CSCI-UA-4-005

Intro to Web Design + Computer Principles

Vector Graphics – Day 1

Professor Emily Zhao
M/W 12:30PM – 1:45PM

Agenda

- **Midterm Format**
- **Assignment #4 Wrap Up**
- **Introduce Vector Graphics + Practice Exercises**
  - What they are
  - Coding them
  - Styling them
  - Creating links

Midterm

Midterm

Date: Monday, October 23rd

Format: Multiple Choice

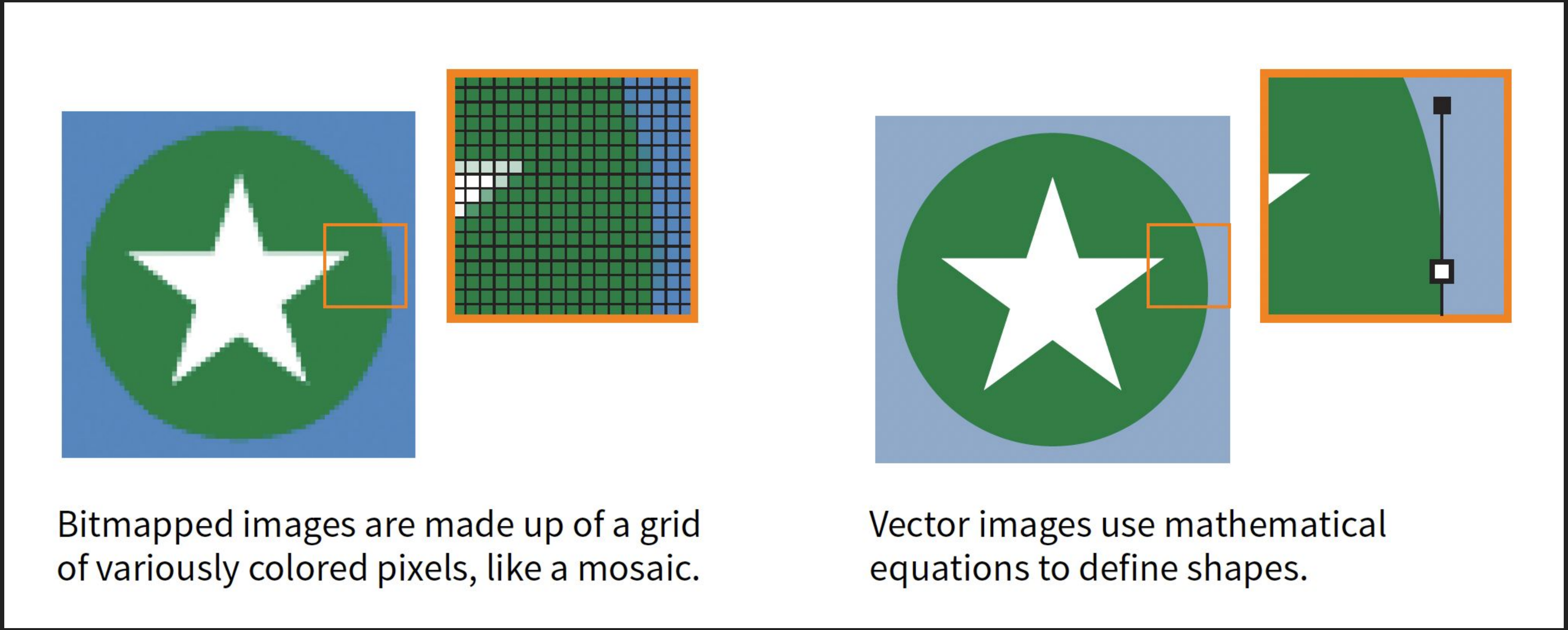
Topics Covered: Computer Principles, The Internet, Unix, HTML, CSS, Web Graphics

- Paper exam; no laptops/internet
- Open note (bring in whatever you need)
- 5-10 multiple choice questions per unit
- 25-35 multiple choice questions in reference to attached code

srcset

Vector Graphics

Raster vs Vector



Bitmapped images are made up of a grid of variously colored pixels, like a mosaic.

Vector images use mathematical equations to define shapes.

