DS4200: Information Presentation and Visualization

Introduction to HTML/CSS/JS/D3

Xiaoyi Yang
Khoury College of Computer Sciences
Northeastern University

Goals for today

- Intro to HTML/CSS/JS
- Publish website on Github
- SVG and its coordinate system
- Basic SVG shapes
- Modify elements in JS
- Making basic figures with D3

Web design

Web design: creating and designing websites, focusing on the layout, aesthetics, and usability of a site. It combines creativity with technical skills to ensure a website is visually appealing, functional, and accessible.

- 1991: Tim Berners-Lee introduced the first website using simple HTML.
- 1993: The release of the first graphical browser, Mosaic
- 1995: JavaScript was introduced by Netscape
- 1996: The release of CSS
- 2000: The adoption of web standards (by the W3C)

Key Technologies in Web Design

- HTML (HyperText Markup Language): content and structure
- CSS (Cascading Style Sheets): visual presentation
- JavaScript (JS): interaction

HTML

- Defines headings, paragraphs, images, links, and forms.
- Used to create the structure of a webpage.
- Use tags to indicate the components
- Example: <h1>Heading</h1>, This is a paragraph .

HTML tags

- <**h1**> to <**h6**>: Heading tags.
- <**p**>: Paragraph tag for blocks of text.
- <**b**> / <**strong**>: Bold text.
- <**i**> / <**em**>: Italic text, with <**em**> adding emphasis.
- <**a**>: Anchor tag for hyperlinks.
- <**ul**>: Unordered (bulleted) list.
- <**ol**>: Ordered (numbered) list.
- <**li**>: List item (used inside <**ul**> or <**ol**>).
- <**img**>: Embeds an image.
- <**table**>: Defines a table.
- <**tr**>: Table row.
- <**td**>: Table cell.
- <**th**>: Table header cell.

CSS

- Controls fonts, colors, margins, spacing, and layouts.
- Helps in making the website responsive (adjustable for different devices).
- Layouting with grids or flexbox
- Can be written inline, internal and external
- Example:
  <h1 style="color: blue; font-size: 24px;"> Hello, World!</h1>

HTML with external CSS file

In ‘main.html‘:

```html
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" type="text/css" href="styles.css">
</head>
<body>
    <h1>Hello, World!</h1>
</body>
</html>
```

In ‘styles.css‘:

```css
body {
    background-color: lightgray;
}
h1 {
    color: blue;
    font-size: 24px;
}
```

JavaScript

- Adds interactivity and dynamic content updates.
- Can modify HTML/CSS in real-time without reloading the page.
- Can be written inline, internal and external
- Example: Show/hide elements, form validation, animations.

Code demonstration

Github

- GitHub is a web-based platform for version control and collaboration.
- Built on top of **Git**, a distributed version control system created by Linus Torvalds.
- It allows multiple people to collaborate on projects, track changes, and maintain a history of their code.
- Popular among developers for open-source and private projects.

Why Use GitHub?

- Version Control: Keep track of changes in your code and collaborate with others.
- Collaboration: Multiple contributors can work on the same project.
- Backup: Your projects are stored in the cloud.
- Open Source Community: Share your work with others and contribute to projects.

In this course: we are going to hold the project on Github.
Beyond this course: you can hold your personal website on Github.

Demonstration

Be a "designer"

Now, we know how to make a website (even though it is a simple one).

