D3 CodeAlong
Intro to D3

H. Quamen — hquamen@ualberta.ca

Loading D3

The D3 library is always accessible online and it's easy to load it from the original source. To do so, we include a **<script>** tag with a **src** attribute somewhere in our page (usually in the header).

Best practice is to load the "minified" version. It's not really human-readable, but it's readable by the computer, and it's a much smaller file:

```
<script src="https://d3js.org/d3.v7.min.js"></script>
```

However, you can always load the "normal," un-minified version (a much bigger document but human-readable):

```
<script src="https://d3js.org/d3.v7.js"></script>
```

D3 Intro Page:
   https://d3js.org/

---

1.   d3/01_circles

This tutorial adds five circles to an HTML **svg** element. We learn how to use anonymous functions to set each circle's attribute properties: **cx**, **cy**, **r**, and **fill**.

---

2.   d3/02_move_circle

This tutorial appends a circle to the HTML **svg** element and attaches an "event listener" to an HTML button. When clicked, the button activates the listener's "callback" function, which moves the circle and changes its attributes.

---

3.   d3/03_paragraphs

This tutorial works with text paragraphs on an HTML page. The D3 technique when working with text is slightly different than with **svg** elements (notice that we need to use the **.join()** command). We attach an event listener to a button. Each time the button is clicked, the top paragraph is eliminated, each remaining paragraph moves "up," and one new paragraph is added to the bottom.

4.   d3/04_background

This tutorial uses D3 to swap the background colour of a page every time a button is clicked. The page switches from black text on white background to white text on black background. To do so, we use the concept of a "global variable" so that the page can remember which state it's currently in. Given time in class, there are a few modifications we may work through.

5.   d3/05_circles_load

This tutorial teaches us how to load data from a CSV file into D3. Because Javascript has been "lobotomized" for safety, it cannot load data files, which is an action needed by practically every D3 script. Trying to load data improperly (that is, not using the HTTP protocol) will result in a "CORS" error ("Cross-Origin Resource Sharing").

To solve the problem, we need to start a mini-web-server in the directory where your code resides. To do so, use the command line to change directories to the place where your code lives. Then start the server with this Python command:

python -m http.server 8888

Next, use the address bar of your browser to surf to:

http://localhost:8888/*name_of_your_file.html*

If and when you put your final project on a real web server, you won't get the CORS error because everyone will access your project through the HTTP protocol. It's a problem for us only in the developmental stages of our project.