon ma tę zaletę, że ikony można łatwo przeskalować w górę i w dół, w zależności od tego, w którym miejscu aplikacji mają być wyświetlane, oraz od rozmiaru ekranu, na którym jest wyświetlana aplikacja. Ikony SVG mają tę przewagę nad grafiką bitmapową, że nadal wyglądają dobrze po skalowaniu w górę lub w dół. Grafika bitmapowa ma tendencję do pikselizacji po skalowaniu i utraty jakości (pikseli) po zmniejszeniu.

Używanie ikon SVG w aplikacjach internetowych

Przy wyświetlaniu ikon SVG najłatwiej jest użyć elementu `img` HTML, aby wyświetlić ikony. Element `img` HTML ułatwia skalowanie ikony SVG w górę i w dół. Przykład

```
<img src = "przyklad.svg">
```

Aby przeskalować ikonę SVG w górę i w dół, po prostu użyj właściwości CSS `width` lub `height`. Oto przykładowy element `img` z dodaną właściwością stylu wysokości CSS:

```
<img src = „przyklad.svg" style = "height: 50px">
```

Aby zachować proporcje ikony SVG podczas skalowania w górę lub w dół, należy ustawić tylko wartość szerokości lub wysokości - nie obie. Ustawiając tylko szerokość dla jednej z właściwości, przeglądarka odpowiednio przesunie ikonę SVG wzdłuż drugiej osi, aby ikona SVG zachowała proporcje.

SVG

SVG - tworzenie obszaru roboczego

Aby utworzyć obszar roboczy, na którym będziemy pracowali należy elementowi svg nadać wymiary, tj.:

```
<svg width="2000px" height="2000px">
```

```
</svg>
```

Tworzenie własnych ikon SVG

Czasami może być konieczne utworzenie własnych ikon SVG. Ikona SVG jest po prostu obrazem SVG zawartym we własnym pliku SVG. Oto bardzo prosta ikona koła, wykonana z elementem koła SVG:

```
<svg>
```

```
<circle cx="150" cy="50" r="50" style="fill:red;"></circle>
```

```
</svg>
```



SVG

PAMIĘTAJ:

ustal wartość width i height na svg. Jeżeli tego nie zrobisz, a będziesz miał za dużą grafikę (lub kilka grafik) to mogą one nie zmieścić się w obszarze roboczym!

```
<svg width=„5000px” height=„%000px”>
```

```
<circle cx="150" cy="50" r="50" style="fill:red;"></circle>
```

```
</svg>
```



Matematyczny / kartezjański układ współrzędnych

W normalnym układzie współrzędnych kartezjańskich punkt $x = 0$, $y = 0$ znajduje się w lewym dolnym rogu wykresu. Gdy x rośnie, punkty przesuwają się w prawo w układzie współrzędnych. Gdy x zmniejsza się, punkty przesuwają się w lewo w układzie współrzędnych. Gdy y wzrasta, punkty przesuwają się w górę w układzie współrzędnych. Gdy y maleje, punkty przesuwają się w dół w układzie współrzędnych.

Oto ilustracja normalnego układu współrzędnych grafu z 0,0 w lewym dolnym rogu:

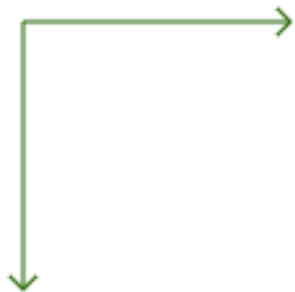


SVG

System współrzędnych SVG

W systemie współrzędnych SVG punkt $x = 0$, $y = 0$ jest lewym górnym rogiem. Oś y jest zatem od

Oto ilustracja układu współrzędnych SVG z 0,0 w lewym górnym rogu:



SVG

Możesz określić, co odpowiada 1 jednostce w systemie współrzędnych SVG. Robisz to wszędzie, gdzie możesz określić współrzędne (pozycja x i y, szerokość i wysokość itp.). Możesz określić jednostki po wartości, na przykład 10 cm lub 125 mm. Jeśli nie określisz żadnych jednostek po wartości współrzędnych, przyjmuje się, że jednostką są piksele (px).