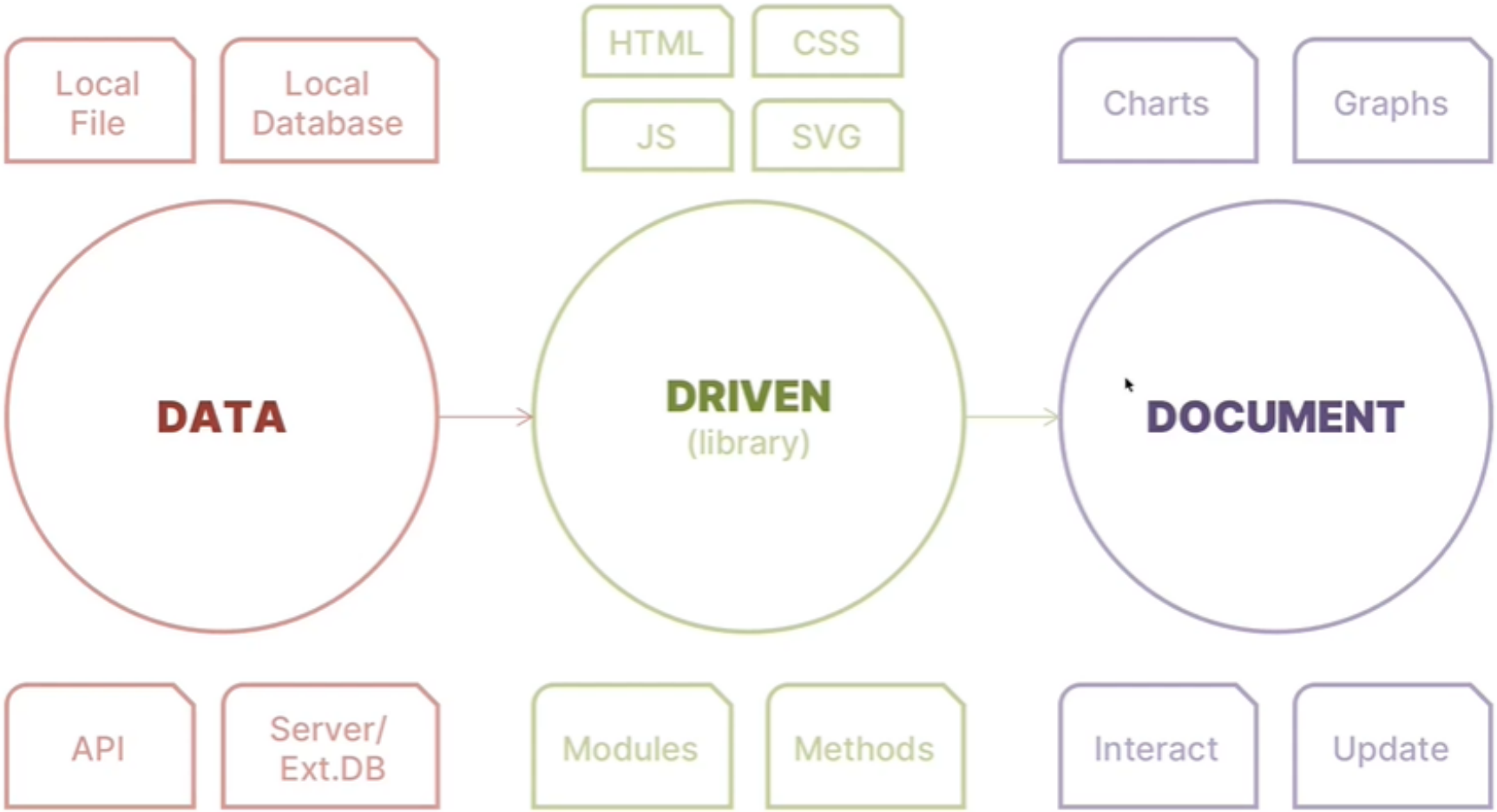
Let's explore another use on the JavaScript: svg and D3.js

Grab a paper. Draw a boxplot. Think about each steps you have done.

D3.js

- https://d3js.org
- D3 (Data-Driven Documents) is a powerful JavaScript library for creating interactive and dynamic data visualizations in web applications
- Not a charting or data visualization library (it's not like Altair, ggplot2, plotly, matplotlib, seaborn...)
- No out of the box charts (no functions to automatically build a chart)

