DLI Accelerated Data Science Teaching Kit

Lecture 9.2 - Prerequisites: Javascript & SVG

Chrome Inspector and Console

- Open the webpage
- Right-click on anything
- Click “inspect”
- Open the console too, so you can see the error messages

Starting a Local Web Server

Necessary for Chrome, not for Safari or Firefox
(This is a security measure: to prevent reading from your file systems)

- Python 2.x
  - python -m SimpleHTTPServer 8000
- Python 3.x
  - python –m http.server 8000
- http://localhost:8000

If you’re new to JavaScript…

prepare for a lot of…
confusion (wat??)
and hair **pulling**

I’m serious.

If you’re new to JavaScript…

Screenshot from video

https://www.destroyallsoftware.com/talks/wat

(start video at 1:20)

JavaScript 101

- All variables are global, unless declared using var
  - x = 300 (global)
  - var x = 300 (local)
- Semicolons are optional
- **“text”** is the same as **‘text’**
- JS arrays and objects are almost exactly the same syntax as python’s lists [ ] and dicts { }
- object.key is the same as object[‘key’]
- Print to the console using console.log( )

JavaScript 102: Functional Programming

- Javascript supports **functional programming**
  - Functions are themselves objects
  - Functions can be stored as variables
  - Functions can be passed as parameters
- D3 uses these abilities extensively!

Some people say javascript is a “multi-paradigm” programming language.
http://stackoverflow.com/questions/3962604/is-javascript-a-functional-programming-language

What does that mean?

Examples

Mapping an array of numbers to an array of square roots

The following code takes an array of numbers and creates a new array containing the square roots of the numbers in the first array.

Passing Math.sqrt (a function) as a parameter

```
1 | var numbers = [1, 4, 9];
2 | var roots = numbers.map(Math.sqrt);
3 | // roots is now [1, 2, 3], numbers is still [1, 4, 9]
```

https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/Array/map

MDN – the BEST Javascript reference

- Mozilla Developer Network
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference
- (Easier: google “<command> mdn”)

Method Chaining

"Syntactic Sugar" paradigm where each method returns the object that it was called on

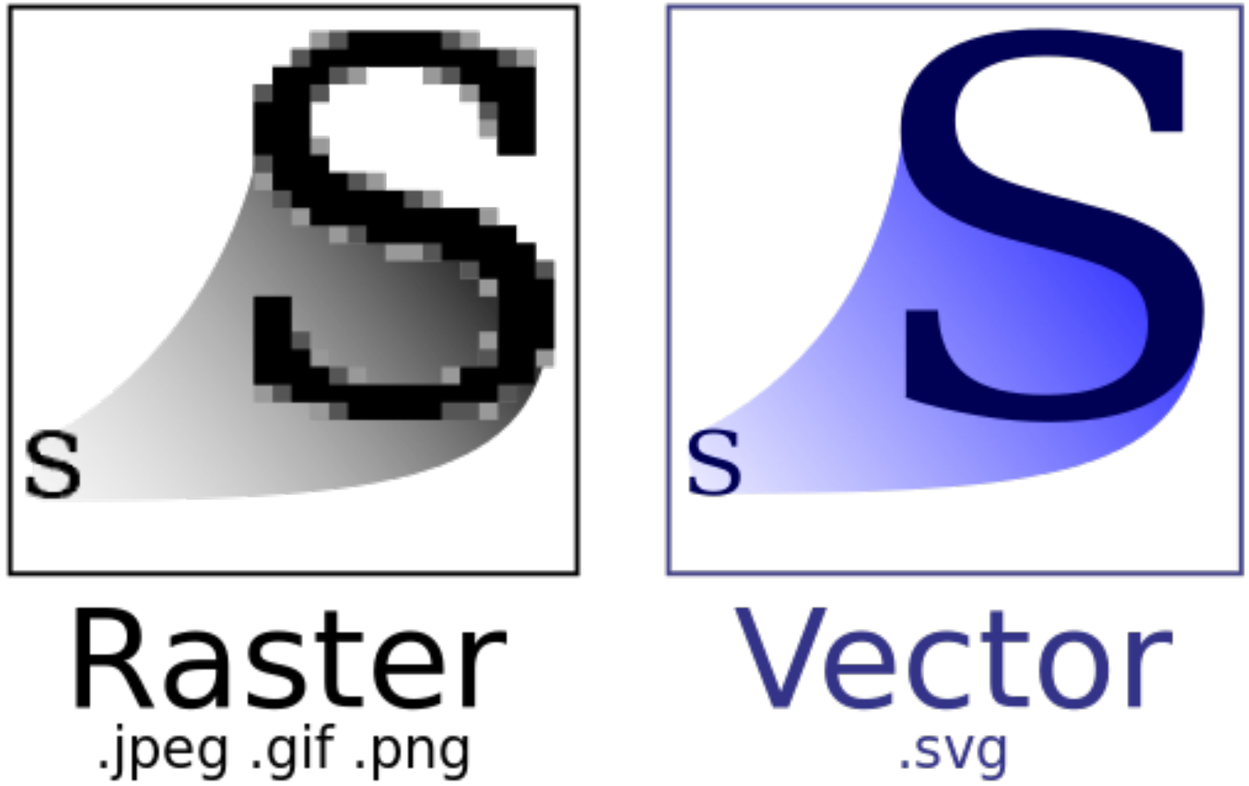
```
group
      .attr("x",5)
      .attr("y",5); //returns group
```

