The Ohio State University

Introduction to D3.js

By Rui Li
01/12/2023

Slide Material Source Credits

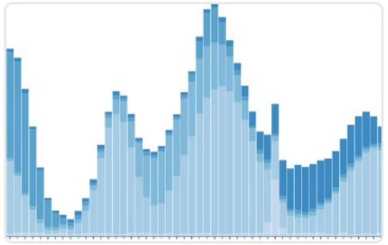
- https://d3js.org/
- https://www.d3indepth.com/
- https://d3-graph-gallery.com/
- https://observablehq.com/@d3/gallery
- Prof. Han-Wei Shen, Jiayi Xu, and Wenbin He

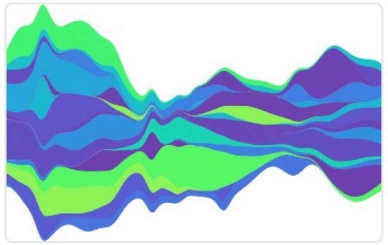
Outline

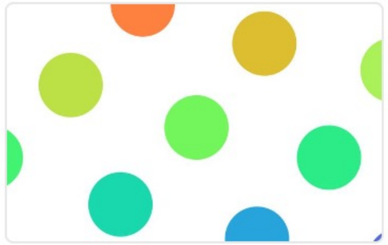
What is D3?

Data-Driven Documents

- It is an open-source JavaScript library developed by Mike Bostock to create custom interactive data visualizations in the web browser using SVG, HTML and CSS.
- Official website: d3js.org

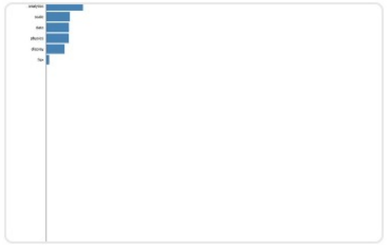

Stacked-to-grouped bars

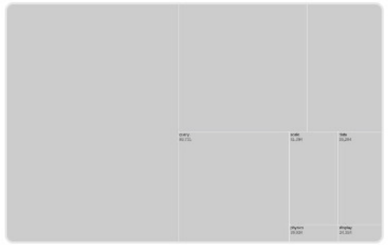

Streamgraph transitions

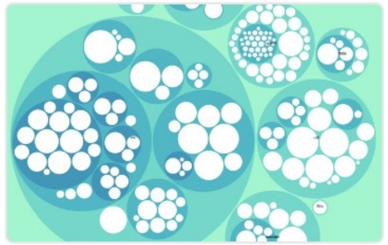

Smooth zooming

Zoom to bounding box

Walmart's growth


Hierarchical bar chart


Zoomable treemap


Zoomable circle packing

credit by: https://observablehq.com/@d3/gallery

Technology foundations

Web technologies

- HTML
  - SVG
- CSS
- JavaScript

HTML CSS JS

What we need to start

- Editor
  - Visual studio code (preferred)
    - Live server plugin
  - Sublime
  - WebStorm
  - Vim
- Web browser

What we need to start

- Editor
- Web browser
  - Chrome (preferred)
  - Edge
  - Safari

HTML - Hyper Text Markup Language

- HTML is the standard markup language for creating Web pages
  - HTML describes the structure of Web pages using markup
- HTML elements
  - HTML elements are the building blocks of HTML pages
  - represented by tags