Oto lista jednostek, których możesz używać z elementami SVG:

em

px

pkt Punkty (1/72 cala)

pc Picas (1/6 cala)

cm

mm

in cale

Zagnieżdżanie SVG w SVG

Zagnieżdżanie elementów SVG może być przydatne do grupowania kształtów SVG razem i ustawiania ich jako kolekcji. Wszystkie kształty zagnieżdżone wewnątrz elementu `svg` zostaną umieszczone (x, y) w stosunku do pozycji (x, y) otaczającego elementu `svg`. Przesuwając współrzędne X i Y otaczającego elementu `svg`, przesuwasz również wszystkie zagnieżdżone kształty.

Oto przykład, w którym dwa prostokąty są zagnieżdżone w dwóch elementach `svg`. Z wyjątkiem kolorów dwa prostokąty mają te same definicje dla x , y , wysokości i szerokości. Obejmujące elementy `svg` mają różne wartości x . Ponieważ pozycja x prostokątów jest interpretowana względem ich otaczających elementów `svg` elementów x , dwa prostokąty są wyświetlane w różnych pozycjach x .

SVG

```
<svg>

  <
    x="10">

    <rect x="10" y="10" height="100" width="100" style="stroke:#ff0000; fill: #0000ff"/>

  </
    >

  <
    x="200">

    <rect x="10" y="10" height="100" width="100" style="stroke:#00cc00"/>

  </
    >

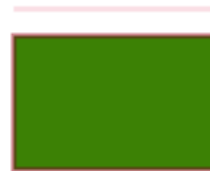
</svg>
```



SVG

Element SVG `<g>` służy do grupowania kształtów SVG. Po zgrupowaniu możesz przekształcić całą grupę kształtów tak, jakby był to pojedynczy kształt. Jest to zaleta w porównaniu do zagnieżdżonego elementu `<svg>`, który sam nie może być celem transformacji.

Możesz również stylizować pogrupowane elementy i ponownie używać ich tak, jakby były pojedynczym elementem.



Zgrupowany tekst

SVG

```
<svg>
```

```
<g>
```

```
<line x1="10" y1="10" x2="85" y2="10"
```

```
style="stroke: pink;"/>
```

```
<rect x="10" y="20" height="50" width="75"
```

```
style="stroke: brown; fill: green"/>
```

```
<text x="20" y="120" style="stroke: yellow; fill: blue">
```



Zgrupowany tekst

SVG

Ten przykład pokazuje 3 kształty zgrupowane w elemencie `<g>`. Jak to jest tutaj wymienione, nie ma szczególnych korzyści w tej grupie. Ale zobacz, co się stanie, gdy poprosimy o przekształcenie `<g>` -elementu. Oto kod:

```
<g transform="rotate(45 50 50)">
```



Dziedziczenie właściwości CSS

Przeanalizuj kod

```
<svg>  
  
<g style="stroke: red; stroke-width: 10px; fill: blue">  
  
  <rect x="10" y="10" width="100" height="50" />  
  
  <circle cx="150" cy="35" r="25" />  
  
  <circle cx="250" cy="35" r="25" style="stroke: black; fill: green;" />  
  
</g>
```

```
</svg>
```

Prostokąt:

Element SVG `<rect>` reprezentuje prostokąt. Za pomocą tego elementu możesz narysować prostokąty o różnej szerokości, wysokości, innym obrysie (konturze) i wypełnieniu kolorów, ostrymi lub zaokrąglonymi narożnikami itp.