is the same as

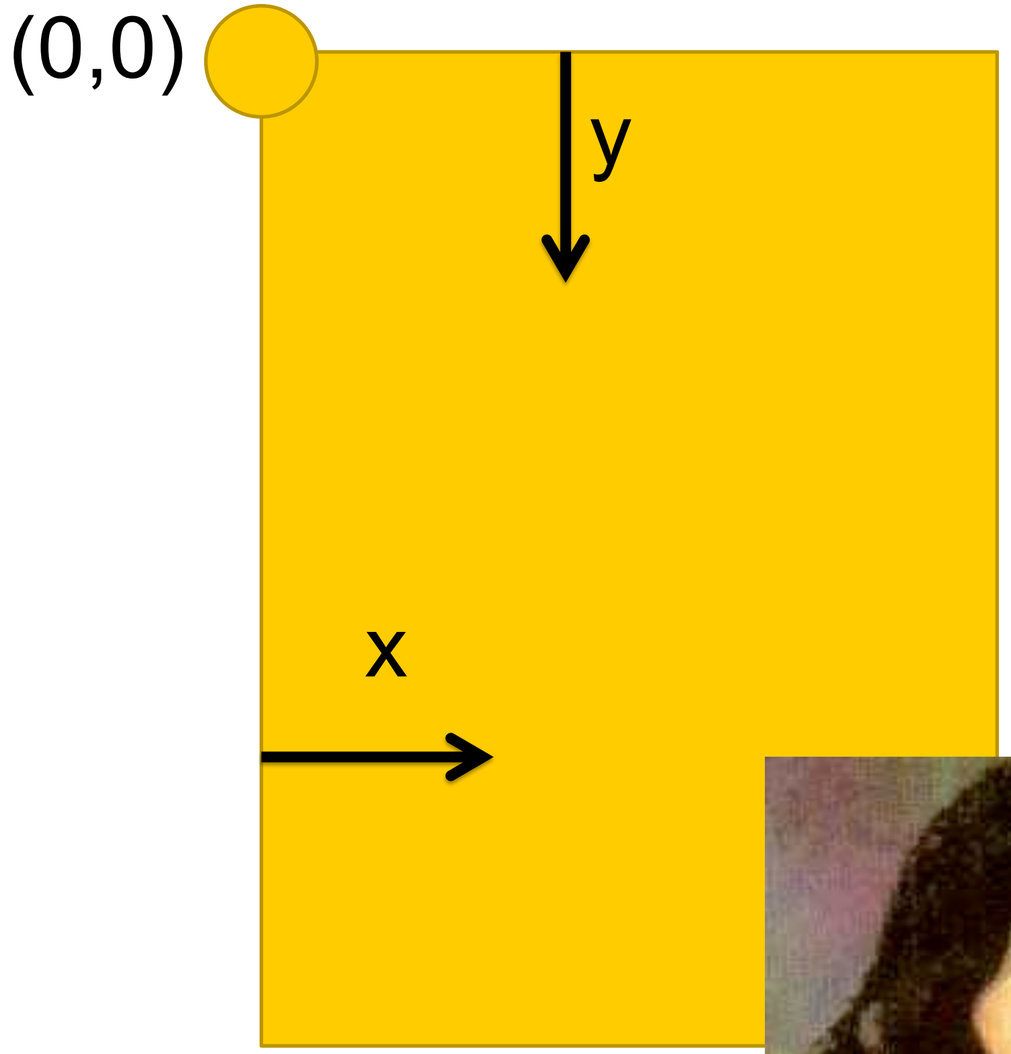
```
group.attr("x",5) //returns group
group.attr("y",5) //returns group
```

SVG Basics

SVG = Scalable Vector Graphics



Retrieved on December 1, 2017 https://en.wikipedia.org/wiki/Scalable_Vector_Graphics



http://smg.photobucket.com/user/Pavan2099/media/RvB/Descart-weeping.png.html

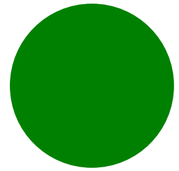
SVG Basics

SVG -> XML Vector Graphics

(Scalable Vector Graphics)

SVG Basics

- XML Vector Graphics
  - Tags with Attributes
  - `<circle r=5 fill="green"></circle>`



- W3C Standard
  - http://www.w3.org/TR/SVG/
- Supported by all the major browsers

SVG Basics

- <svg>
- <circle>
- <rect>
  <path>
- <g>
- <text>

<svg> element

- Overarching canvas

- (optional) Attributes:
  - width
  - height

```
<body>
  <div id="vis">
    <svg></svg>
  </div>
</body>
```

