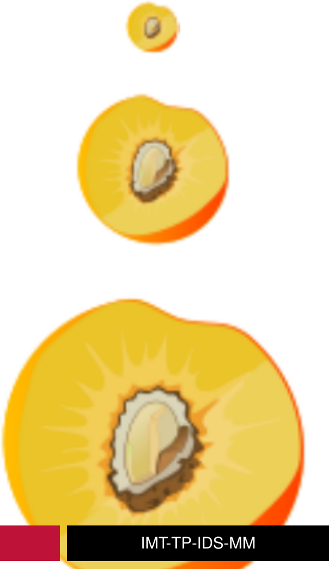
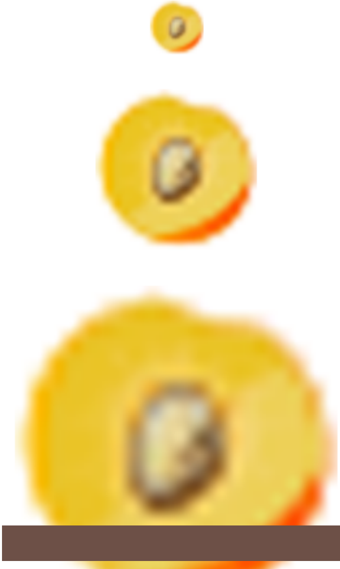
Scalable Vector Graphics

Vector Graphics

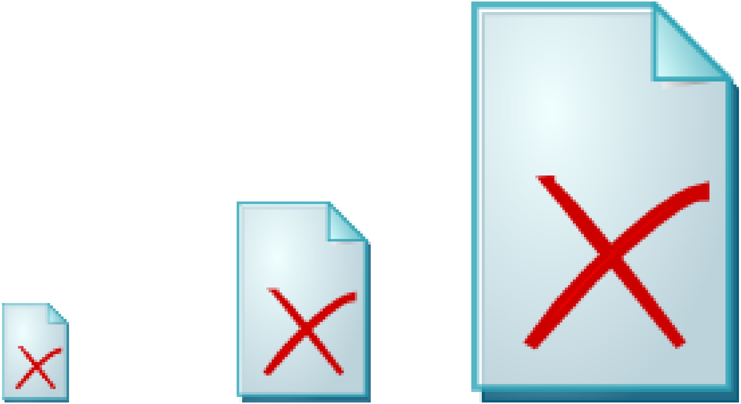
- Contrary to raster/bitmap images (pixel description)
- Graphics are described using mathematical/geometrical primitives
  - 2D objects : lines, curves, circles, rectangles, text
  - or 3D equivalent : meshes, nurbs, spheres, . . .
- Better suited for simple geometrical shapes, not for natural images, and when the scene is not complex
- Formats : SVG, VML, AI, PS, PDF, Flash. . .
- Properties :
  - Can be scaled without pixel artifacts
  - Trade-off image quality vs. rendering cost (Client-side rendering vs. server-side rendering)

Vector Graphics Properties : Zoomability





Vector Graphics Properties : Scalability



				T
PNG	25x37 / 1,55 Ko	50x75 / 3,89 Ko	100x150 / 9,89 Ko	1
SVG	-	any resolution	-	5
SVGZ	-	any resolution	-	1

SVG : a bit of history

- Initial ecosystem :
  - HTML 4.01 : 1999
  - CSS 1.0 (2nd ed.) : 1996
  - XML 1.0 (2nd ed.) : 1998
- Initial competing technologies : VML (Microsoft) and PGML (Adobe)
  - SVG 1.0 (2001)
  - SVG 1.1 (2011)
- New ecosystem : tight integration with
  - HTML 5
  - CSS 3
  - SVG 2 (2016, Candidate Recommendation)

What is SVG ?

- XML standard for
  - 2D Vector Graphics
    - Including text & fonts
    - With specific drawing, layout, positioning rules
  - With support for :
    - Styling (using CSS),
    - Animations (using JavaScript or SMIL),
    - Scripting (using JavaScript)
    - Interactivity (using JavaScript & DOM Events),
    - Raster images (PNG, JPG)
    - Multimedia (audio, video)
- Examples
  - SVG WoW
  - Snap
  - SVG animations

SVG : Benefits/Drawbacks from XML

- Benefit : SVG documents can be handled by generic XML tools
  - Syntax verification, validation
  - Modification using JS/DOM
  - Transformations using XSLT
  - …
- Drawback : Verbose
  - Some attributes are hard to read, such as `d` of `path`
  - XML requires difficult syntax (simplifications in SVG 2)