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">

  <rect x="10" y="10" height="100" width="100"
        style="stroke:#006600; fill: #00cc00"/>

</svg>
```



Zaokrąglone rogi:

Na prostokątach można narysować zaokrąglone rogi. Atrybuty rx i ry określają, jak będą zaokrąglone rogi. Atrybut rx określa, jak szerokie jest zaokrąglenie, a ry określa, jak wysokie będzie zaokrąglenie. Oto trzy prostokąty z wartościami rx i ry ustawionymi na 5, po 10 i po 15. Zwróć uwagę na różne rozmiary w zaokrągleniu.

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">

  <rect x="10" y="10" height="50" width="50"
        rx="5" ry="5"
        style="stroke:#006600; fill: #00cc00"/>

  <rect x="70" y="10" height="50" width="50"
```



Zaokrąglone rogi:

W tych przykładach rx i ry mają ustawione takie same wartości w każdym prostokącie. Jeśli ustawisz tylko atrybut rx, atrybut ry otrzyma taką samą wartość jak rx. Jest to skrót do definiowania równomiernie zaokrąglonych narożników.

Oto dwa przykłady, w których atrybuty rx są ustawione na 10, ale atrybuty ry są ustawione na 5 i 15. To pokaże, jak zaokrąglone prostokąty wyglądają z różną wysokością i szerokością zaokrąglenia.

SVG

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">

  <rect x="10" y="10" height="50" width="50"
        rx="10" ry="5"
        style="stroke:#006600; fill: #00cc00"/>

  <rect x="130" y="10" height="50"
        rx="10" ry="15"
        style="stroke:#006600; f

</svg>
```



SVG

Styl obrysu (kontur) prostokąta można za pomocą właściwości stylu obrysu SVG. Ten przykład us

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" link">
```

```
<rect x="20" y="20" width="100" height="100"
```

```
style="stroke: #009900;
```

```
stroke-width: 3;
```

```
fill: none; " />
```

```
stroke-dasharray: 10 5;
```

```
</svg>
```



Przetestuj:



SVG

Możesz również uczynić wypełnienie przezroczyste za pomocą właściwości stylu wypełnienia-nieprzezroczystości. Ten przykład pokazuje dwa prostokąty, jeden częściowo na drugim, a górny półprzezroczysty:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<rect x="20" y="20" width="100" height="100"
```

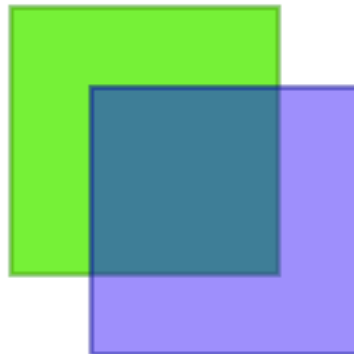
```
style="stroke: #009900;
```

```
fill: #33ff33;
```

```
"
```

```
/>
```

```
<rect x="50" y="50" width="100" height="100"
```



SVG

Okrąg

Element SVG <circle> służy do rysowania okręgów. Oto prosty przykład:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <circle cx="40" cy="40" r="24" style="stroke:#006600; fill:#00cc00"/>  
</svg>
```



SVG

Okrąg

Można ustawić obrys (kontur) okręgu SVG przy użyciu właściwości stylu obrysu SVG. W pierwszym

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <circle cx="40" cy="40" r="24" style="stroke:#006600; stroke-width: 3; fill:#00cc00"/>  
</svg>
```



SVG

Okrąg

Zauważ, że obrys jest teraz szerszy (grubszy) niż w pierwszym przykładzie pokazanym na tej stronie. Możesz także narysować obrys linią przerywaną, korzystając z właściwości `stroke-dasharray`. Oto

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <circle cx="40" cy="40" r="24" style="stroke:#006600; stroke-width: 3; stroke-dasharray: 1  
</svg>
```



SVG

Okrąg

Właściwość typu wypełnienia kontroluje sposób wypełniania okręgu. Możesz nie wypełniać wca

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <circle cx="40" cy="40" r="24" style="stroke: #006000; fill:none"/>  
</svg>
```



SVG

Okrąg

Właściwość typu wypełnienia kontroluje sposób wypełniania okręgu. Możesz nie wypełniać wca

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

Wypełnienie:

