SUPSI

Human-computer interaction

HCI 2018-2019

anno accademico 2018-2019

Nicola Rizzo - nicola.rizzo@supsi.ch

“XML is crap. Really. There are no excuses. XML is nasty to parse for humans, and it's a disaster to parse even for computers. There's just no reason for that horrible crap to exist.”

–Linus Torvalds, 2014

SVG

(Scalable Vector Graphics)

Viewport

- La viewport è la “finestra” attraverso la quale si osserva il documento SVG

- È assegnabile attraverso gli attributi width e height, che possono avere le consuete unità di misura oppure nessuna (user units, cioè pixel), o percentuali

Inclusione

- direttamente nel documento html, via **<svg>...</svg>**
- come immagine
- come background
- come **<object>**
- come <embed>

SUPSI

Inclusione

- è preferibile usare i due metodi perché sono standard e permettono di accedere al documento svg via css e via script

Sistema di coordinate (viewport)

SUPSI

- L’angolo 0 0 è quello in alto a sinistra
- L’asse X è positivo verso destra, l’asse Y verso il basso
- Le unità di misura sono i pixel della viewport
- Le dimensioni della viewport possono essere stabilite anche via css (height e width)

Sistema di coordinate (utente)

- Inizialmente coincide con il sistema di coordinate della viewport

- L'attributo **viewBox** permette di modificarlo (viene detto *sistema di coordinate corrente* o *user space in uso*)

viewBox

- viewBox = <min-x> <min-y> <width> <height>
- agendo sull’attributo viewBox è possibile zoomare avanti e indietro il documento SVG o eseguire una crop (con valori inferiori agli attributi della viewport)
- min-x e min-y possono essere negative, width e height no (e quando queste ultime due valgono 0 il documento è invisibile)

preserveAspectRatio

- ha effetto solo se viewBox è definita
- preserveAspectRatio = <align> <meetOrSlice>?
- per default viene mantenuta l’aspect ratio
- un valore “none” per align fa in modo che l’SVG venga scalato senza mantenere le proporzioni
- meetOrSlice: meet (default) si comporta come background-size:contain (css), l’immagine viene scalata in modo da stare tutta nella viewport, se la ratio è la stessa; se la ratio di viewport e viewBox è diversa, ci saranno delle parti non coperte dalla viewBox
- meetOrSlice: slice funziona come background-size:cover: se le ratio non sono le stesse, la viewBox viene tagliata e la viewport resta tutta coperta in modo

preserveAspectRatio / meetOrSlice = meet

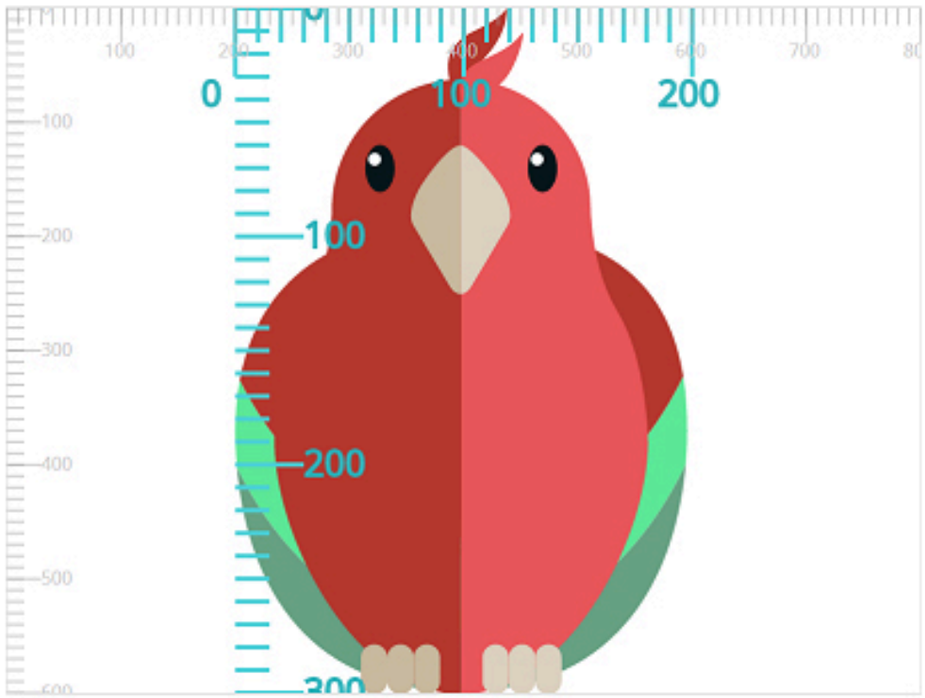
SUPSI

- meetOrSlice: meet (default) si comporta come background-size:contain (css), l'immagine viene scalata in modo da stare tutta nella viewport, se la ratio è la stessa; se la ratio di viewport e viewBox è diversa, ci saranno delle parti non coperte dalla viewBox

