# Computer Graphics

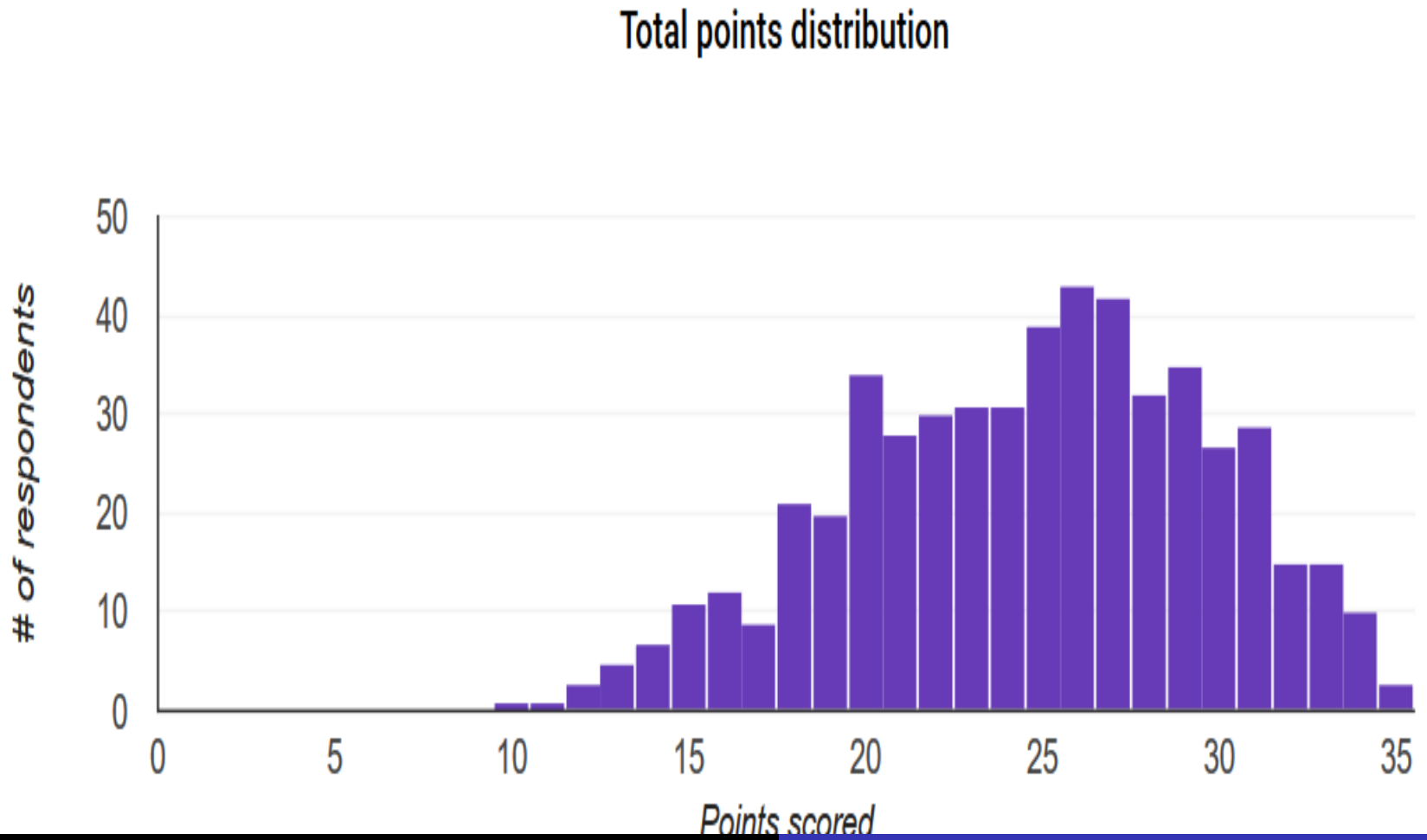
## Lecture - 05

Haitham A. El-Ghareeb

Faculty of Computers and Information Sciences
Mansoura University
Egypt
helghareeb@mans.edu.eg