```
<circle cx="40" cy="40" r="24" style="stroke: #006000; fill:none"/>
```

```
</svg>
```

```
<circle cx="40" cy="40" r="24"
```

```
style="stroke: #660066;
```

```
fill: #00ff00"/>
```



SVG

Okrąg

Można również ustawić wypełnienie jako przezroczyste za pomocą właściwości stylu wypełnienia

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <circle cx="40" cy="40" r="24" style="stroke: #660000; fill: #cc0000" />  
  <circle cx="64" cy="40" r="24" style="stroke: #000066; fill: #0000cc" fill-opacity: 0.5  
</svg>
```



Elipsy

Element SVG <ellipse> służy do rysowania elips. Elipsa to okrąg, który nie ma równej wysokości

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <ellipse cx="40" cy="40" rx="30" ry="15" style="stroke:#006600; fill:#00cc00"/>  
</svg>
```



Elipsy

Elipsa jest wyśrodkowana w cx , cy jak koło. Ale promień w kierunkach x i y jest określony przez dwa atrybuty, a nie jeden: atrybuty rx i ry . Jak widać, atrybut rx ma wyższą wartość niż atrybut ry , dzięki czemu elipsa jest szersza niż jest wysoka. Ustawienie atrybutów rx i ry na tę samą liczbę spowoduje regularne koło.

Elipsy

Możesz ustawić szerokość obrysu elipsy za pomocą właściwości stylu szerokość-szerokość. Oto przykład:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <ellipse cx="50" cy="50" rx="40" ry="30" style="stroke: #ff0000; stroke-width: 5;  
  fill: none"/>
```

wypróbuj

```
</svg>
```

```
stroke-dasharray: 10 5;
```

```
stroke-opacity: 0.5;
```



SVG

Linia

Linia zaczyna się w punkcie ustawionym przez atrybuty x1 i y1.

Linia kończy się w punkcie ustawionym przez atrybuty x2 i y2.

Atrybut stylu określa kolor i grubość obrysu (linii). Atrybuty stylu są bardziej szczegółowo omówione w innym tekście.

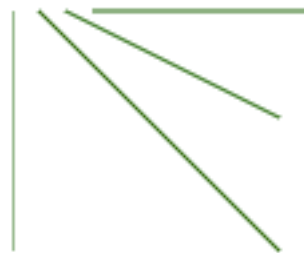
```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<line x1="0" y1="10" x2="0" y2="100" style="stroke:#006600;"/>
```

```
<line x1="10" y1="10" x2="100" y2="100" style="stroke:#006600;"/>
```

```
<line x1="20" y1="10" x2="100" y2="50" style="stroke:#006600;"/>
```

```
<line x1="30" y1="10" x2="110" y2="10" style="stroke:#006600;"/>
```



SVG

Polilinia

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <polyline points="0,0 30,0 15,30" style="stroke:#006600;"/>  
</svg>
```



SVG

Polilinia

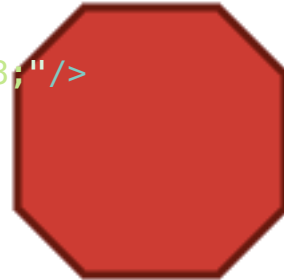
```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<polygon points="50,5 100,5 125,30 125,80 100,105
```

```
50,105 25,80 25, 30"
```

```
style="stroke:#660000; fill:#cc3333; stroke-width: 3;"/>
```

```
</svg>
```



Ścieżki

Pierwsze polecenie rysowania wewnątrz atrybutu <ścieżka> d powinno zawsze być poleceniem p

Ten przykład przenosi pióro wirtualne do punktu 50,50. Następne polecenie rysowania rozpocz

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<path d="M50,50"
```

```
style="stroke:#660000; fill:none;"/>
```

```
</svg>
```



Ścieżki

Rysowanie linii jest prawdopodobnie najprostszym poleceniem, które możesz podać jako element `<path>`. Rysowanie linii odbywa się za pomocą poleceń `L` i `l`. Oto przykład:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<path d="M50,50
```

```
l100,100"
```



Ten przykład rysuje linię od punktu 50,50 (punkt polecenia M) do punktu 100 100 (punkt polecenia l).

