

Assignment 1

This assignment is adapted from a [similar one](#) from the University of Massachusetts, Dartmouth.

Goals

Get started with Web programming. This assignment will introduce (or reintroduce) you to [HyperText Markup Language \(HTML\)](#), [Cascading Style Sheets \(CSS\)](#), [Scalable Vector Graphics \(SVG\)](#), and [JavaScript](#).

Instructions

There are four parts for this assignment. All students should complete the first three; the fourth is highly recommended. You will need to know how to do all this stuff soon. You may complete the assignment in a single HTML file or use multiple files (e.g. one for CSS, one for HTML, and one for JavaScript) in a Zip archive. **You may not use D3 or any other external libraries for any of this assignment** You may refer to the screenshot of a [sample solution](#), but note that because your solution must use your own name, it will look different. The [Mozilla Developer Network documentation](#) serves as a great reference for the technologies involved in this assignment. In addition, Scott Murray has written a great book named [Interactive Data Visualization for the Web](#), which is available for free on the Web; [Chapter 3](#) explains the basics of SVG

Due Date

Due to snow days, the assignment is now due at 5pm on **Monday, February 16**.

Submission

You should submit any files required for this assignment on [myCourses](#). You may complete the assignment in a single HTML file or use multiple files (e.g. one for HTML, one for CSS, and one for JavaScript). Note that the files should be linked to the main HTML document accordingly. The filename of the main HTML document should be `assignment1.html`.

Details

1. Info (HTML & CSS)

Create an HTML web page with the title "Assignment 1". It should contain the following text:

- Your name
- Your student id
- The course title ("Data Visualization (CIS 467/602-01)"), and
- The assignment title ("Assignment 1")
- The text "This assignment is all my own work. I did not copy the code from any other source."

The first line should be a level-3 [heading element](#). The other three should be a [paragraph elements](#).. **Do not put anything except the header tags (with `id` and `class` attributes only) and the text in HTML.**

You now should style the text using [CSS](#) so that your name is displayed in red, your student id and the course title are rendered in bold text, and the assignment name is in italics only (not bold). **You must do this using CSS; do not use any HTML attributes except id and class!** The screenshot below shows what the final result should look like:

John Adam Smith

#000000000

Data Visualization (COM3571)

Assignment 1

This assignment is all my own work. I did not copy the code from any other source.

Hints:

- For the most efficient solution, use a `class` attribute for some of the elements.
- The black border shown around the solution is not part of the solution. Don't create one.

2. Initials (SVG)

In the *same web page*, create an [SVG](#) graphic that displays your three initials using SVG. For example, consider the name "John Adam Smith": its initials are "JAS". You do not need to draw periods after each letter. If you have more than three names, use only one of your middle names. If you do not have a middle name, use the second letter of your first name.

You may only use SVG lines, polylines, rectangles, or paths. You do not need to use curves; you may use straight lines to render letters. (See [this font](#) for ideas on how to draw letters without curves.) **Do not draw your initials using SVG or HTML text elements.** You will receive no credit if you use SVG or HTML text elements.

You should draw all content in a single svg element that has dimensions 250x150, e.g.

```
<svg id="initials" width="250" height="150">
  <!-- YOUR SVG HERE -->
</svg>
```



Example Solution for Part 2

Hints:

- Remember that SVG coordinates start from the top-left part of the element.
- Investigate the SVG [fill](#) and [stroke attributes](#) to style your initials.
- You can style SVG elements with CSS in a similar manner to HTML elements.

3. Histogram (JavaScript + SVG)

In the same web page, create a **JavaScript function** that creates a [histogram](#) that shows the distribution of characters in a string. Use this function to draw a bar chart for the letters in your first name. If your first name has characters outside of the "standard" A-Z letters, please use a version that only uses these 26 letters.

Your function should take an svg element and a string and update the svg element so that it displays a

histogram of the string with six bins (A-D, E-H, I-L, M-P, Q-U, V-Z). The number of characters that are in those ranges will determine the height of the bars for each bin. Each bin should be 50 pixels wide and each letter occurrence should contribute 50 pixels of height to the bar. The bars should be filled with a blue color and the lines drawn in black.

