

Il formato SVG

- **SVG** (Scalable Vector Graphics) è un formato basato su **XML** (quindi, testuale)
- Il documento XML descrive un **albero**
- I **nodi** dell'albero descrivono **oggetti geometrici** o **trasformazioni** da applicare agli oggetti sottostanti
- Il disegno vettoriale viene **renderizzato** visitando l'albero e disegnando le figure contenute nei nodi

Il formato SVG

- I file SVG sono (sempre) editabili
- Possono essere manipolati con appositi editor grafici (o anche a mano)
- Possono essere renderizzati dai principali browser WWW, e adottano i principali standard del web (XML, CSS)
 - si tratta quindi del formato più indicato se le immagini vettoriali devono essere usate su pagine web

Richiami di XML

- Un documento XML è costituito da un insieme di *nodi*
- Ciascun nodo ha un nome (**tag**), uno o più **attributi**, e opzionalmente un **contenuto**
- Sintassi:

$\langle \text{tag attr}_1 = \text{val}_1 \dots \text{attr}_n = \text{val}_n \rangle$

contenuto

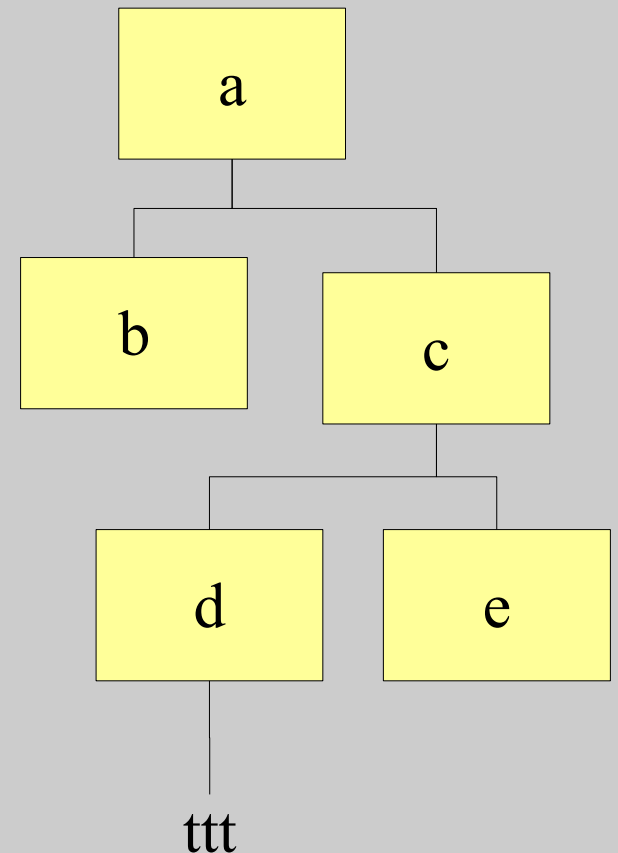
$\langle / \text{tag} \rangle$

$\langle \text{tag attr}_1 = \text{val}_1 \dots \text{attr}_n = \text{val}_n / \rangle$

Richiami di XML

- Il contenuto di un nodo può consistere di altri nodi
- Si viene così a creare una struttura ad albero

```
<a>  
  <b/>  
  <c>  
    <d>  
      ttt  
    </d>  
  </c>  
</a>
```



Un esempio di SVG

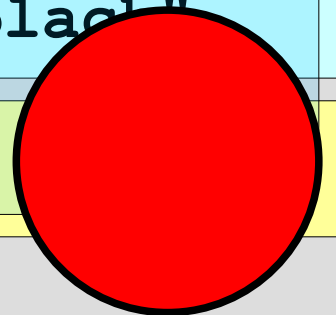
```
<?xml version="1.0" standalone="no"?>
```

```
<!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>
```