```
</svg>
```

Ścieżki

Różnica między komendami L i l polega na tym, że wersja wielka (L) rysuje linię do punktu absolutnego przekazywanego do polecenia, podczas gdy wersja wersji pisanej małymi literami (l) rysuje linię do punktu względnego przekazanego do polecenia. Punktem względnym jest punkt wirtualnego pióra przed rozpoczęciem linii oraz współrzędne podane komendzie l.

Jeśli wirtualny długopis znajduje się na 50,50 i użyjesz polecenia l100, 100, linia zostanie narysowana do $50 + 100, 50 + 100 = 150, 150$. Użycie polecenia L100, 100 spowodowałoby dokładne narysowanie linii do 100,100, niezależnie od położenia wirtualnego pióra.

Ścieżki

Łuki

Rysowanie łuków za pomocą elementu `<path>` odbywa się za pomocą A i poleceń. Podobnie jak w przypadku wierszy, polecenie wielkiej litery (A) używa bezwzględnych współrzędnych dla punktu końcowego, gdzie polecenie małe litery (a) używa względnych współrzędnych (względem punktu początkowego). Oto przykład:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<path d="M50,50
```

```
A30,50 0 0,1 100,100"
```

Ten przykład rysuje łuk od punktu 50,50 do punktu 100,100 (podanego jako ostatni w poleceniu A).

```
stroke="green", fill:none; </> >
```



SVG

```
<svg>
```

```
<g style="stroke: red; stroke-width: 10px; fill: blue">
```

```
<rect x="10" y="10" width="100" height="50" />
```

```
<circle cx="150" cy="25" r="25" />
```

```
<circle cx="250" cy="25" r="25" style="stroke: green;" />
```

```
</g>
```

```
</svg>
```



SVG

Możliwość przekształcania wszystkich kształtów w elemencie `<g>` jest zaletą w porównaniu do grupowania.

Element `<g>` ma jednak jedną wadę w porównaniu do elementu `<svg>`. Nie można przesuwać elementów.

Jeśli chcesz przenieść wszystkie kształty w elemencie `<g>` za pomocą atrybutów `x` i `y`, musisz zagnieść je.

SVG

```
<svg xmlns="http://www.w3.org/2000/svg"
xmlns:xlink="http://www.w3.org/1999/xlink">

    <    x="10">

        <rect x="10" y="10" height="100" width="100"

style="stroke:#ff0000;fill: #0000ff"/>

    </    >

    <    x="200">

        <rect x="10" y="10"

style="stroke:#009900;fill: #00cc00"/>

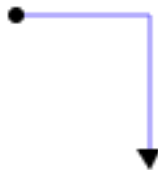
    </    >
```



Markery

Znaczniki te są tworzone przy użyciu elementu `<marker>`. Element `<marker>` musi być zagnie

SVG



```
<svg>
```

```
  <defs>
```

```
    <marker id="markerCircle" markerWidth="8" markerHeight="8" refX="5" refY="5">
```

```
      <circle cx="5" cy="5" r="3" style="stroke: none; fill:#000000;"/>
```

```
    </marker>
```

```
    <marker id="markerArrow" markerWidth="13" markerHeight="13" refX="2" refY="6" orient="a
```

```
      <path d="M2,2 L2,11 L10,6 L2,2" style="fill: #000000;" />
```

```
    </marker>
```

```
  </defs>
```



Przede wszystkim zwróć uwagę na element `<defs>` z dwoma elementami `<marker>` wewnątrz. Dwa elementy `<marker>` definiują znaczniki początku i końca pokazane na poprzednim obrazie.

Po drugie, zauważ, że element `<path>` odwołuje się do dwóch elementów `<marker>` z wnętrza jego atrybutu stylu, używając właściwości CSS znacznika start i końca znacznika. W ten sposób określasz, które znaczniki mają być użyte dla danej ścieżki.