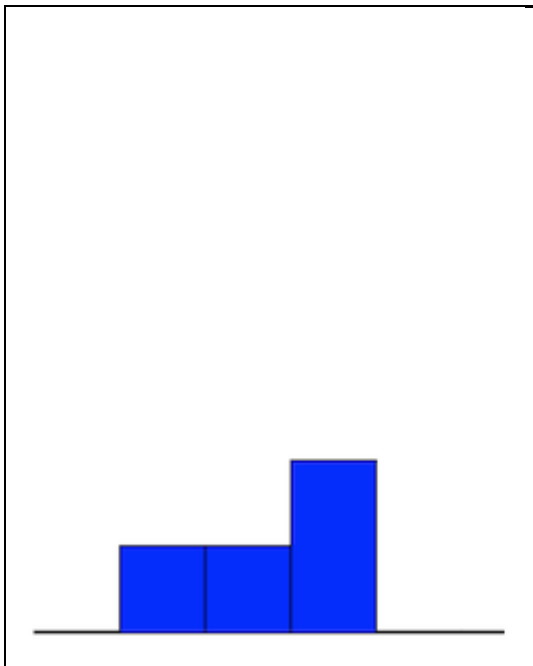
Your svg element should be empty:

```
<svg id="histogram" width="300" height="400">
  <!-- NOTHING GOES HERE -->
</svg>
```

You may use the following helper function to add the bars to the SVG element:

```
function addEltToSVG(svg, name, attrs)
{
  var element = document.createElementNS("http://www.w3.org/2000/svg", name);
  if (attrs === undefined) attrs = {};
  for (var key in attrs) {
    element.setAttributeNS(null, key, attrs[key]);
  }
  svg.appendChild(element);
}
```

You should define a function `createHistogram` such that a call `createHistogram(svgElement, "John")` will create the histogram.



Example Solution for Part 3:

```
createHistogram(svgElement, "John")
```

Hints:

- Make sure your JavaScript occurs **after** the svg elements are defined. HTML is processed sequentially. You may also use an [onload function](#).
- Give a zero bin a height of 1 pixel so that it will show on the screen. Otherwise, it will not render.
- You can access a character of a string in JavaScript via subscripts.
- You can compare characters with standard comparison operators (e.g. `<`, `>`, `=`) as in Java.
- JavaScript's `String.toUpperCase` or `String.toLowerCase` methods should make your code more robust.

- Use JavaScript's `document.getElementById` function to get a reference to the svg element.

(Not required) For a challenge, add text labels to the bins.

4. [Extra Credit] Pie Chart (JavaScript + SVG Paths)

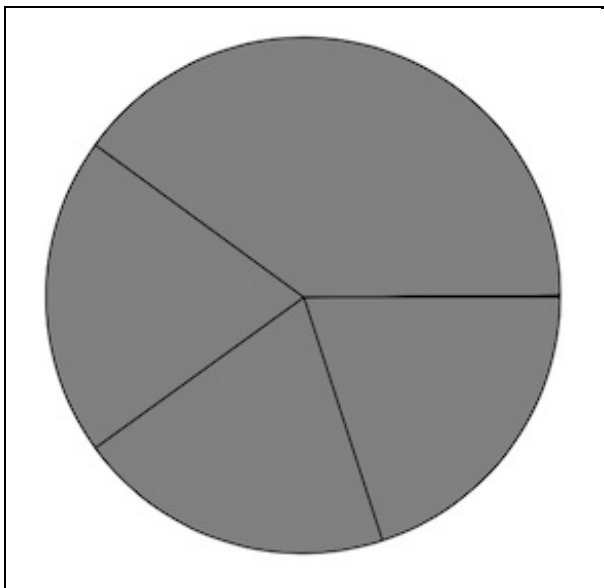
In the same web page, create a second **JavaScript function** that creates a **pie chart** that shows the distribution of characters in a string. Use this function to draw a pie chart for the letters in your last name.

Your function should take an svg element and a string and update the svg element so that it displays a pie chart that has six slices (A-D, E-H, I-L, M-P, Q-U, V-Z). Each slice be sized according to the percentage of characters from that character range (e.g. "Smith" -> [0%, 20%, 20%, 20%, 40%, 0%]). The A-D slice should start on the far right of the circle. Note that because of the 0% slices, you may see fewer than six slices. Again, you will need a second empty svg element:

```
<svg id="piechart" width="400" height="400">
  <!-- NOTHING GOES HERE -->
</svg>
```

Your pie chart should be use a circle of radius 150 pixels. Note that you will need to use **SVG paths** to draw the slices. Specifically, you will need to use **arcs** in order to draw the curved part of the path. Draw each slice with a gray fill color and a black stroke color.

You should define a function `createPieChart` such that a call `createPieChart(svgElement, "Smith")` will create the pie chart.



Example Solution for Part 4:

```
createPieChart(svgElementt, "Smith")
```

Hints:

- Trigonometry will be required here so use the **JavaScript Math library**. Calculate the angles first, and then figure out how to draw the slices based on the angles.
- Use JavaScript's string concatenation to build the path string, e.g.

```
'M 200 200 A ' + x + ' ' + y + ...
```

(Not required) For a challenge, highlight the slice with the first letter of your name in red. For a further challenge, label the pie slices.