March 10, 2019

| Average | Median | Range |
| --- | --- | --- |
| 24.61 / 35 points | 25 / 35 points | 10 - 35 points |



1. Introducing SVG
   - Which Image?
2. SVG Benefits
   - SVG Images are Not Images
   - Demo
3. Comparing Formats
4. Basic Example
5. Implementing SVG Shapes

# Which Image?

- which image format should I use: bitmap or vector, PNGs, JPEGs...?
- There isn't a single image format that is ideally suited to the web, which encompasses the best qualities of all image types
- Google has tried to push this with the WebP format

# JPEG or PNG

- JPEG or PNG images would be favored
- These are perfect for complex images where detail is essential (such as photographs)
- if you need to display clear line drawings, or 2D images, for example, then these formats aren't ideal
- There are several reasons why, but one of the key weaknesses is maintaining quality
- Try resizing a schematic saved as a PNG, and the quality soon becomes very poor!

# Not a Replacement

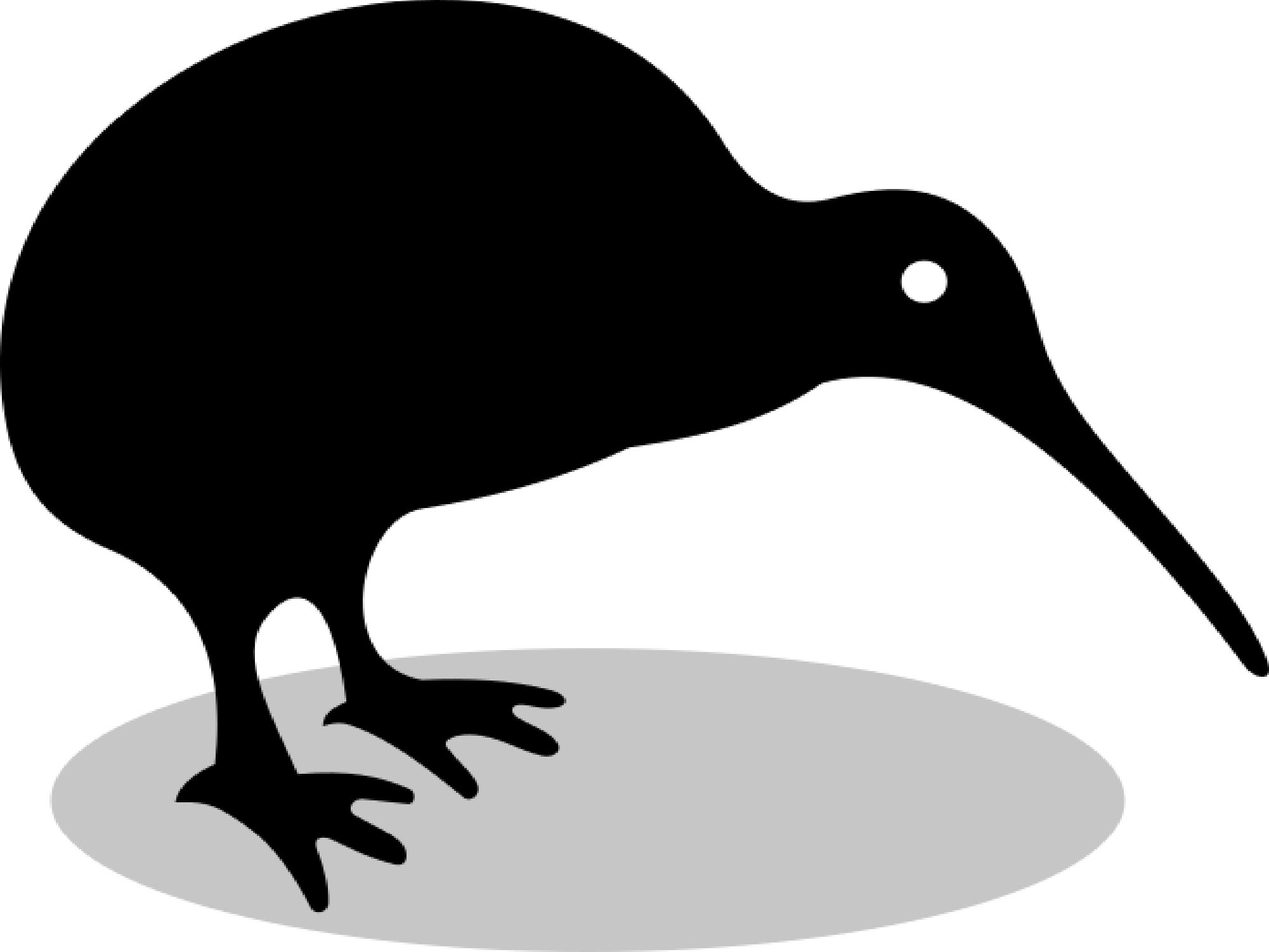
- it is not intended as a replacement for JPEG or PNG images
- works in different ways
- Shines when used to display vector images on the web