Vector Graphics

- Vector graphics contain geometric objects, such as lines and curves.
- Images can be scaled up or down without a loss of quality because the software can recalculate the shapes based on the new size.
- Since all modern displays are raster-oriented, the difference between raster-only and vector graphics comes down to where they are rasterized.
- Vector graphics are “rasterized” client side; raster graphics are, by nature, already rasterized on the server.

Scalable Vector Graphics

- Scalable Vector Graphics (SVG) is a markup language for describing two-dimensional graphics.
- SVG allows for three types of graphic objects: vector graphic shapes, images, and text.
- SVG drawings can be interactive and even styled with CSS.
- SVG defines vector graphics in XML format.

XML (eXtensible Markup Language)

- XML is a general-purpose markup language used to structure data in a way that's both human-readable and machine-readable.
- It doesn't define how data should be presented; instead, it defines the data's structure and hierarchy.
- In XML, you define your own tags and document structure; they are “extensible.” XML doesn't have predefined tags like HTML.
- SVG provides a rich, structured description of vector and mixed vector/raster graphics with pure XML.

Example XML

```xml
<person>
    <name>John Doe</name>
    <age>30</age>
    <address>
        <street>123 Main St</street>
        <city>New York</city>
    </address>
</person>
```

- Tags and structure are user-defined.
- XML doesn't define how this data should be displayed; it’s only used for structuring data.
- XML is flexible/extensible; HTML is specific to web page content + presentation.

Scalability

- To be scalable, means to increase or decrease uniformly.
- In terms of graphics, it means not being limited to a single, fixed, pixel size.
- On the web, scalability means that a particular technology can grow over time.
- SVG is scalable in both senses of the word.

Advantages of SVG

- SVG images can be created and edited with any text editor.
- SVG images can be searched, indexed, scripted, and compressed.
- SVG images are scalable, can be printed at any resolution, and are zoomable without degradation.
- SVG is an open standard!

Okay, I get it, SVGs are great – how do I make them?

How to create SVGs

1) Text Editor (Manual Coding)
2) Vector Graphics Software
    a) Adobe Illustrator
    b) Inkscape (open source)

SVG Code

```
<svg width="100" height="100">
    <rect x="10" y="10" width="80" height="80" fill="red" />
</svg>
```

SVG Drawing Elements

Rectangle

Specify attributes for top, left point of rect (x, y) and size (width and height)

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

Circle

Specify attributes for center point (cx, cy) and radius (r)

```
<circle cx="100" cy="100" r="50"/>
```

Line

Specify attributes for 2 points (x1, y1) and (x2,y2) as well as the line color (stroke)

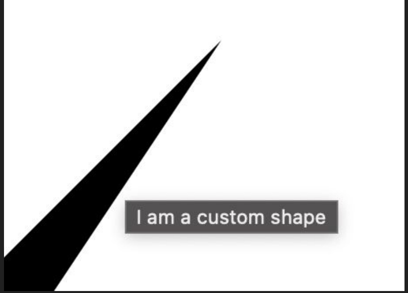
```
<line x1="0" y1="80" x2="100" y2="20" stroke="black" />
```

SVG Drawing Elements

title

Provides an accessible, short-text description of any SVG; not rendered as part of graphic but displayed rather as a tooltip

```
<title> This is a description </title>
```



group

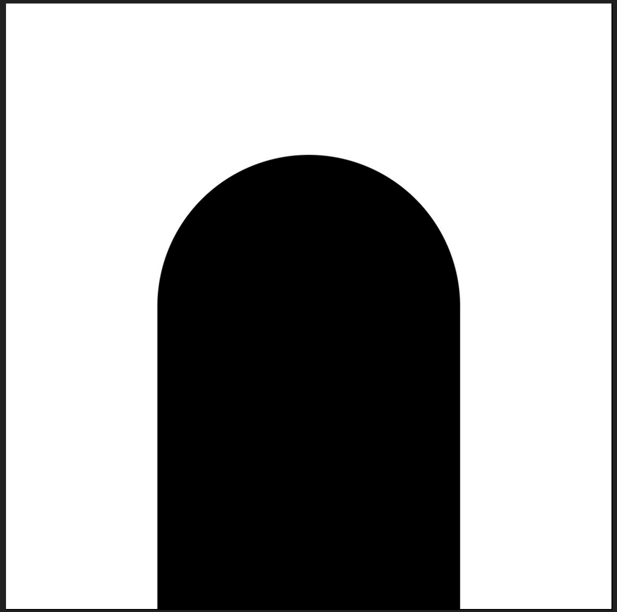
Used as a container to group SVG elements

```
<g>
    <circle cx="40" cy="40" r="25" />
    <circle cx="60" cy="60" r="25" />
</g>
```

SVG Exercise: Draw this shape!