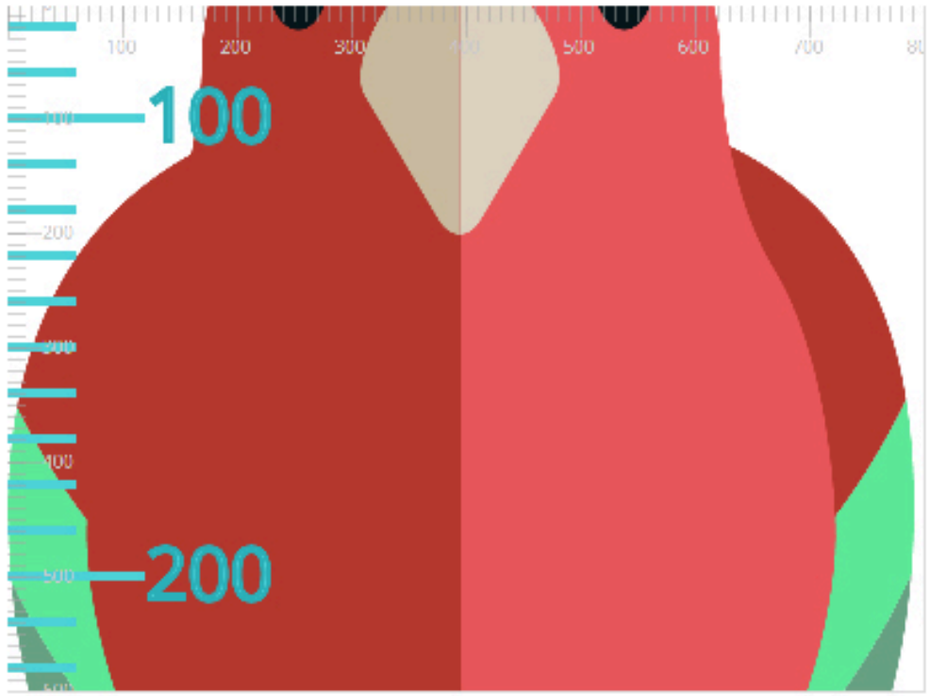
- meetOrSlice: slice funziona come background-size:cover: se le ratio non sono le stesse, la viewBox viene tagliata e la viewport resta tutta coperta in modo

preserveAspectRatio / meetOrSlice = slice

- meetOrSlice: slice funziona come background-size:cover: se le ratio non sono le stesse, la viewBox sarà tagliata; la viewBox è scalata quanto basta per coprire tutta la viewport



"xMidYmid *meet*"



"xMidYMid *slice*"

preserveAspectRatio / align

- align può avere uno fra i nove valori definiti oppure none e viene utilizzato per allineare e scalare uniformemente la viewBox nella viewport

- il default è xMidYMid

trasformazioni

- ogni elemento SVG può essere trasformato con le consuete trasformazioni (rotate, scale, skewX, skewY, translate e matrix)

- attenzione: ogni trasformazione esegue una copia del sistema di riferimento e l'origine della trasformazione è nelle coordinate 0 0 dello user space!

- è comunque possibile applicare trasformazioni via css con la consueta proprietà transform-origin

attributi di presentazione

- gli attributi fill, stroke e stroke-width cambiano rispettivamente il colore di riempimento, del bordo e la grandezza del bordo.

- possono essere modificate anche via css

l’elemento path

- <path d=“…”></path>

- l’elemento <path> è quello che permette di ottenere la maggior varietà di forme, grazie all’attributo d

l’elemento path

- <path d=“…”></path>

- l’elemento <path> è il modo più versatile per disegnare delle forme in SVG

- le direttive di disegno sono contenute tutte nell’attributo d

l’elemento path

- lettere maiuscole: coordinate assolute
- lettere minuscole: coordinate relative
- per esempio <*path d=“M10 10”*></*path*> indica di “spostare la penna” nel punto 10 10

l’elemento path, comandi

- Mx y - spostamento assoluto
- m dx - dy spostamento relativo
- L x y - segmento dal punto corrente a quello di coordinate x, y (analogamente per l dx dy)
- … lista completa su mdn https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths

l'elemento path, tools

- Potete utilizzare strumenti free come gravitDesigner, disegnare i path ed esportarli in SVG per poi includerli nelle interfacce

filtri

- È possibile definire dei filtri che prendano in input un nodo SVG e forniscano in output il nodo modificato

- è possibile ottenere ombre, bagliori, sfocature o altri effetti

clipPath

- usando l’elemento clipPath è possibile definire una qualsiasi forma attraverso cui “tagliare” un altro elemento, per poi applicarla via css (non supportato da tutti) o via attributo clip-path

gruppi

- è possibile utilizzare l'elemento <g> per raggruppare logicamente altri elementi e per poter applicare stili e trasformazioni a più elementi contemporaneamente (pensare ai gruppi in Photoshop per esempio)

accesso via script

- querySelector non funziona
- si devono usare le varianti document.getElementsByTagNameNS e document.createElementNS rispettivamente per accedere e creare nuovi nodi

Link

- https://www.sarasoueidan.com/blog/svg-coordinate-systems/

- demo interattiva https://www.sarasoueidan.com/demos/interactive-svg-coordinate-system/