EDAV2 Notes

Framework of an .html file

```
<!DOCTYPE html>
<html lang=”en”>
    <head>
        <meta charset=”utf-8”>
        <title>EDAV1</title>

        <script src=”https://d3js.org/d3.v4.min.js”></script>

        <style type=”text/css”>
            h1 {color:red;}
            p {color:blue;}
        </style>

    </head>

    <body>
        <h1>h1 header</h1>
        <h2>h2 header</h2>
        <h3 style=”font-family: Palatino;”>h3 header</h3>
        <p>paragraph</p>

        <svg width=”500” height=”300”>
            <rect x=”20” y=”20” width=”460” height=”260” fill=”aliceblue”></rect>
            <circle cx=”50” cy=”75” r=”20” fill=”blue”></circle>
            <ellipse cx=”175” cy=”100” rx=”45” ry=”30” fill=”green”></ellipse>
            <text x=”150” y=”200”>(150, 200)</text>
            <line x1=”250” y1=”150” x2=”300” y2=”200” stroke=”red” stroke-width=”3”></line>
        </svg>

        <script>
            // JavaScript / D3 will go here
        </script>

    </body>

</html>
```

Sections

HTML

HEAD

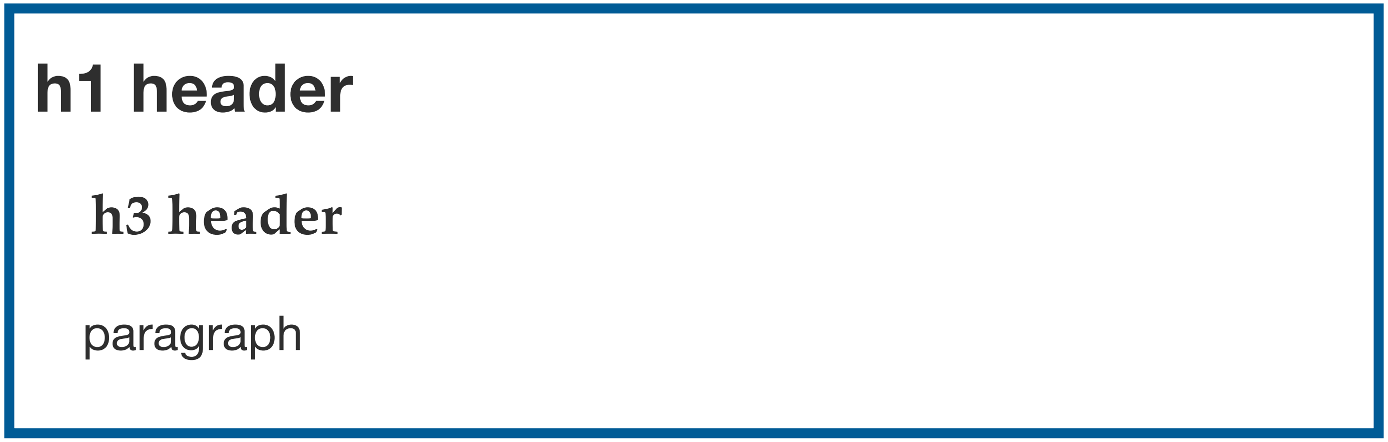
1. Title
2. Link to D3
3. CSS (styles)

BODY

1. HTML (text)
2. SVG (graphics)
3. D3 / JavaScript (dynamic content)

HTML

```html
<h1>h1 header</h1>
<h3 style="font-family: Palatino;">h3 header</h3>
<p>paragraph</p>
```