Definiowanie markera

Znacznik definiujesz za pomocą elementu `<marker>`. Oto przykład:

```
<marker id="markerCircle" markerWidth="8" markerHeight="8" refX="5" refY="5">  
  <circle cx="5" cy="5" r="3" style="stroke: none; fill:#000000;"/>  
</marker>
```



```
<marker id="markerCircle" markerWidth="8" markerHeight="8" refX="5" refY="5">  
  <circle cx="5" cy="5" r="3" style="stroke: none; fill:#000000;"/>  
</marker>
```

Ten przykład definiuje znacznik o szerokości 8 (markerWidth = "8"), o wysokości 8 (markerHeight = "8"). Szerokość i wysokość są niezbędne do wyraźnego ustawienia, ponieważ znacznik jest oddzielnym elementem graficznym.

Atrybut id elementu <marker> służy do odniesienia do tego znacznika z elementów <path>.

Jako punkty odniesienia należy używać zestawów współrzędnych refX i refY, które wskazują punkt wewnątrz znacznika. Punktem odniesienia jest to, co jest pozycjonowane na początku ścieżki za pomocą znacznika. W tym przykładzie refX i refY są ustawione na środku koła, co oznacza, że środek koła zostanie umieszczony na początku ścieżki.

Wewnątrz elementów <marker> znajduje się element <circle>. Element koła definiuje się środkiem (cx i cy) przy 5,5. znaczników MarkWidth i markerHeight.

Orientacja

Oto kolejny przykład definicji <marker>. Ten przykład definiuje trójkąt używany jako strzałka dla ś

```
<marker id="markerArrow" markerWidth="13" markerHeight="13" refX="2" refY="6"
    orient="auto">
    <path d="M2,2 L2,11 L10,6 L2,2" style="fill: #000000;" />
</marker>
```

<path> wewnątrz elementu <marker> rysuje trójkąt z końcówką skierowaną w prawo. Jeśli jednak ścieżka nie jest linią poziomą, potrzebny jest obrót trójkąta, tak aby pasował do kierunku ścieżki, która go używa. Dokonujesz tego, ustawiając atrybut orient na auto. To spowoduje obrócenie kształtu wewnątrz elementu <marker> w celu dopasowania do ścieżki, która się do niego odwołuje.

Odwoływanie się do znacznika ze ścieżki

Możesz odwoływać się do znacznika ze ścieżki, korzystając z następujących właściwości CSS:

marker-start

marker-mid

marker-end

Te trzy właściwości CSS przypisują znacznik do początku, środka i końca ścieżki.

Właściwości CSS muszą znajdować się wewnątrz atrybutu stylu używanego elementu `<path>`. Odwołujesz się do elementu `<marker>`, odwołując się do jego atrybutu `id`:

`marker-start: url (#markerId);`

Znacznik IDd powinien zostać zastąpiony przez wartość atrybutu `id` elementu `<marker>` do odniesienia.

SVG

Tekst:

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">
```

Ten przykład definiuje tekst ustawiony na $x = 10$ i $y = 40$. Tekst do wyświetlenia to "Przykładowy tekst".

SVG

Tekst:

Pozycja tekstu jest określona przez atrybuty x i y elementu <text>. Atrybut x określa gdzie zlokalizować

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">

  <text x="20" y="40">Example SVG text 1</text>

  <line x1="10" y1="40" x2="150" y2="40" style="stroke: #000000"/>
```

</svg>



Zakotwiczenie tekstu określa, która część tekstu jest umieszczona w pozycji x określonej w atrybucie x elementu tekstowego. Domyślnie zakotwiczeniem tekstu jest lewa krawędź tekstu. Ale możesz też użyć środka tekstu jako kotwicy lub prawej krawędzi - końca tekstu.

Możesz ustawić właściwość CSS text-anchor, aby ustawić zakotwiczenie tekstu. Może przyjmować trzy wartości: początek, środek i koniec. Oto przykład kotwicy tekstowej SVG pokazujący trzy różne opcje zakotwiczenia tekstu:

```
<text x="50" y="20"
      style="text-anchor: start">
  Start
</text>

<text x="50" y="40"
      style="text-anchor: middle">
  Middle
```

Obracanie tekstu:

Możliwe jest obracanie tekstu SVG, tak jak można obrócić dowolny inny kształt SVG. Obracasz te