# SVG Benefits

- SVG-based images do not lose quality when they are resized or zoomed in the browser
- SVG images can be animated using JavaScript or CSS
- SVG images integrate with the DOM very well, and they can be manipulated using JavaScript or CSS
- SVG images can be indexed by search engines, which is perfect for SEO purposes
- SVG images can be printed at any resolution

# SVG images are not images

SVG images are not images



```
<?xml version="1.0" encoding="utf-8"?>
<!-- Generator: Adobe Illustrator 16.0.4, SVG Export Plug-In . SVG Version: 6.00 Build 0)  -->
<!DOCTYPE svg PUBLIC "-//W3C//DTD SVG 1.1//EN" "http://www.w3.org/Graphics/SVG/1.1/DTD/svg11.dtd">
<svg version="1.1" id="Layer_1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" x="0px" y="0px"
     width="612px" height="502.174px" viewBox="0 65.326 612 502.174" enable-background="new 0 65.326 612 502.174"
     xml:space="preserve">
<ellipse fill="#C6C6C6" cx="283.5" cy="487.5" rx="259" ry="80"/>
<path id="bird" d="M210.333,65.331C104.367,66.105-12.349,150.637,1.056,276.449c4.303,40.393,18.533,63.704,52.171,79.03
    c36.307,16.544,57.022,54.556,50.406,112.954c-9.935,4.88-17.405,11.031-19.132,20.015c7.531-0.17,14.943-0.312,22.59,4.341
    c20.333,12.375,31.296,27.363,42.979,51.72c1.714,3.572,8.192,2.849,8.312-3.078c0.17-8.467-1.856-17.454-5.226-26.933
    c-2.955-8.313,3.059-7.985,6.917-6.106c6.399,3.115,16.334,9.43,30.39,13.098c5.392,1.407,5.995-3.877,5.224-6.991
    c-1.864-7.522-11.009-10.862-24.519-19.229c-4.82-2.984-0.927-9.736,5.168-8.351120.234,2.415c3.359,0.763,4.555-6.114,0.882-7.875
    c-14.198-6.804-28.897-10.098-53.864-7.799c-11.617-29.265-29.811-61.617-15.674-81.681c12.639-17.938,31.216-20.74,39.147,43.489
    c-5.002,3.107-11.215,5.031-11.332,13.024c7.201-2.845,11.207-1.399,14.791,0c17.912,6.998,35.462,21.826,52.982,37.309
    c3.739,3.303,8.413-1.718,6.991-6.034c-2.138-6.494-8.053-10.659-14.791-20.016c-3.239-4.495,5.03-7.045,10.886-6.876
    c13.849,0.396,22.886,8.268,35.177,11.218c4.483,1.076,9.741-1.964,6.917-6.917c-3.472-6.085-13.015-9.124-19.18-13.413
    c-4.357-3.029-3.025-7.132,2.697-6.602c3.905,0.361,8.478,2.271,13.908,1.767c9.946-0.925,7.717-7.169-0.883-9.566
    c-19.036-5.304-39.891-6.311-61.665-5.225c-43.837-8.358-31.554-84.887,0-90.363c29.571-5.132,62.966-13.339,99.928-32.156
    c32.668-5.429,64.835-12.446,92.939-33.85c48.106-14.469,111.903,16.113,204.241,149.695c3.926,5.681,15.819,9.94,9.524-6.351
    c-15.893-41.125-68.176-93.328-92.13-132.085c-24.581-39.774-14.34-61.243-39.957-91.247
    c-21.326-24.978-47.502-25.803-77.339-17.365c-23.461,6.634-39.234-7.117-52.98-31.273C318.42,87.525,265.838,64.927,210.333,65.331
    z M445.731,203.01c6.12,0,11.112,4.919,11.112,11.038c0,6.119-4.994,11.111-11.112,11.111s-11.038-4.994-11.038-11.111
    C434.693,207.929,439.613,203.01,445.731,203.01z"/>
</svg>
```