h1 header

h3 header

paragraph

http://www.dolekemp96.org/news/releases/releases.html#Press_Releases

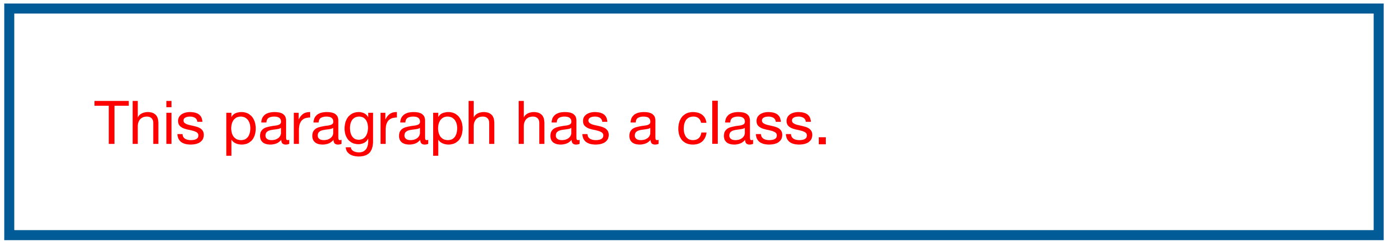
https://www.nytimes.com/

Interactive Data Visualization for the Web (IDVW) pp. 19-25

CSS

```
    .myclass2 {
      color:red;
      font-size: 30px;
     }
```

```
<p class="myclass2">This paragraph has a class.</p>
```



This paragraph has a class.

CSS rules enable styling **and** selecting.

IDVW pp. 30-35

SVG

```
<svg width="500" height="300">  <!-- some SVG -->
    <rect x="20" y="20" width="460" height="260" fill="aliceblue"></rect>
    <circle cx="50" cy="75" r="20" fill="blue"></circle>
    <ellipse cx="175" cy="100" rx="45" ry="30" fill="green"></ellipse>
    <text x="150" y="200">(150, 200)</text>
    <line x1="250" y1="150" x2="300" y2="200" stroke="red" stroke-width="3"></line>
</svg>
```

```
<rect x="20" y="20" width="460" height="260" fill="aliceblue"></rect>
<circle cx="50" cy="75" r="20" fill="blue"></circle>
<ellipse cx="175" cy="100" rx="45" ry="30" fill="green"></ellipse>
<text x="150" y="200">(150, 200)</text>
<line x1="250" y1="150" x2="300" y2="200" stroke="red" stroke-width="3"></line>
```

IDVW pp. 52-61

JavaScript

IDVW pp. 52-61

Be generally familiar with arrays, objects, functions

Be aware that ES6 does things differently (not covered in **IDVW**)

Arrow functions `d => d.value`

Template Literals

```
let a = 3;
let b = 4;
console.log(`This is an equation: ${a} + ${b} = ${a + b}`);
```

instead of:

```
console.log("This is an equation: " + a + " + " + b + " = " + (a + b));
```

```
This is an equation: 3 + 4 = 7
```

Breaking it down

```
d3.select("circle").transition().duration(2000)
  .attr("r", "50");
```

1. Chaining

2. Selections

3. Modifying elements

Chaining methods

```
<p id="id2" style="font-size: 72px;">
   Watch me turn red and shrink.</p>
```

Watch me turn red and shrink.

D3:

```
d3.select("#id2").transition().duration(3000)
  .style("font-size", "24px").style("color", "red");
```

Chaining methods: two transitions

```html
<p id="id3" style="font-size: 72px;">Watch me turn red,
   <em>then</em> shrink.</p>
```

Watch me turn red, ***then*** shrink.

D3:

```javascript
d3.select("#id3")
  .transition().duration(3000).style("color", "red")
  .transition().duration(3000).style("font-size", "24px");
```

Selecting by tag

Select the first one:

```
d3.select(”circle”);
```

Select all:

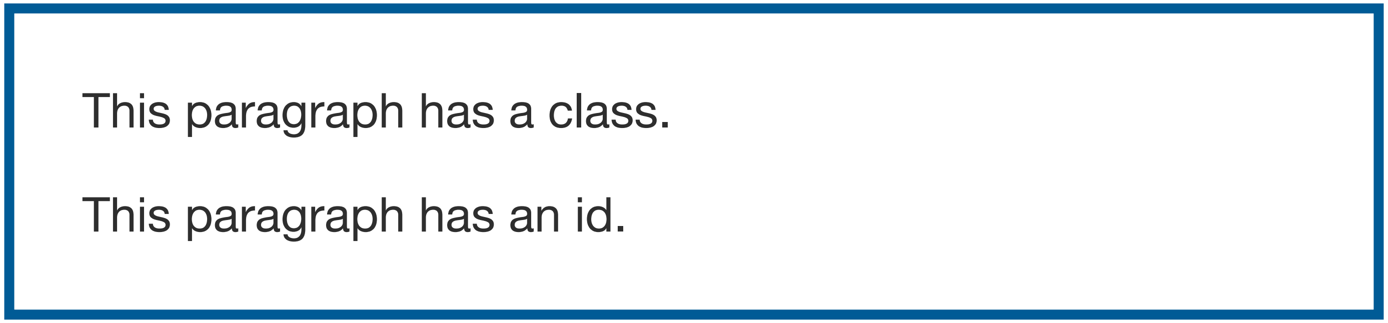
```
d3.selectAll(”circle”);
```

```
d3.selectAll(”p”);
```