Attributi e stile

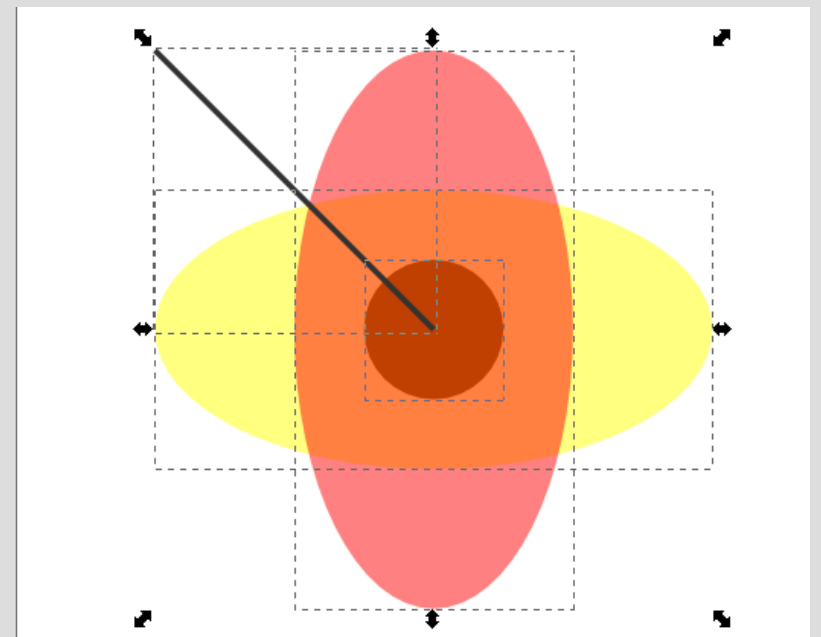
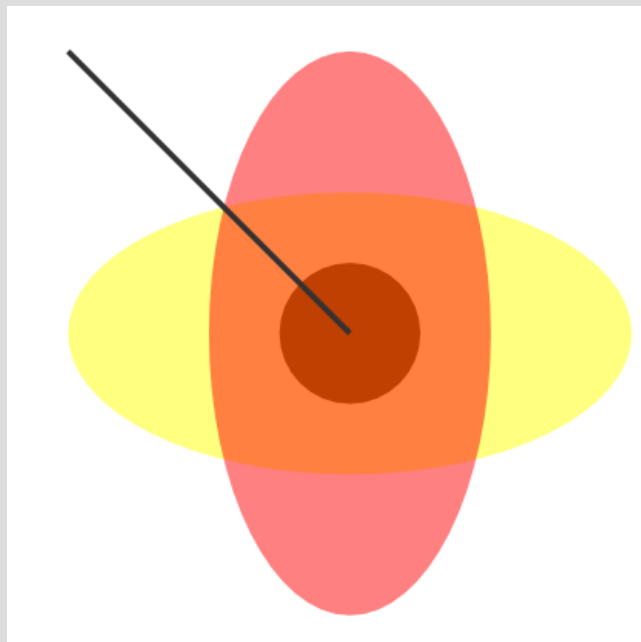
- Molti attributi in SVG possono essere espressi sia come argomenti del tag che come elementi di stile
- Esempio di argomenti:
 - `<circle ... fill="red">`
- Esempio di stile:
 - `<circle ... style="fill: red;">`
- L'uso degli stili è del tutto analogo al CSS che già conoscete!

Attributi e stile

- L'insieme degli attributi definibili in uno stile è assai vasto
- Fra gli altri:
 - colori, gradienti
 - trasparenza / opacità
 - larghezze, colore e tratteggio delle linee
 - font, dimensione, spaziatura dei testi
 - ...

Esempio

```
<?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="300" cy="300" r="50" color="blue"/>  
<ellipse cx="300" cy="300" rx="200" ry="100" style="fill:yellow; opacity:0.5"/>  
<ellipse cx="300" cy="300" rx="100" ry="200" style="fill:red; opacity:0.5"/>  
<line x1="100" y1="100" x2="300" y2="300" style="stroke:rgb(50,50,50); stroke-width:4"/>  
</svg>
```



Elementi semplici

- Rettangolo **<rect>**
- Cerchio **<circle>**
- Ellisse **<ellipse>**
- Linea **<line>**
- Polilinea **<polyline>**
 - sequenza di segmenti di retta
- Poligono **<polygon>**
 - area delimitata da una polilinea
- Path **<path>**
 - costruito più generale (lo conosciamo già)
- Testo **<text>**