This SVG is 200px by 200px.

- Add a title to your SVG so that on hover, a dialogue box appears with a description
- *Hint: Can you separate the larger shape into smaller shapes?*



SVG Viewbox

`viewBox` defines the logical coordinate system and aspect ratio for the SVG content, allowing for flexible and responsive scaling. Rectangular are is specified in user coordinates (`x` `y` `width` `height`)

```
<svg viewBox="0 0 100 100">
    <!-- SVG content goes here -->
</svg>
```

`width` and `height` set the physical dimensions of the SVG element on the screen or within the document but may not preserve the content's aspect ratio. It's typically used for fixed-size SVGs.

```
<svg width="200" height="100">
    <!-- SVG content goes here -->
</svg>
```

Styling SVGs

Common SVG Styling Properties

`fill`

sets the color inside the shape/object

`stroke`

sets the color of the line drawn around the shape/object

`stroke-width`

defines the width of the stroke

supply a value that is a number; don’t use px units!

`opacity`

specifies the opacity/transparency of a shape/object

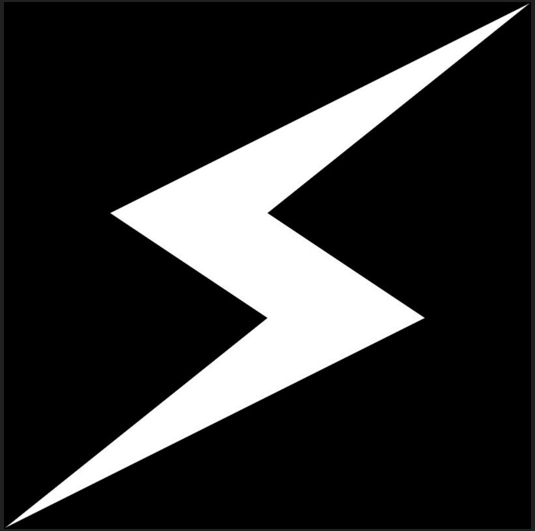
supply a value that is a floating point number from 0 to 1 (i.e. 0.5)

Using CSS Pseudo-classes

- Pseudo-classes are used to style elements that cannot be targeted using only standard element selectors.
- Pseudo-classes are denoted by a colon (":") followed by their name.
- This should look familiar to how we have styled different link <a> states
- They can be applied to SVGs as well

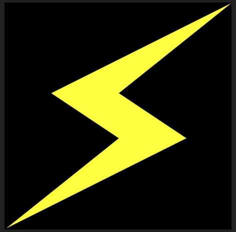
```
circle:hover {

    opacity: 0.4;

}
```

SVG Exercise: Draw this shape!

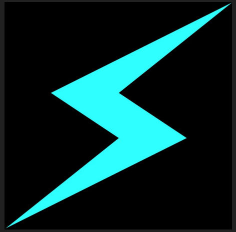


Create this shape using the `polygon` tag

- Make the background color of your webpage `black`
- Change the fill of your polygon to `white`
- On hover, the shape should turn `yellow`; on active the shape should turn `cyan`