# Scary?

- Numbers are just coordinates that trace the outline of the image
- We won't be expected to write code like that
- ; instead, we would add an SVG image using the standard image tag:

```
<img class="mama" src="img/kiwi.svg">
```

# Change Color

```
<ellipse fill="C6C6C6" cx="283.5" cy="487.5" rx=
```

Change to 834*DCF*

# Comparing Formats - 01

- JPEG or PNG – the best choice for size and support in today's modern browsers, when working with photo-based imagery
- However, websites must be accessible on different devices, these formats do not scale well if we need to use line-based drawings

# Resolution Independence

- with many image formats, we might have to download extra data or assets to fix resolution-based issues.
- A great example is when using retina screens, which require us to apply a @2x hack to force higher-resolution images to be displayed.
- This isn't the case with SVG images; these can be fully resized, irrespective of device or resolution used, and without the need for additional tags.

# Accessible DOM API

- SVG elements can be manipulated using nothing more than standard JavaScript or CSS
- this could be as simple as changing colors or as complicated as attaching event handlers to perform specific events.

# No Unnecessary HTTP Requests

- Unlike standard images, SVG images are created using XML and CSS
- This avoids the need for the browser to request an image from the server, making it faster and more user friendly.

# Content

- Content can be indexed, scaled, searched, scripted, and compressed.

# Text Editor

- We can create images using nothing more than a text editor – yes, it might be easier to create them in a suitable graphics application
- However, let us not forget that the key principle outlined earlier: Why download and install a graphics package if we can achieve the same result in a text editor that we already have?

# SVG Used For

- Logos and icon with strong, geometric, vector-friendly designs
- Graphics that need to be displayed in multiple sizes and screens
- Graphics that respond to their device
- Graphics that need to be edited, updated, and redeployed

# Lossy Images

- Lossy images do not retain all of the data in an image, particularly when converted to JPEG;
- other formats retain data (i.e., are lossless), but do not offer capabilities such as built-in animation or clear scalability.

| | Category | Palette | Used for |
|---|---|---|---|
| JPG / JPEG | Lossy | Millions of colors | Still Images, Photography |
| GIF | Lossless | Maximum 256 colors | Simple animationsGraphics with flat colorsGraphics without gradients |
| PNG-8 | Lossless | Maximum 256 colors | Similar to GIFBetter transparency but no animationGreat for icons |
| PNG-24 | Lossless | Unlimited colors | Similar to PNG-8Handles still images and transparency |
| SVG | Vector/lossless | Unlimited colors | Graphics/logos for webRetina/high-dpi screens |
| WebP | Lossless | Unlimited colors | Similar to PNGs, but typically 26% smaller in size – take-up isn't so extensive, with only Chrome and Opera supporting the format at present |

Activate Windows
Go to Settings to activate Windows.

# Can I use _____ ? Settings

16 results found

## SVG (basic support) - REC

Usage: Global — % of all users ?

94.51% + 0.33% = 94.84%

Method of displaying basic Vector Graphics features using the embed or object elements. Refers to the SVG 1.1 spec.