SVG Basic Example

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 500 600"
 <rect x="100" y="100" width="400" height="200"
          fill="yellow" stroke="black" stroke-width="3"/>
 <rect x="100" y="350" rx="100" ry="50" width="400" height="2
          fill="salmon" stroke="black" stroke-width="3"/>
 <circle cx="100" cy="100" r="80"
          fill="orange" stroke="navy" stroke-width="10"/>
</svg>
```

SVG Example with group

```
<svg xmlns="http://www.w3.org/2000/svg" viewBox="0 0 500 600"
 <rect x="100" y="100" width="400" height="200"
          fill="yellow" stroke="black" stroke-width="3"/>
 <g>
  <rect x="100" y="350" rx="100" ry="50" width="400" height="2
          fill="salmon" stroke="black" stroke-width="3"/>
  <circle cx="100" cy="100" r="80" fill="orange" stroke="navy
 </g>
</svg>
```

SVG Files

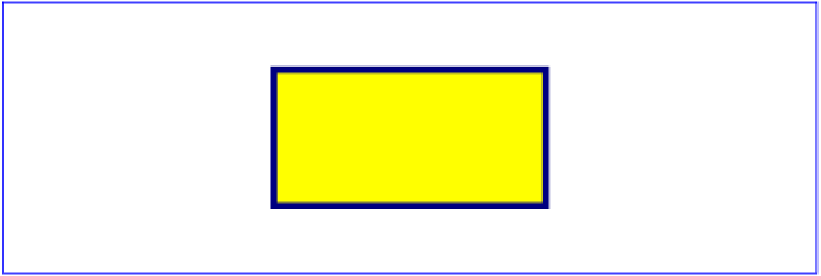
- Internet Media Type (a.k.a. MIME Type)
  - image/svg+xml
- File Extensions :
  - .svg
  - .svgz when compressed using GZIP

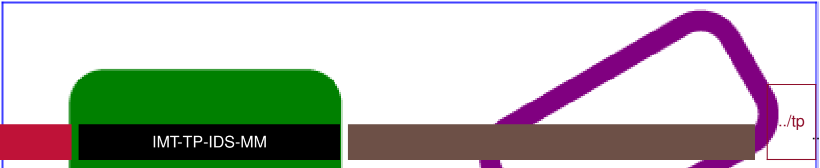
Basic shapes

Graphical Primitives

`<rect>`

- Anchored on its top left corner (x, y)
- Possible rounded corners (rx, ry)

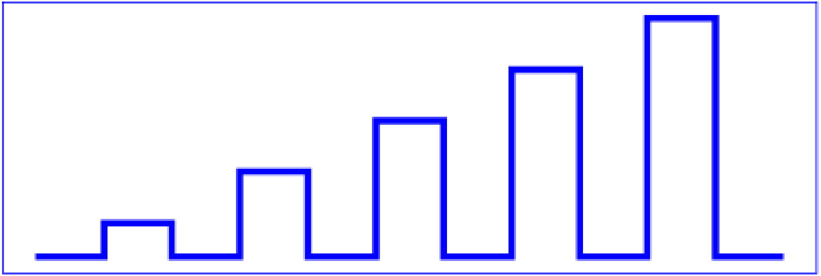




Basic shapes 2

Point/Coordinate-based primitives

- `<line>`, `<polygon>`, `<polyline>`
- `<path>` : complex curves



SVG Curves

Line segments

Bézier Curves

- Cubic (C)
- Cubic Symetrical (S)
- Quadratic (Q)
- Quadratic Symetrical (T)

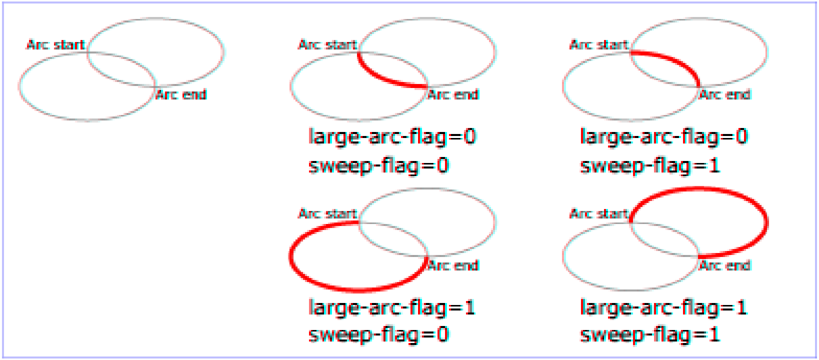
Catmull-Rom Curves (in SVG 2)

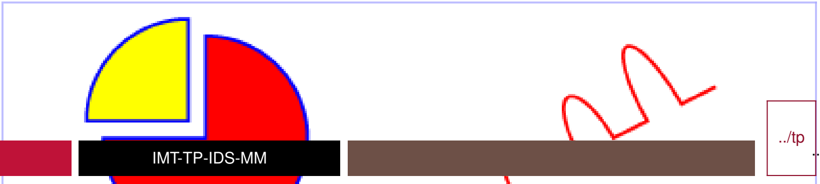