Class and ID attributes

```
<p class=”myclass”>This paragraph has a class.</p>

<p id=”myid”>This paragraph has an id.</p>
```



This paragraph has a class.

This paragraph has an id.

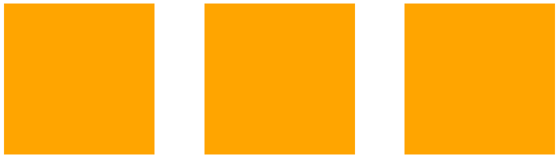
Selecting by ID

```
<p id="id1">Watch me grow.</p>
```

Watch me grow.

```
d3.select("#id1").transition().duration(3000)
  .style("font-size", "72px");
```

Selecting by class



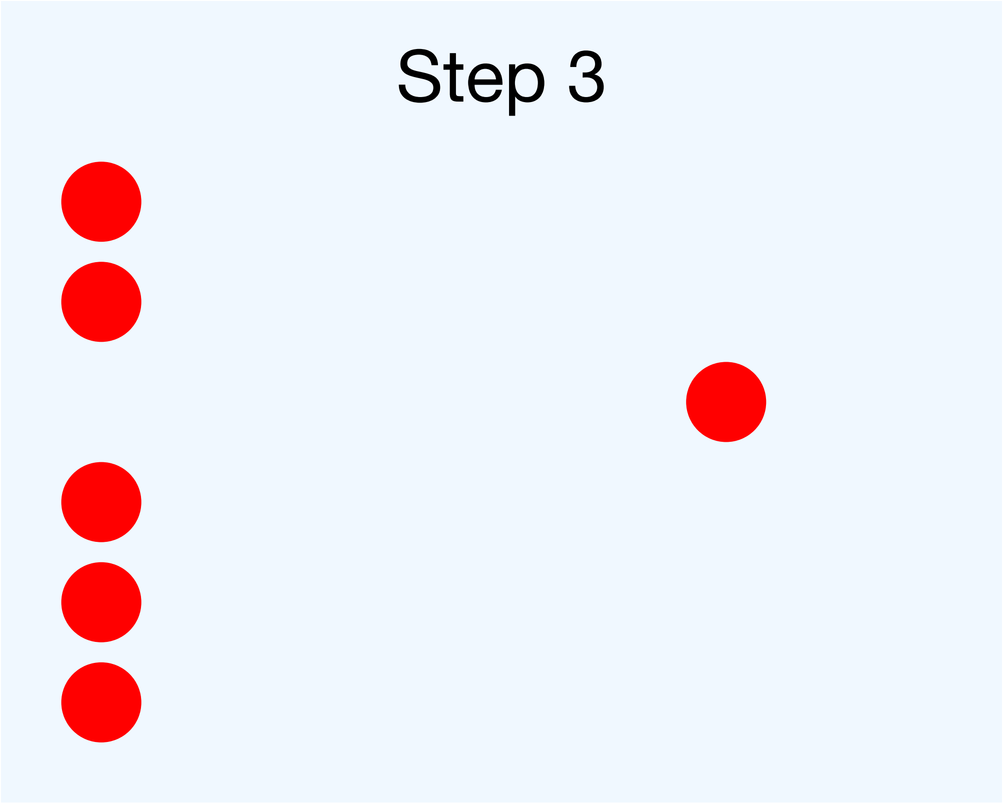
D3:

```
d3.selectAll("rect.trio").transition().duration(3000)
  .attr("y", "150").attr("fill", "orange");
```

PRACTICE

Download and open EDAV2.html

1. Move all the circles to the right.
2. Move them back to the left **and** change their color.
3. Add an id to one circle, and then move only that circle to the right.
4. Move all the circles to the middle of the screen, **then** move them all to the same location.