Current aligned | Usage relative | Date relative | Apply filters | Show all ?

| IE | Edge * | Firefox | Chrome | Safari | iOS Safari * | Opera Mini * | Chrome for Android | UC Browser for Android | Samsung Internet |
|---|---|---|---|---|---|---|---|---|---|
| | [2] 17 | 64 | 71 | | 11.4 | | | | 4 |
| [2] 11 | [2] 18 | 65 | 72 | 12 | 12.1 | all | 71 | 11.8 | 8.2 |
| | | 66 | 73 | 12.1 | 12.2 | | | | |
| | | 67 | 74 | TP | | | | | |
| | | | 75 | | | | | | |

Notes | Known issues (4) | Resources (7) | Feedback

Activate Windows
Go to Settings to activate Windows

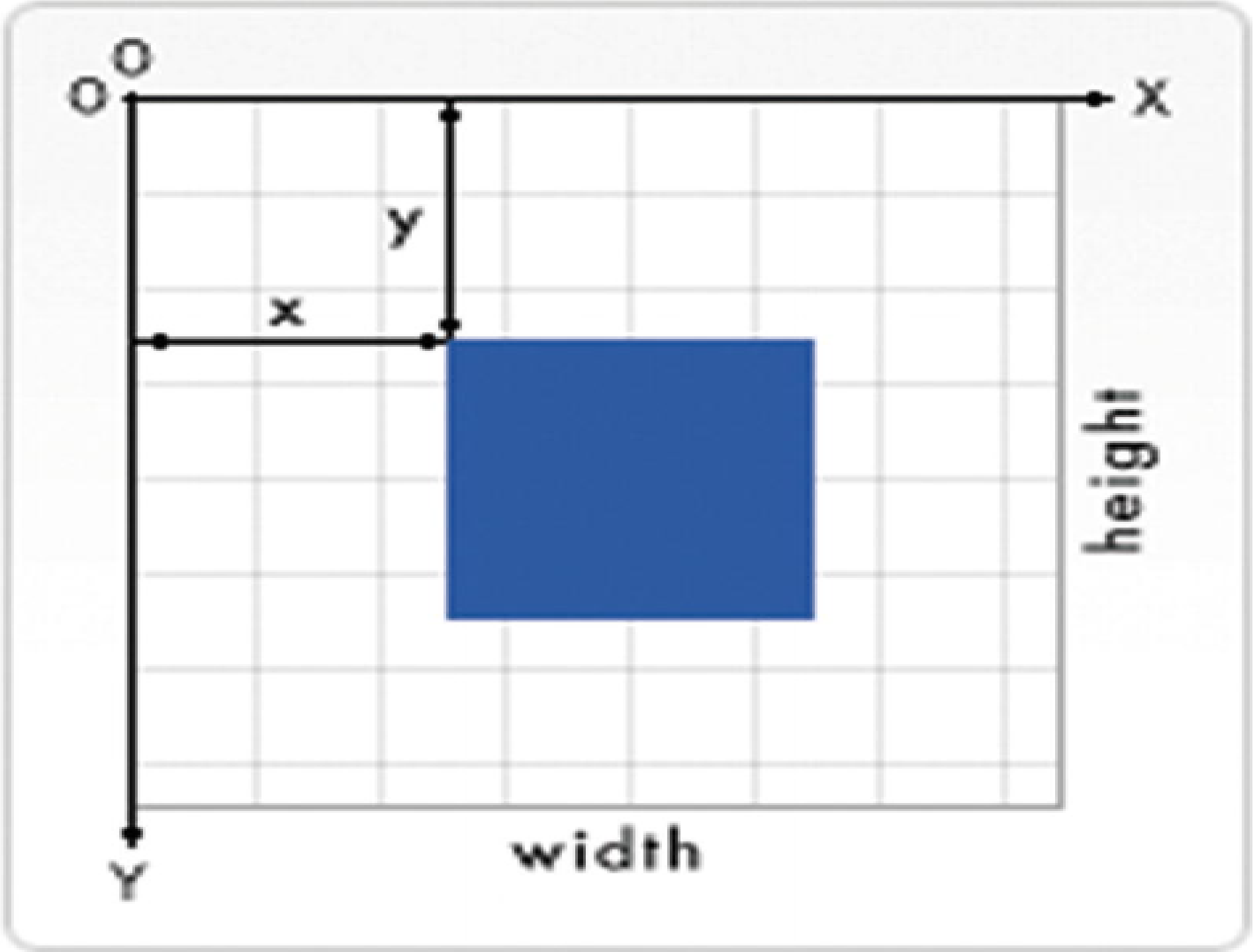
[2] IE9-11 desktop & mobile don't properly scale SVG files. Adding height, width, viewBox, and CSS rules seem to be the best workaround.