- Dotty.svg

. . .

SVG Arcs

- Start-point, end-point + arc parameter





SVG Path

Element used to describe complex graphics

`<path>`

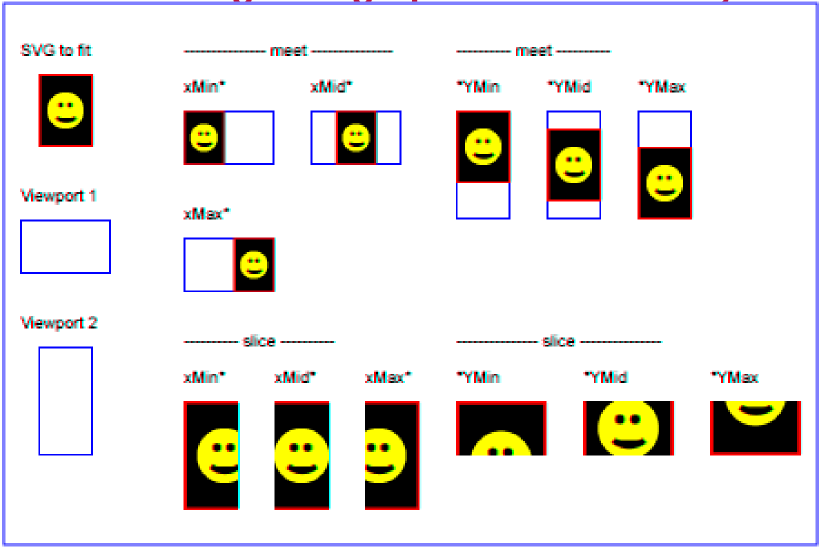
Drawing commands are described using the `d` attribute

- List of 2D points separated by drawing commands
- Use of relative or absolute user units

Text in SVG

- SVG uses specific elements for text
  - Different from HTML
    - No flowing text
    - No paragraph
  - Graphical primitives as others, can be filled, stroked, ...
  - With additional CSS text properties : `font-size`, ...
- SVG Text elements
  - `<text>` renders characters on a single line
  - `<tspan>` used to change the style of some characters on a line
  - `<textPath>` draws a text along a path (ex : legend on a river)

Viewing SVG graphics : Vocabulary



- SVG canvas

ts are

../tp

./logo-IPP-s

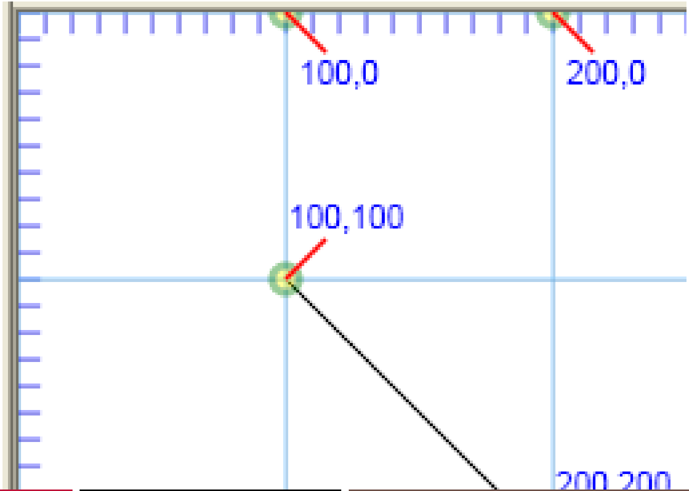
Positioning SVG : Coordinate Systems

- SVG Canvas Coordinate System
  - X-axis right-wards, Y-axis downward
  - Origin usually corresponding to the top-left corner of the viewbox
- Local Coordinate Systems
  - Origin : typically top-left or center of a shape
- Intermediate Coordinate Systems
  - Transformation of a local coordinate system
  - using <g> elements
- Units for positioning and transformations
  - Default arbitrary unit
    - Mapped to physical units based on viewBox
    - Possible to use units from CSS : cm, px, em, . . .
  - No precision limit

Example of Local Coordinate Systems

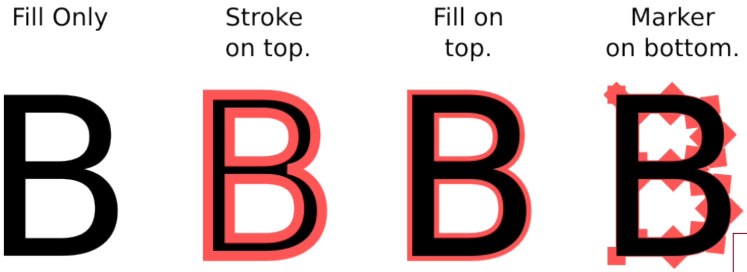
- Graphics