Modifying elements

1. Attributes
2. Styles
3. Text

https://github.com/d3/d3-selection#modifying-elements

Attributes

HTML

```
<p>Paragraph</p>
```

```
d3.select("p").attr("id", "p1");
```

```
<p id="p1">Paragraph</p>
```

Attributes

SVG

```
<circle cx=”50” cy=”100” r=”5”></circle>
```

```
d3.select(”circle”).attr(”r”, ”100”);
```

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

Styles

HTML

```
<p id="id5" style="color: green;">It's not easy being green</p>
```

```
d3.select("#id5").style("color", "red");
```

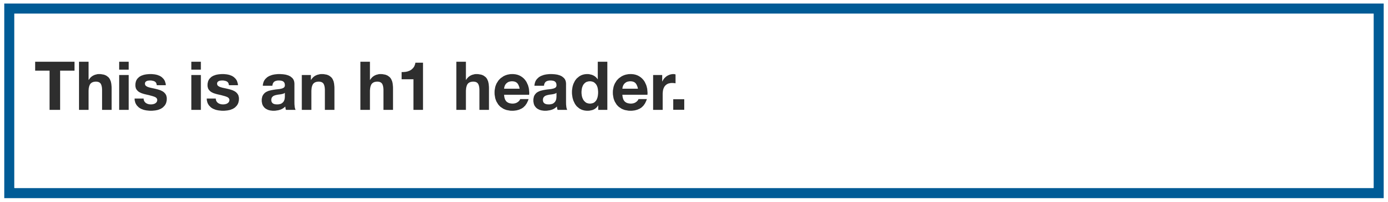
```
<p id="id5" style="color: red;">It's not easy being green</p>
```

Styles

Rough timeline of HTML / CSS history:

Early 90s:

```
<h1>This is an h1 header.</h1>
```



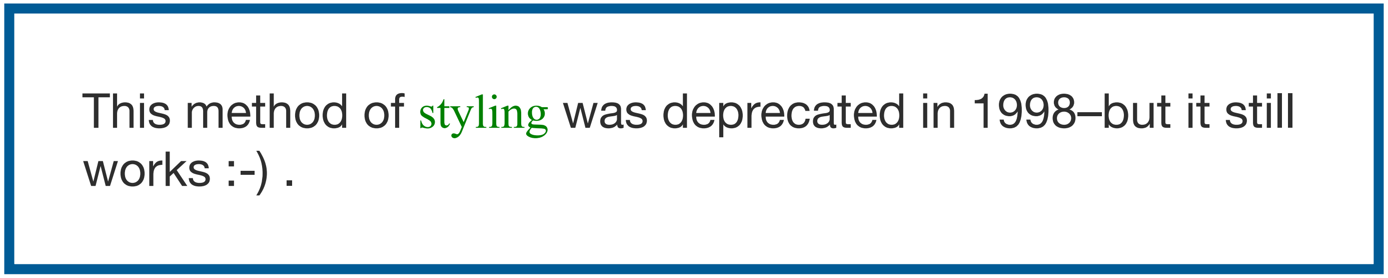
This is an h1 header.

http://www.pmichaud.com/toast/

Styles

Mid 1990s (don’t use):

```
<p>This method of <font color=”green” face=”Times New Roman”>
  styling</font> was deprecated in 1998--but it still works :-) .</p>
```



This method of styling was deprecated in 1998–but it still works :-) .