L'elemento <path>

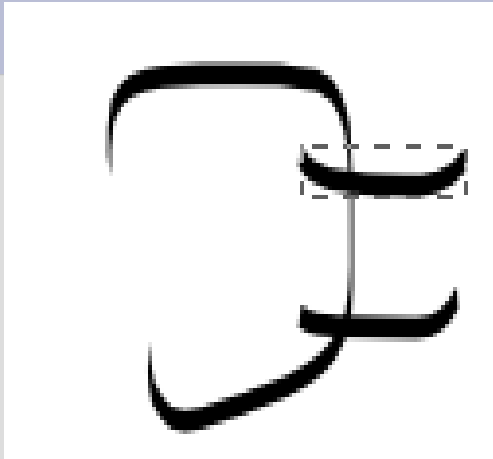
- L'elemento <path> ha un argomento **d** che ha per valore un insieme di comandi che definiscono il path
- Ogni comando è dato da una lettera alfabetica, che esprime il tipo di comando, seguito da 0, 1 o più coordinate (secondo il comando) espresse numericamente e separate da spazi o virgole
 - Esempio:
`<path d="M100,100 L150,300 Z"/>`

L'elemento <path>

- M = moveto
- L = lineto
- H = horizontal lineto
- V = vertical lineto
- C = curveto
- S = smooth curveto
- Q = quadratic Bezier curveto
- T = smooth quadratic Bezier curveto
- A = elliptical Arc
- Z = closepath

La lettera maiuscola indica coordinate assolute; la stessa lettera in minuscolo indica coordinate relative (ovvero, lo spostamento rispetto al punto corrente)

Esempio di <path>



Nuovo documento 1: Editor XML (Maiusc+Ctrl+X)

xml abc xml

```
<svg:svg id="svg2">
  <svg:defs id="defs4">
    <sodipodi:namedview id="base">
      <svg:metadata id="metadata7">
        <svg:g id="layer1">
          <svg:path id="path1872">
            <svg:path id="path1874">
              <svg:path id="path1876">
```

Attributo	Valore
d	M 255.39644,126.37462 C 259.03551,136.09127 266.75217,138.00022 275.6448,141.40514
id	path1874
style	fill:black;fill-opacity:1;fill-rule:nonzero;stroke:none;stroke-width:1px;stroke-linecap:butt;stroke-linejoin:miter;stroke-opacity:1

d

Imposta

M 255.39644,126.37462 C 259.03551,136.09127 266.75217,138.00022 275.6448,141.40514 C 293.4822,145.29751 311.84468,145.3711 330.01779,145.66986 C 348.60621,147.33804 361.56251,143.05754 374.10009,130.42224 L 375.35195,144.04601 C 362.16195,156.40765 348.88365,160.90129 329.99722,159.12746 C 311.7075,158.86888 293.24317,158.79184 275.25498,155.09744 C 265.97466,151.79279 257.9653,149.62269 253.17499,140.40033 L 255.39644,126.37462 z |

Selezionato l'attributo **d**. Premere **Ctrl+Enter** quando finito per effettuare i cambiamenti.

Gruppi

- Si possono definire **gruppi** con il tag `<g> ... </g>`
- I gruppi possono avere proprietà, che vengono **ereditate** da tutti i membri del gruppo (ovvero, nodi nel sottoalbero radicato nel gruppo)
- Inoltre, i gruppi possono avere un nome (proprietà `id="..."`) che può essere usato per riferire l'intero gruppo

Filtri

- SVG include la possibilità di definire **filtri**
 - I filtri operano sull'immagine in maniera del tutto analoga a quanto abbiamo visto per la grafica raster...
 - ... ma non modificano realmente l'immagine: vengono applicati solo al momento del disegno finale!
- A ogni tipo di elemento possono essere applicati uno o più filtri (nell'ordine desiderato)

Filtri

- Un filtro è definito da:
 - un nome
 - una sequenza di tipi di filtro (filter element)
 - ciascun filter element ha i propri parametri
- I filtri vanno definiti all'interno di un elemento **<defs>**
- Possono poi essere richiamati con un attributo di stile **filter**

Filtri

- SVG fornisce i seguenti filter element:

feBlend

feMerge

feColorMatrix

feMorphology

feComponentTransfer

feOffset

feComposite

feSpecularLighting

→ feConvolveMatrix

feTile

feDiffuseLighting

→ feTurbulence

feDisplacementMap

feDistantLight

feFlood

fePointLight

→ feGaussianBlur

feSpotLight

feImage

Esempio di filtri

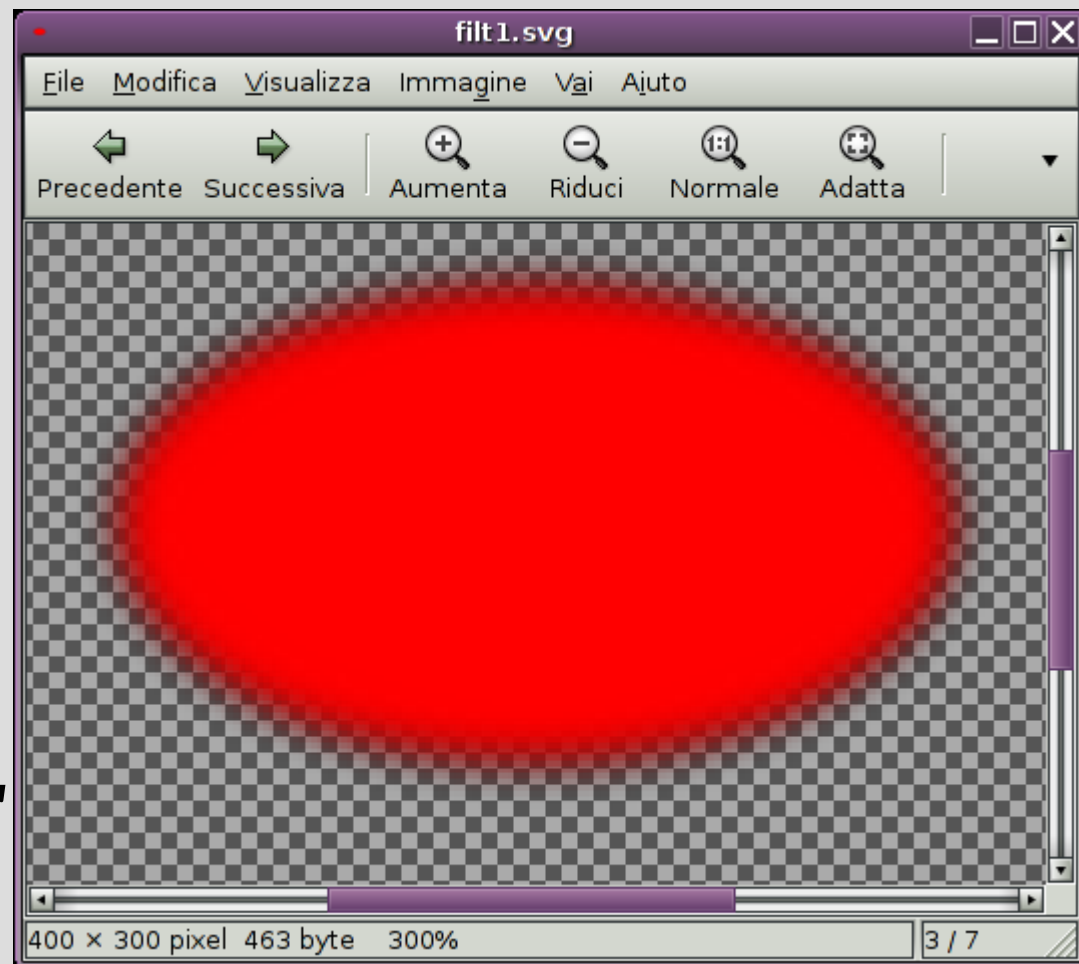
- Esempio: definiamo un filtro di *blur* (diffusione) con distribuzione Gaussiana
- In pratica, rendere progressivamente semitrasparenti (sfumati) i bordi dell'oggetto a cui è applicato

```
<defs>
<filter id="sfumato">
<feGaussianBlur
  in="SourceGraphic"
  stdDeviation="3" />
</filter>
</defs>
```