D3.js

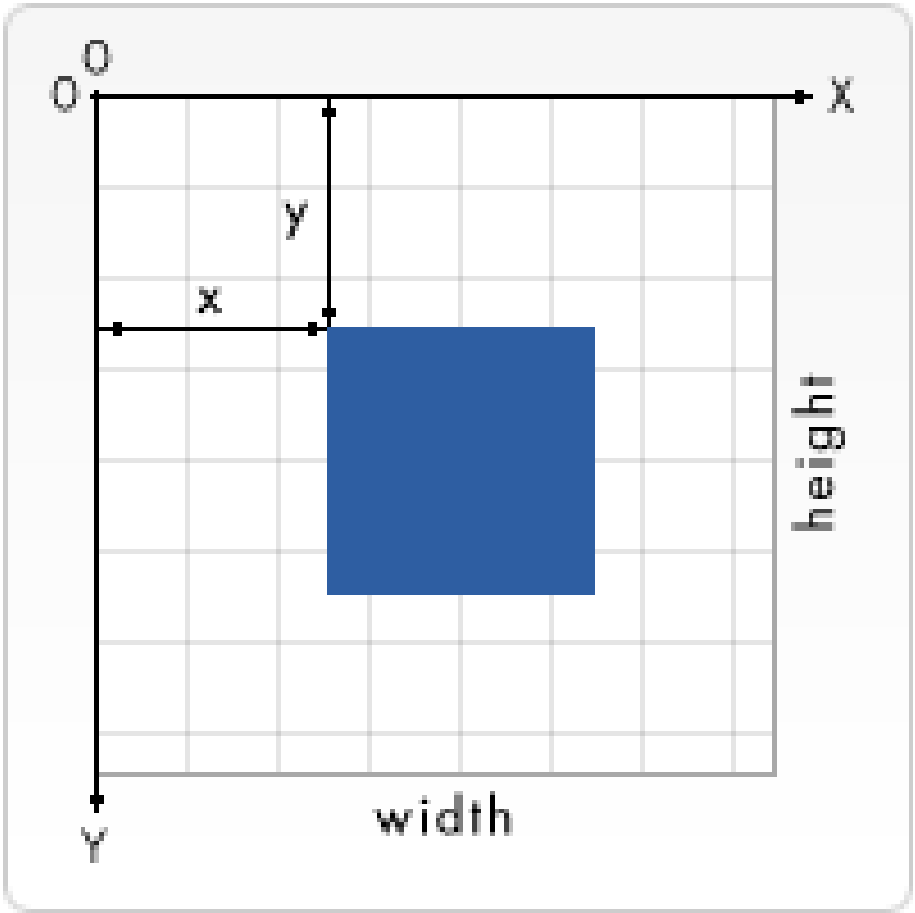

Local File
Local Database
HTML
CSS
JS
SVG
Charts
Graphs
DATA
DRIVEN
(library)
DOCUMENT
API
Server/ Ext.DB
Modules
Methods
Interact
Update

SVG vs. raster

SVG (Scalable Vector Graphics) is an XML-based format used for describing two-dimensional vector graphics.

- Formulas that describe the lines and points that make up an image
- Independent from the size of an image
- Always looks crisp, no matter how much you zoom in or distort the picture
- Graphics in SVG will be heavier to process

SVG coordinate system



SVG example

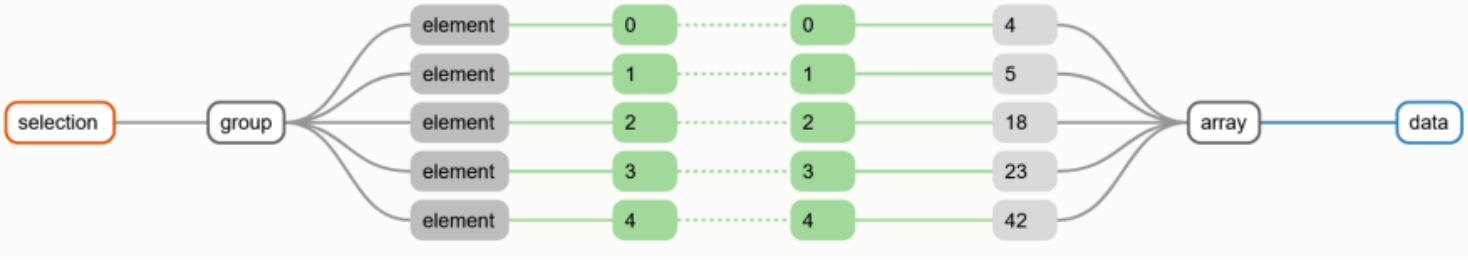
- <svg> tag. E.g., <svg width='500' height='500'>
- Can add <style> attributes
- Basic SVG shapes: rect, circle, line, text, polyline
- Can group elements using the <g> tag

Selections

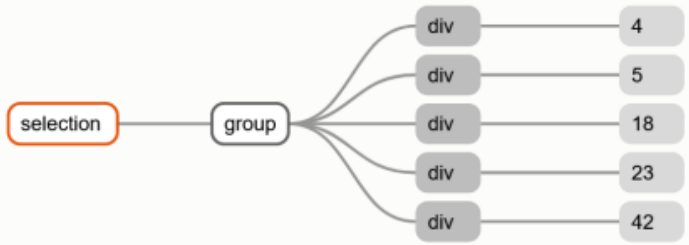
- .select (‘selectors’)
- .selectAll (‘selectors’)
- .select(‘tagname’)
- .select(‘#id’)
- .select(‘.classname’)

Data Binding

```
var numbers = [4, 5, 18, 23, 42];
```




```
d3.selectAll("div").data(numbers);
```



https://bost.ocks.org/mike/selection/

Examples

- scaleLinear( ) // Quantitative attributes
- scaleBand( ) // categorical attributes
- domain( ) // Original values that you will modify
- range ( ) // Values that we want to scale our data to
- padding() // e.g., to control the spacing in between the bars
- transition( ) // creates a transition
- duration( ) // adding duration to the transition
- delay( ) // effect is not going to take place until after this time