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
  <text x="20" y="40"
        transform="rotate(30 20,40)"
        style="stroke:none; fill:#000000;"
        >Tekst pod kątem</text>
```



Tspan

Element SVG `<tspan>` służy do rysowania wielu linii tekstu w SVG. Zamiast konieczności pozycjonowania każdej linii tekstu absolutnie, element `<tspan>` umożliwia pozycjonowanie linii tekstu w stosunku do poprzedniego wiersza tekstu. Element `<tspan>` umożliwia także użytkownikowi zaznaczanie kopiowanie i wklejanie kilku wierszy tekstu na raz, zamiast jednego elementu tekstowego.

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

```
xmlns:xlink="http://www.w3.org/1999/xlink">
```

`tspan linia 1` `tspan linia 2`

```
<text x="20" y="15">
```

```
<tspan>tspan line 1</tspan>
```

```
<tspan>tspan line 2</tspan>
```



Tspan

Jeśli chcesz, aby linie były ustawione względem siebie pionowo, możesz to zrobić za pomocą atrybutu dy (delta y):

```
<svg xmlns="http://www.w3.org/2000/svg"
      xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<text x="20" y="15">
```

```
  <tspan>tspan line 1</tspan>
```

```
  <tspan dy="10">tspan line 2</tspan>
```

```
</text>
```

tspan linia 1
tspan linia 2



Tspan

Aby umieścić tekst relatywnie na osi X, można użyć atrybutu dx (delta x). Poniższy przykład pokazuje wpływ ustawienia dx na 30. Zauważ, że druga linia tekstu jest teraz wyświetlana w 30 pikselach względem końca pierwszego wiersza tekstu (nie na początku):

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  
    <text x="20" y="15">  
        <tspan>tspan linia 1</tspan>  
        <tspan dx="30" dy="10">tspan linia 2</tspan>  
    </text>
```

tspan linia 1 tspan linia 2

< >

Tspan

Aby umieścić tekst relatywnie na osi X, można użyć atrybutu dx (delta x). Poniższy przykład pokazuje wpływ ustawienia dx na 30. Zauważ, że druga linia tekstu jest teraz wyświetlana w 30 pikselach względem końca pierwszego wiersza tekstu (nie na początku):

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  
    <text x="20" y="15">  
        <tspan>tspan linia 1</tspan>  
        <tspan dx="30" dy="10">tspan linia 2</tspan>  
    </text>
```

tspan linia 1 tspan linia 2

< >

Stylowanie

Możliwe jest samodzielne stylizowanie elementów `<tspan>`. W ten sposób można użyć elementu `<tspan>`, aby styl bloku tekstu różnił się od reszty tekstu. Oto przykład:

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">  
  <text x="10" y="20">  
    Przykładowy <tspan style="font-weight: bold;">wytluszczone</tspan> tekst.  
  </text>  
</svg>
```

Tekst na ścieżce

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
  <defs>
```

```
    <path id="myTextPath2"
```

```
      d="M75,20 l100,0 l100,30 q0,100 150,100"/>
```

```
  </defs>
```

```
  <text x="10" y="100" style="stro
```

```
    <textPath xlink:href="#myTex
```

```
      Jakiś próbny tekst stwor
```

```
    </textPath>
```

Jakiś próbny tekst stworzony tylko po to,

za nakładanie te

SVG

Przeanalizuj przykład

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<rect x="10" y="10" height="130" width="500" style="fill: #000000"/>
```

```
<image x="20" y="20" width="300" height="80"
```

```
  xlink:href="https://images.pexels.com/photos/290386/pexels-photo-290386.jpeg?auto=compress"
```

```
<line x1="25" y1="80" x2="350" y2="80"
```

```
  style="stroke: #ffffff; stroke-width: 3;"/>
```

Obrazek jako link

```
<svg xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink">
```

```
<a xlink:href="/svg/index.html" target="_top">
```

```
<rect x="10" y="20" width="75" height="30"
```

```
style="stroke: #333366; fill: #6666cc"/>
```

```
</a>
```



```
</svg>
```



Dziękujemy za uwagę