Esempio di filtri

- Ora possiamo applicare il filtro a un elemento grafico

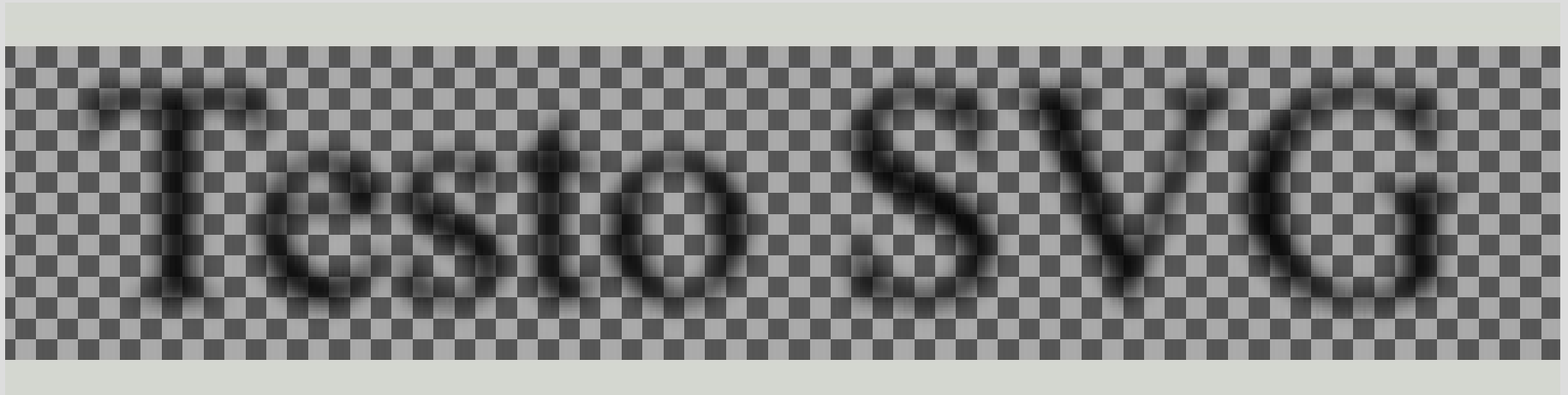
```
<ellipse  
cx="200"  
cy="150"  
rx="70"  
ry="40"  
style="  
  fill:#ff0000;  
  stroke:#000000;  
  stroke-width:2;  
  filter:url(#sfumato) "  
>
```



Esempio di filtri

- Una volta definito, un filtro può essere usato su qualunque elemento grafico

```
<text x="100" y="100" font-size="120"  
color="blue" style="filter:url(#sfumato)">  
Testo SVG  
</text>
```



Esempio di filtri

```
<?xml version="1.0"?>
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN"
  "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg width="7.5cm" height="5cm" viewBox="0 0 200 120"
  xmlns="http://www.w3.org/2000/svg" version="1.1">
<defs>
<filter id="MyFilter" filterUnits="userSpaceOnUse" x="0" y="0" width="200" height="120">
  <feGaussianBlur in="SourceAlpha" stdDeviation="4" result="blur"/>
  <feOffset in="blur" dx="4" dy="4" result="offsetBlur"/>
  <feSpecularLighting in="blur" surfaceScale="5" specularConstant=".75"
    specularExponent="20" lighting-color="#bbbbbb"
    result="specOut">
    <fePointLight x="-5000" y="-10000" z="20000"/>
  </feSpecularLighting>
  <feComposite in="specOut" in2="SourceAlpha" operator="in" result="specOut"/>
  <feComposite in="SourceGraphic" in2="specOut" operator="arithmetic"
    k1="0" k2="1" k3="1" k4="0" result="litPaint"/>
  <feMerge>
    <feMergeNode in="offsetBlur"/>
    <feMergeNode in="litPaint"/>
  </feMerge>
</filter>
</defs>

<rect x="1" y="1" width="198" height="118" fill="#888888" stroke="blue" />
<g filter="url(#MyFilter)" >
  <g>
    <path fill="none" stroke="#D90000" stroke-width="10"
      d="M50,90 C0,90 0,30 50,30 L150,30 C200,30 200,90 150,90 z" />
    <path fill="#D90000"
      d="M60,80 C30,80 30,40 60,40 L140,40 C170,40 170,80 140,80 z" />
    <g fill="FFFFFF" stroke="black" font-size="45" font-family="Verdana" >
      <text x="52" y="76">SVG</text>
    </g>
  </g>
</g>
</svg>
```



Esempio di filtri

immagine base



```
<filter id="MyFilter" filterUnits="userSpaceOnUse" x="0" y="0"
width="200" height="120">
  <desc>Produces a 3D lighting effect.</desc>
  <feGaussianBlur in="SourceAlpha" stdDeviation="4"
    result="blur"/>
  <feOffset in="blur" dx="4" dy="4" result="offsetBlur"/>
  <feSpecularLighting in="blur" surfaceScale="5"
    specularConstant=".75" specularExponent="20"
    lighting-color="#bbbbbb" result="specOut">
    <fePointLight x="-5000" y="-10000" z="20000"/>
  </feSpecularLighting>
  <feComposite in="specOut" in2="SourceAlpha" operator="in"
    result="specOut"/>
  <feComposite in="SourceGraphic" in2="specOut"
    operator="arithmetic" k1="0" k2="1" k3="1" k4="0"
    result="litPaint"/>
  <feMerge>
    <feMergeNode in="offsetBlur"/>
    <feMergeNode in="litPaint"/>
  </feMerge>
</filter>
```