# Create Basic Example

001

# How it Works

- **SVG viewport** – this controls the size of the window, through which we may view a particular SVG element
- We can restrict this by specifying a viewbox attribute – whereas the viewport can be enormous, the viewbox limits the extent of what we see.


O
O
X
y
x
height
width
Y

# Implementing SVG Shapes

- 002 - Squares and Rectangles

## *Table 2-1* Attribute Properties for Rectangles

| Attribute | Purpose |
|---|---|
| `x` | The x position of the top left corner of the rectangle. |
| `y` | The y position of the top left corner of the rectangle. |
| `width` | The width of the rectangle. |
| `height` | The height of the rectangle. |
| `rx` | The x radius of the corners of the rectangle. |
| `ry` | The y radius of the corners of the rectangle. |

Activate W
Go to Settings

# Implementing SVG Shapes

- 003 - Circles and Ellipses

***Table 2-2*** Attribute Properties for Circles and Ellipses

| Attribute | Purpose |
|---|---|
| `r` | The radius of the circle. |
| `cx` | The x position of the center of the circle. |
| `cy` | The y position of the center of the circle. |
| `rx` | The x radius of the ellipse. |
| `ry` | The y radius of the ellipse. |

# Implementing SVG Shapes

- 004 - Lines, Poly-lines, and Polygons

***Table 2-3*** Attributes for Lines, Polylines, and Polygons

| Attribute | Purpose |
|---|---|
| `x1, y1` | The x and y positions of point 1 (our starting point). |
| `x2, y2` | The x and y positions of point 2 (our finishing point). |
| `points` | A list of points, each separated by a comma, space, EOL, or line feed character. Each must contain an x and y coordinate – the drawing automatically closes the path, so a final line will be drawn from the last set of points to the starting set.<br><br>**Note**: this applies to lines and polygons only; to see how it works for polylines, add a fill color to the shape. |

Activate Windows
Go to Settings to activate Windows.

# Paths and Markers

- 005 - Gradient
- Path – series of coordinates that when joined, form a design
- we can apply all manner of different styles to it
- `https://codepen.io/chriscoyier/pen/NRwANp`