```
<path stroke="black" d="M 100 100 L 200 200"/>
```



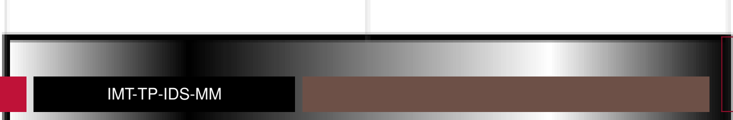
SVG Rendering Model

- Individual graphical element rendering
  - Drawing operations in order
    - Fill then stroke (or stroke then fill), using the paint-order attribute
    - Then markers
    - Then filters
    - Then clip
    - Then mask
- Then group rendering (a.k.a. compositing, blending, . . .)

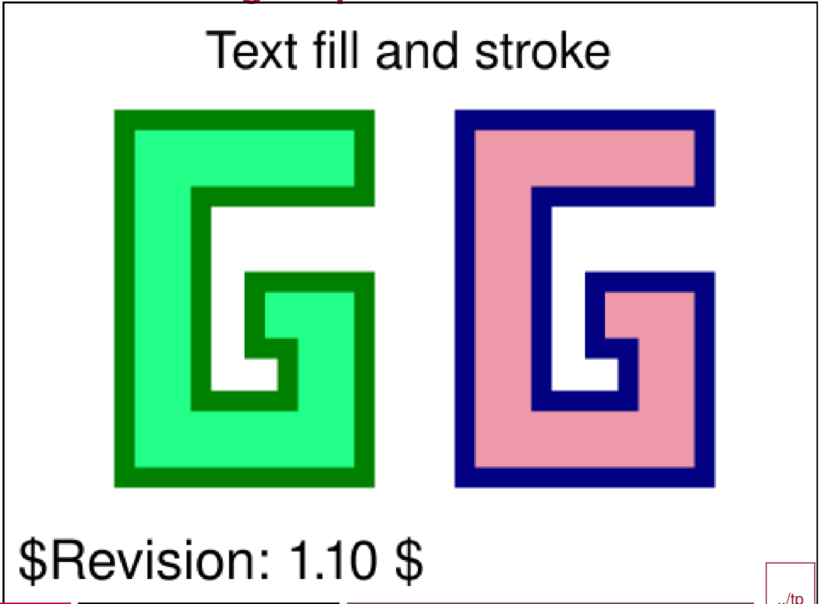


Filling Properties

- `fill`
  - A uniform/solid color
    - sRGB color space or ICC color profile : Extensions in SVG 2 Color Module
    - Syntax : `css    rgb(int[0-255], int[0-255], int[0-255]);    rgb([0-100]%, [0-100]%, [0-100]%); black, white . . .`
  - A linear or radial gradient
    - Also used in CSS
    - Extensions to Gradient Meshes in SVG 2
  - A pattern
    - Extensions to hatches in SVG 2
- `fill-opacity`
  - Transparency used for alpha-blending
- `fill-rule`
  - When a graphical primitive self-intersects



Stroking Properties



Authoring tools

- Your favorite text editor !
- Commercial tools
  - Adobe : Illustrator, EdgeCode, Edge Animate, Dreamweaver
  - Microsoft : Visio
  - CorelDraw
- Open Source Software
  - Inkscape (GUI editor, Free, Open Source)
  - SVG Edit (editor in the browser)
- SVG Cleaners
  - Inkscape : Save as « Plain » or « Optimized » SVG
  - Scripts for cleaning : scour, . . .

Summary of the lesson

- What is vector graphics, history of SVG, pros and cons
- Elements, coordinates, rendering
- Authoring tools