Definizioni: l'elemento <defs>

- L'uso di <defs> che abbiamo visto per i filtri è generalizzabile
- Qualunque elemento SVG può essere inserito all'interno di <defs>
- All'elemento si assegna un nome con l'attributo `id="..."`
- L'elemento definito può quindi essere riusato ogni volta che è necessario

Definizioni: esempio

```
<defs>
  <text id="saluto">Ciao, Pippo!</text>
  <path id="curva" d="M10.5 203 C15.948 197.239 87.7676 153.791 102.5
158 C119.025 162.722 182.543 202 196.5
202 C202.451 202 242.889 237.817 297.5 156"/>
</defs>
<text x="100" y="100" style="word-spacing:3; font-size:24">
  <tref xlink:href="#saluto">
</text>
<text style="font-size:18">
  <textPath xlink:href="#curva">
    <tref xlink:href="#saluto">
  </textPath>
</text>
```

Gradienti

- SVG fornisce la possibilità di usare un **gradiente** come riempimento
 - Un gradiente è definito da un tipo (lineare o radiale), un colore di partenza (con trasparenza), zero o più *stop* intermedi (colore/trasparenza) e un colore finale (con trasparenza; anch'esso è uno *stop*)
 - Servono inoltre informazioni di tipo geometrico: direzione del gradiente nel caso di gradienti linear, centro e raggio nel caso di gradienti radiali

Gradienti

```
<defs>
<linearGradient id="fiamma" x1="0%" y1="0%" x2="0%"
y2="100%">
<stop offset="0%" style="stop-color:rgb(255,255,0);
stop-opacity:1"/>
<stop offset="100%" style="stop-color:rgb(255,0,0);
stop-opacity:0.5"/>
</linearGradient>
</defs>

<ellipse
  cx="200" cy="190"
  rx="85" ry="95"
  style="fill:url(#fiamma)"/>
```



Trasformazioni

- SVG consente di applicare **trasformazioni** affini a elementi o gruppi
 - Come già visto, le trasformazioni vengono applicate moltiplicando fra loro le matrici di trasformazione relative
- Esiste anche qui il concetto di **matrice di trasformazione corrente**
 - Le trasformazioni all'interno di un sottoalbero sono applicate a partire da quella della radice del sottoalbero

Trasformazioni

- Esempio:

```
<g transform="translate(10,20) scale(2)
rotate(30) scale(1,3)">
  disegno di altri elementi
</g>
```

- La matrice può essere arbitraria:

```
<g transform="matrix(a,b,c,d,e,f)">
  disegno di altri elementi
</g>
```

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix}$$

Trasformazioni

- Lo spazio di disegno iniziale è dato da un rettangolo con $(0,0)$ **in alto a sinistra**
 - a differenza del Postscript, dove $(0,0)$ è in basso a sinistra
- L'estensione è nel tag **`<svg width=... height=...>`**
 - può essere espressa in unità, pixel o in percentuale rispetto al contenitore (per il web)
 - si tratta comunque di valori *arbitrari*: l'importante è che poi il disegno sia espresso con le stesse coordinate

Animazioni e scripting

- SVG costruisce un proprio **Document Object Model (DOM)**
- Come in DHTML, gli attributi del DOM possono essere cambiati da programma
 - per esempio, con una funzione JavaScript
- Usando questa tecnica, si possono realizzare *animazioni* in SVG
- Si possono anche costruire disegni interattivi!
 - onclick(), onmouseover(), onscroll(), ...