HTML tag history

http://www.martinrinehart.com/frontend-engineering/engineers/html/html-tag-history.html

Styles: External style sheet (preferred method)

Late 1990s - present: efforts to separate style from content

```
<head>
    <link rel=”stylesheet” href=”style.css”>
</head>
```

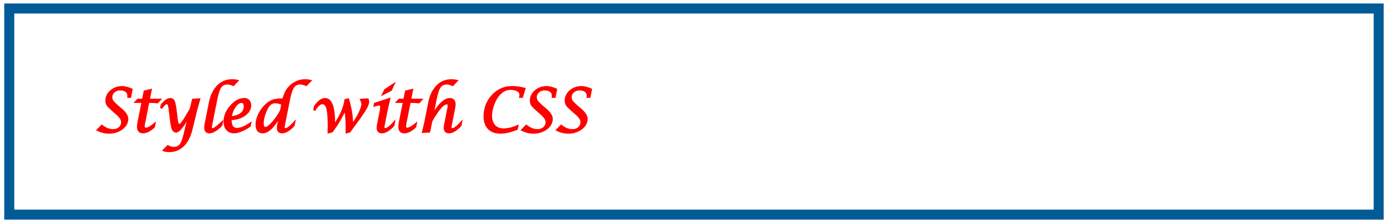
`style.css`:

```
.formal {color: red;
    font-size: 30px;
    font-family: Lucida Calligraphy;
    }
```

Styles: External style sheet (preferred method)

Body of html file:

```
<body>
  <h2 class="formal">Styled with CSS</h2>
</body>
```



Styled with CSS

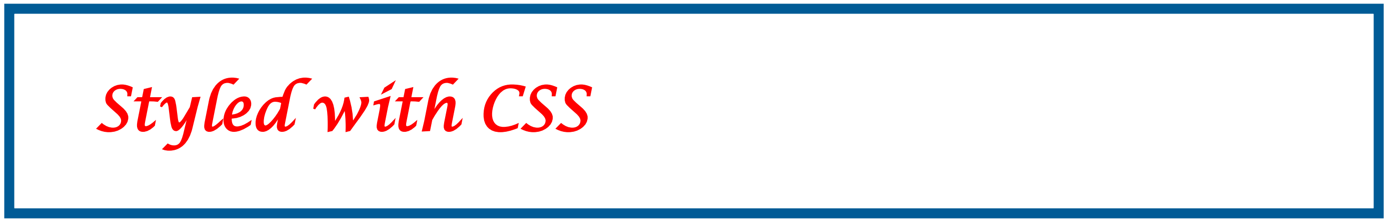
http://www.csszengarden.com/ (started 2003)

Styles: Internal style sheet

<style> tag in <head> section:

```
<head>
  <style type="text/css">
    .formal {color: red;
        font-size: 30px;
        font-family: Lucida Calligraphy;
        }
  </style>
</head>

<body>
  <h2 class="formal">Styled with CSS</h2>
</body>
```

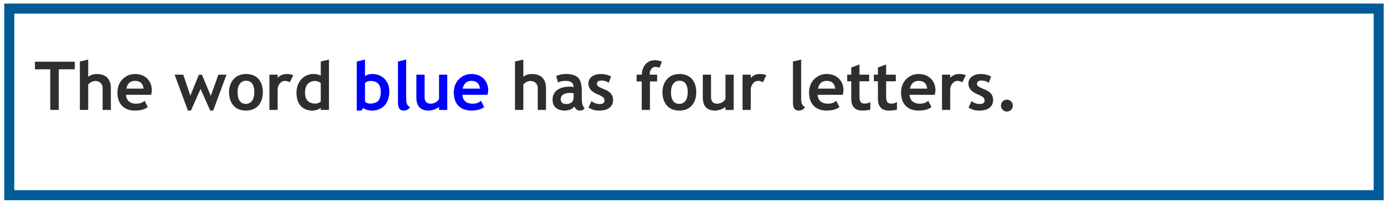


Styled with CSS

Styles: Inline style attributes

- Not recommended if you are adding styling manually
- However, JavaScript/D3 add styling **inline**

```
<h1 style="font-family: Bookman;">The word
<span style="color: blue;">blue</span>
has four letters.></h1>
```



The word blue has four letters.