: hover



: active

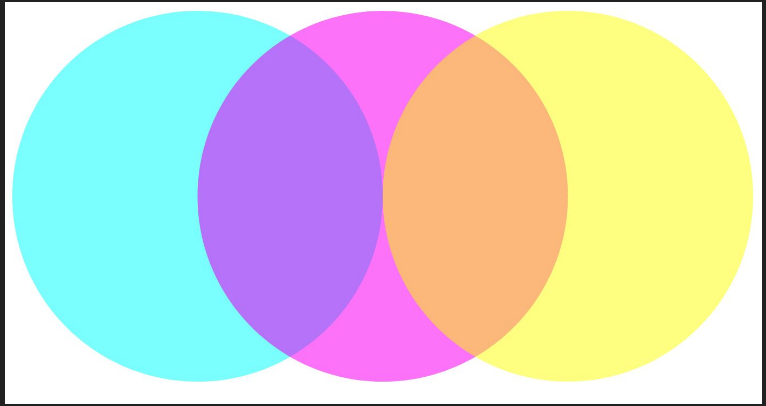
Making SVGs interactive

Making SVGs clickable

We can nest SVG drawing elements within <a> HTML elements

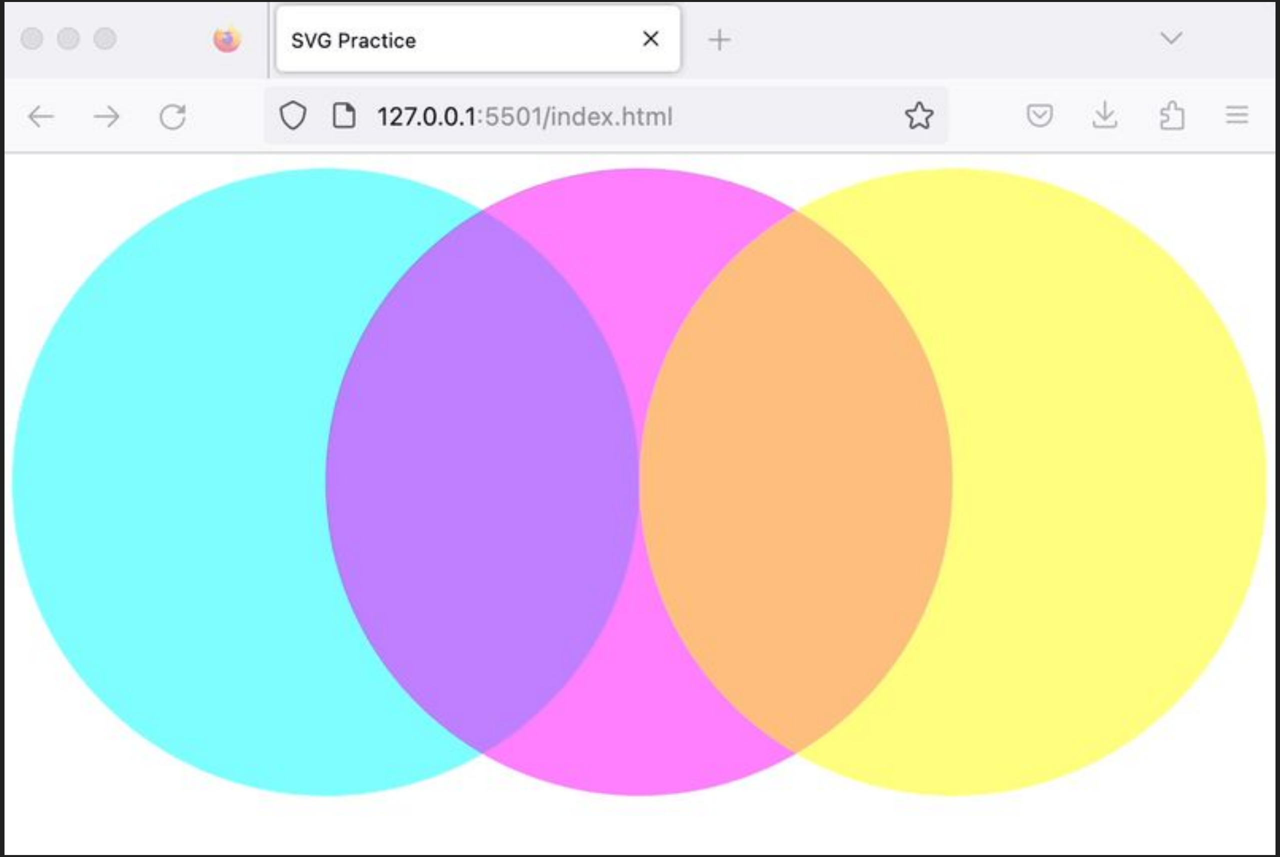
```
<svg width="100" height="100">
    <a href="www.google.com">
        <rect x="50" y="50"width="100" height="50" />
    </a>
</svg>
```

SVG Exercise: Make an SVG website



Create three circles

- The first circle should be filled with `cyan`; the second `magenta`; the third `yellow`
- All circles should have an opacity of 0.5
- On hover, the circles should become full opacity (1)
- The first circle should link to our class website; the second to exercise 1 (the door shape); the third to exercise 2 (the lightning bolt)



SVG Practice
127.0.0.1:5501/index.html