Inkscape: un editor SVG

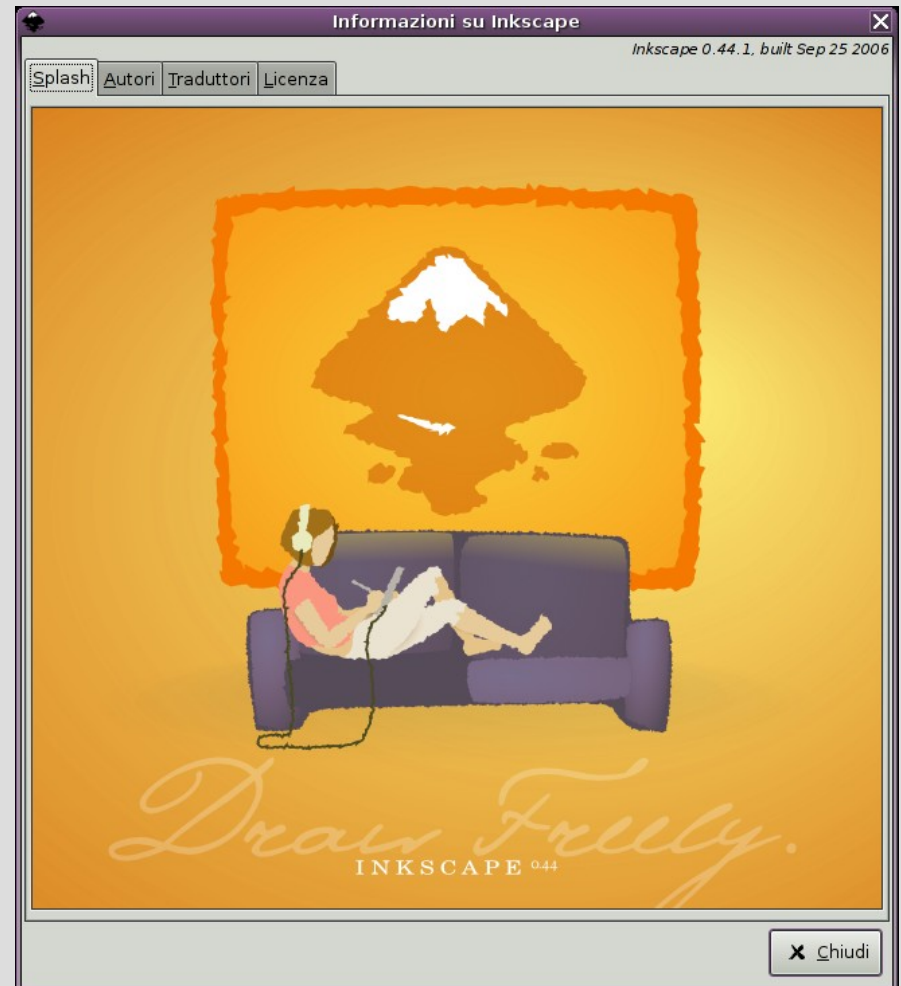
- Inkscape è un editor di grafica vettoriale basato su SVG
 - Altri editor di uso comune sono basati su Postscript o su modelli proprietari
- Non si tratta di una conversione: Inkscape mantiene in memoria un modello XML del documento SVG, e lo modifica man mano che l'utente manipola l'immagine
 - Si può vedere il testo XML che cambia quando si modifica il disegno, e anche il disegno che cambia quando si modifica l'XML!

Inkscape: un editor SVG

- Vari strumenti consentono di creare elementi di tutti i tipi che abbiamo visto e di assegnarne le proprietà
- Consente poi di
 - manipolare oggetti e tracciati con varie operazioni
 - inserire e modificare testo, farlo fluire dentro i tracciati, trasformarlo in tracciato
 - vettorizzare disegni raster

Inkscape: un editor SVG

- L'interfaccia utente, benché ragionevolmente intuitiva, è troppo complessa per essere spiegata con i lucidi...
- Lo vedremo in azione in laboratorio!



Riferimenti

- Lo standard SVG è su <http://www.w3.org/TR/SVG11/index.html>
 - si riferisce a SVG 1.1, la versione corrente. La versione 1.2 è in lavorazione
- Adobe offre un tutorial su SVG alla URL <http://www.adobe.com/svg/basics/intro.html>
- Un tutorial semplificato (e altro materiale) sono disponibili alla pagina <http://www.w3schools.com/svg/default.asp>
- Non mancano anche dei libri su SVG: per esempio, in Italiano, *SVG*, Francesco-Saverio Caccavella, McGraw-Hill (ma fate una visita in libreria, ci saranno testi più recenti)
- La home page di Inkscape è <http://www.inkscape.org>; da qui è possibile scaricare le versioni per Linux, Windows e Macintosh
 - Il menu “Aiuto” di Inkscape contiene una serie di lezioni e trucchi che guidano alle funzioni del programma
 - Un libro on-line su Inkscape è disponibile liberamente all'URL <http://tavmjong.free.fr/INKSCAPE/MANUAL/html/index.php>