# Paths and Markers - Cont.

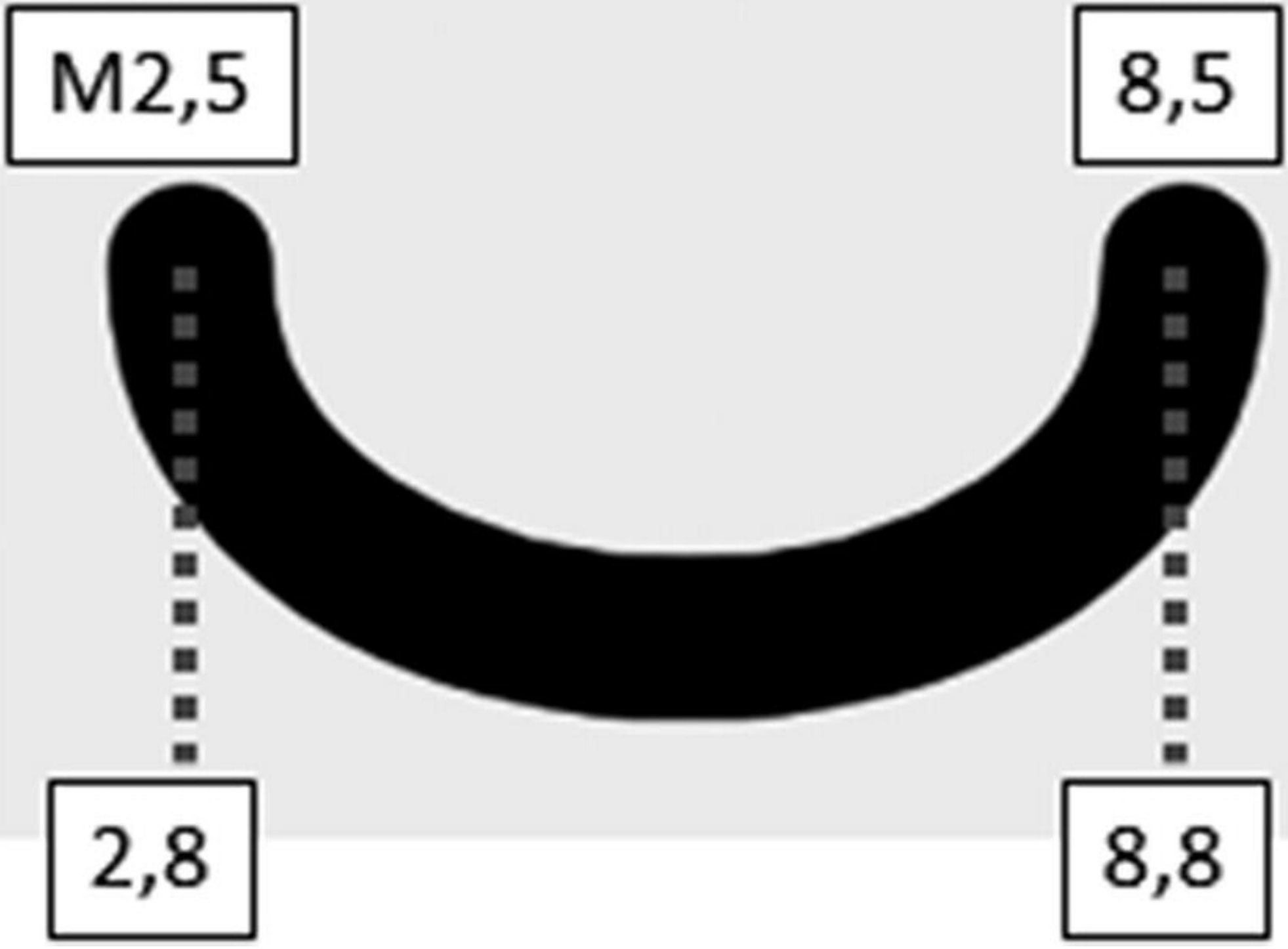
```
<svg viewBox="0 0 10 10" class="svg-4">
<path d="M2,5 C2,8 8,8 8,5" />
</svg>
```

# Understanding Code in Details

- The first line of code is the standard opening tag for any SVG shape or design
- The real magic happens in line 2 – we have the ¡path¿ tag, inside which we assign a series of numbers and or letters to create our design.
- There is, however, method in the apparent madness of that string of characters – they are a series of commands to define how our shape should appear.
- To make sense of it, there is one important concept we should be aware of: the difference between absolute and relative commands

# Understanding Code in Details - Cont.

- Let's take the first command: M2,5. It means "move to the exact location 2, 5".
- The next command, C2,8 8,8 8,5, is a little more complex:
- we use this to create a **Bezier curve**.
- The starting point for it was defined with the initial command;
- the next three coordinates define the degree of curve and end point of our Bezier curve.


M2,5
8,5
2,8
8,8

# Vector Conflict - Solved :)

- There are differences between absolute and relative commands
- Most commands come in pairs – either as uppercase characters, or as lowercase equivalents
- The uppercase characters represent absolute commands, whereas the lower case ones are relative
- To put this into context, our example directed the starting point to be at the absolute location of 2,5.
- If we had made it relative (i.e., used a lowercase m instead), then it would read as "move 2 to the right, and 5 down," from our current location instead.

# More Info. on Paths

https://css-tricks.com/svg-path-syntax-illustrated-guide/

# Markers

- 006-Path
- Let's understand the Code