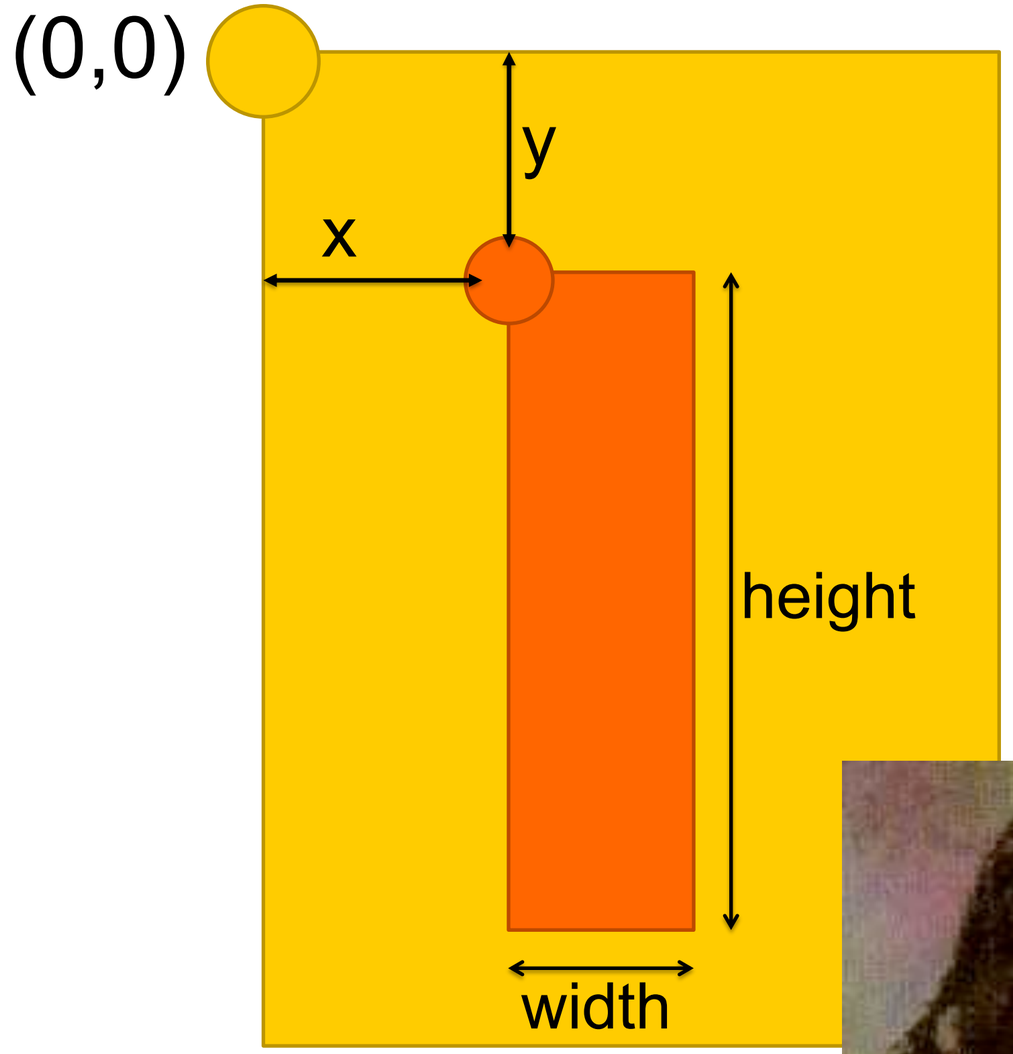
- Create with
  - `d3.select("#vis").append("svg")`

<circle> element

- Attributes:
  - cx (relative to the LEFT of the container)
  - cy (relative to the TOP of the container)
  - r (radius)
- (optional) Attributes:
  - fill (color)
  - stroke (the color of the stroke)
  - stroke-width (the width of the stroke)
- Create with
  - `.append("circle")`

<rect> element

- Attributes:
  - x (relative to the LEFT of the container)
  - y (relative to the TOP of the container)
  - width (cannot be negative)
  - height (cannot be negative)
- (optional) Attributes:
  - fill (color)
  - stroke (the color of the stroke)
  - stroke-width (the width of the stroke)
- Create with
  - `.append("rect")`



http://smg.photobucket.com/user/Pavan2099/media/RvB/Descart-weeping.png.html

Rather than positioning each element, what if we want to position (or style) a **group** of elements?

<g> element

- Generic container (Group) element

- Attributes
  - transform
  - (fill,stroke,etc.)
- Create with:
  - `var group = vis.append("g")`
- Add things to the group with:
  - `group.append("circle")`
  - `group.append("rect")`
  - `group.append("text")`

CSS Selectors Reference

- By ID: `#vis` → `<tag id="vis">`
- By tag name: `circle` → `<circle>`
- By class name: `.canary` → `<tag class="canary">`
- By attribute: `[color="blue"]` → `<tag color="blue">`
- And many more ways
  - http://www.w3schools.com/cssref/css_selectors.asp
- And any combinations…
  - AND
    circle.canary → <circle class="canary">
  - OR
    circle, .canary → <circle> <circle class="canary"> <tag class="canary">

DLI Accelerated Data Science Teaching Kit

Thank You

We thank Dr. Chad Stolper for sharing teaching materials for visualization and D3.