- Note that style is an **attribute** (but gets handled differently by D3.)

view-source:http://www.dolekemp96.org/agenda/issues/education.htm

Modifying style attributes

```
<p id="id4">This is a paragraph.</p>
```

```
d3.select("#id4").style("color", "red");
```

But since style is an attribute, this would also work:

```
d3.select("#id4").attr("style", "color: red;");
```

The change to the DOM is the same in either case:

```
<p id="id4" style="color: red;">This is a paragraph.</p>
```

Modifying HTML text

```
<p id="typo”>Manhatten</p>
```

Manhatten

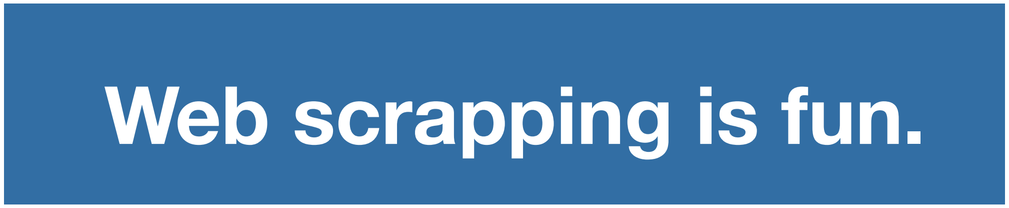
Hover to execute this code (and fix the typo):

```
d3.select("#typo”).text("Manhattan”);
```

Modifying SVG text

```
<svg width="500" height="100">
  <rect width="500" height="100" fill="#326EA4"></rect>
<text id="svgtypo" x="50" y="70" fill="white"
    font-weight="bold" font-size="40px">
    Web scrapping is fun.</text>
</svg>
```

svg



Web scrapping is fun.

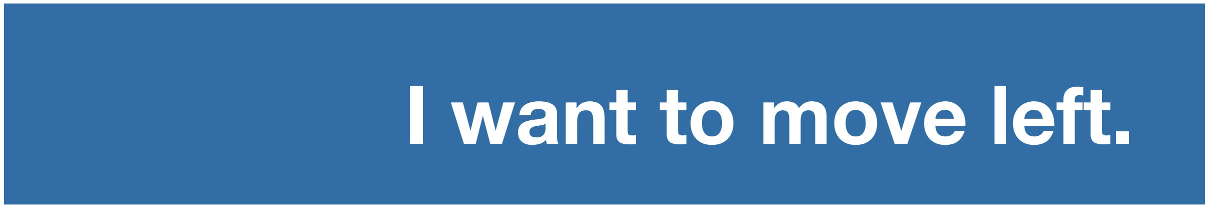
Hover to execute this code (and fix the typo):

```
d3.select("#svgtypo").text("Web scraping is fun.");
```

Moving (and modifying) SVG text

```
<svg width="600" height="100">
  <rect width="600" height="100" fill="#326EA4"></rect>
  <text id="moveleft" x="200" y="70" fill="white"
      font-weight="bold" font-size="40px">
      I want to move left.</text>
</svg>
```

svg



I want to move left.

Hover to execute this code:

```
d3.select("#moveleft").attr("x", "20").text("Thanks, now I'm happy!");
```

Modifying elements

Summary:

```
d3.select("p").attr("id", "myid");

d3.select("h1").style("color", "red");

d3.select("text").text("Changing some svg text.");
```

Modifying elements heads-up

Text color

HTML

```
d3.select("p").style("color", "red");
```

SVG

```
d3.select("text").style("fill", "red");
```

Modifying elements heads-up

SVG styles vs. attributes

```
d3.select("circle”).style("fill”, "red”);
```

```
<circle cx="50” cy="50” r="50” style="fill: red;"></circle>
```

OR

```
d3.select("circle”).attr("fill”, "red”);
```

```
<circle cx="50” cy="50” r="50” fill="red"></circle>
```

(If both are specified, style takes precedence.)

Adding elements

HTML

```
d3.select("body").append("p");
```

SVG

```
d3.select("svg").append("circle");
```

Removing elements

HTML

```
d3.select(”p”).remove();
```

SVG

```
d3.select(”circle”).remove();
```

PRACTICE 2

Download and open a fresh copy of EDAV2.html

1. Add a class to one of the circles.
2. Use an internal style sheet to style that circle with a green fill, orange border (“stroke”), and stroke width (“stroke-width”) of 5.
3. Open the JavaScript Console and use D3 to change the class of all the circles to your newly created class.
4. Use D3 to add an svg text element so that the top circle has a “1” in white in its center.
5. Use D3 to transition the text to the second circle, changing the text to “2”. (Note that the text changes immediately at the start of the transition.)