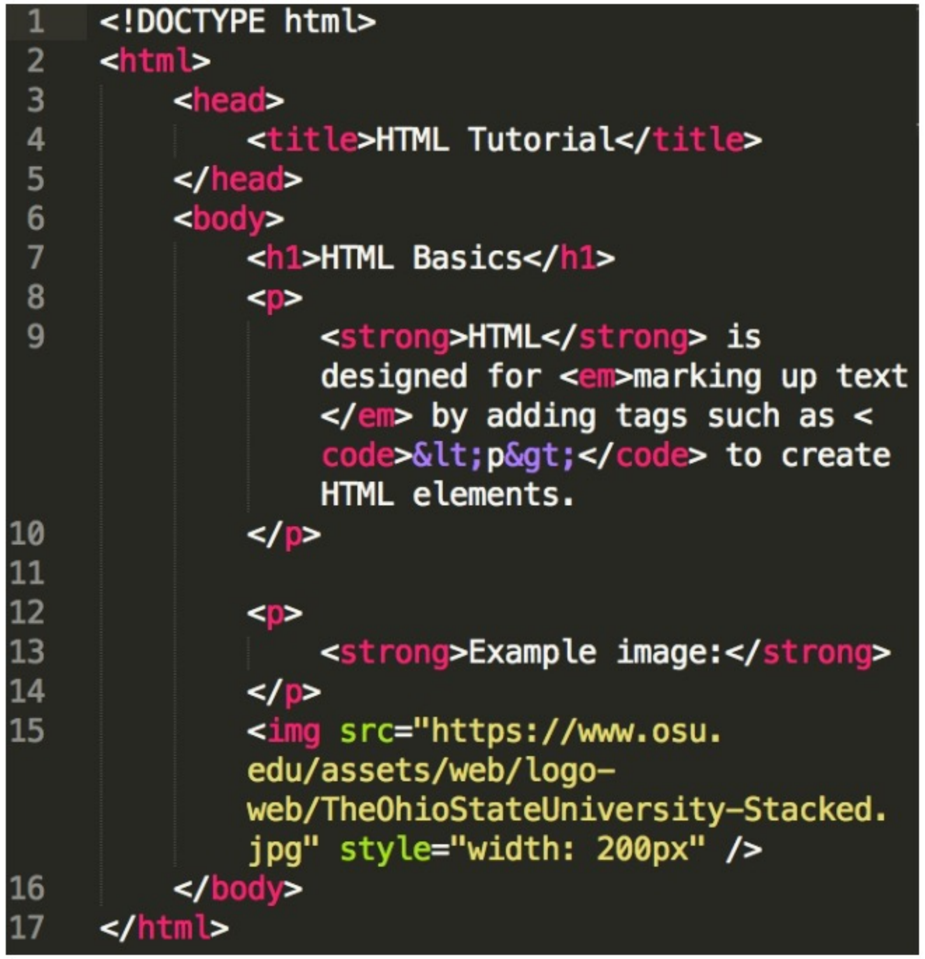
Tag syntax

- <tagname>content</tagname>
  - <h1>This is a h1 tag</h1>
- HTML tags label pieces of content such as
  - <head> tag for “heading”
  - <p> for “paragraph”
  - <table> for “table” and so on
- Browsers do not display the HTML tags, but use them to render the content of the page

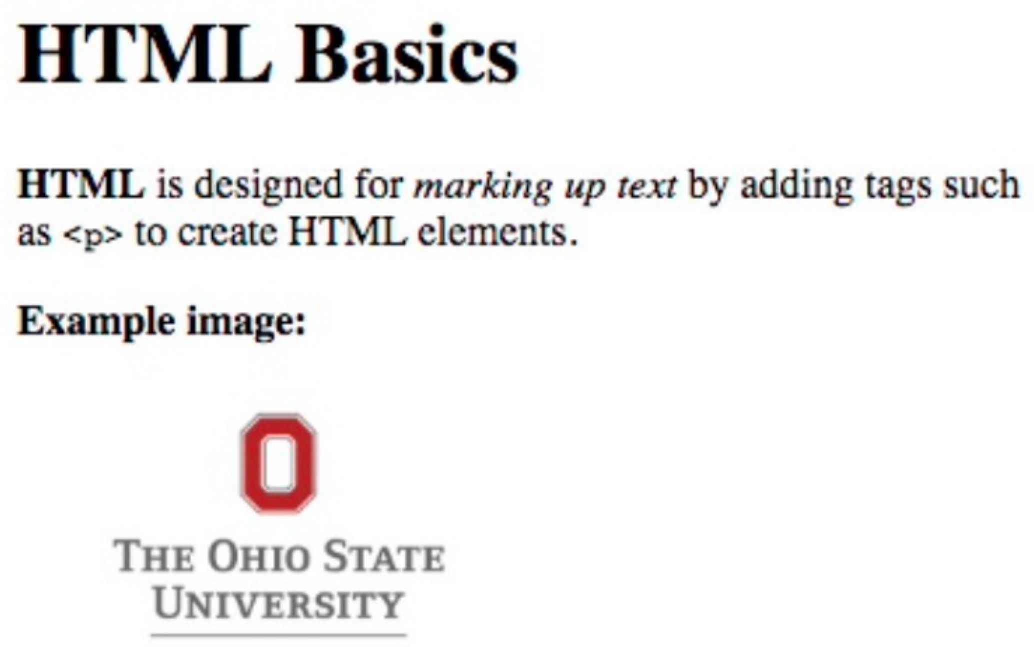
HTML – Tag attributes

- All tags can have attributes
- Provide information about an element
- Key/value pairs
- There are some pre-defined attributes
  - id
  - class
  - src
  - …

HTML - Codes and the Result



```
<!DOCTYPE html>
<html>
    <head>
        <title>HTML Tutorial</title>
    </head>
    <body>
        <h1>HTML Basics</h1>
        <p>
            <strong>HTML</strong> is
            designed for <em>marking up text
            </em> by adding tags such as <
            code>&lt;p&gt;</code> to create
            HTML elements.
        </p>

        <p>
            <strong>Example image:</strong>
        </p>
        <img src="https://www.osu.
        edu/assets/web/logo-
        web/TheOhioStateUniversity-Stacked.
        jpg" style="width: 200px" />
    </body>
</html>
```



HTML Basics

HTML is designed for *marking up text* by adding tags such as `<p>` to create HTML elements.

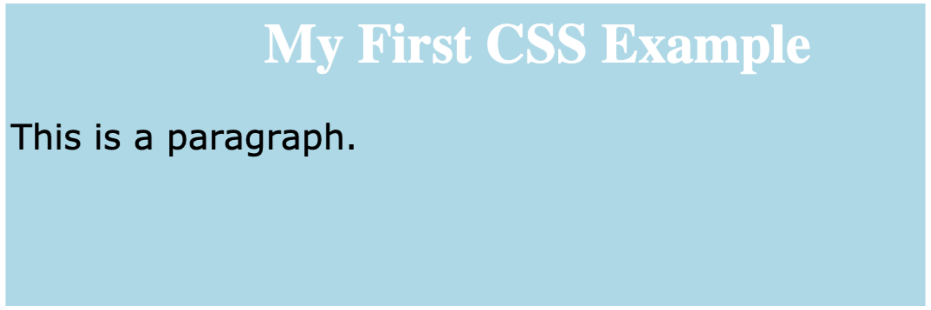
Example image:

Demo

CSS - Cascading Style Sheets

- CSS describes how HTML elements are to be displayed on screen
- It can control the appearance of multiple elements and web pages all at once

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```


My First CSS Example

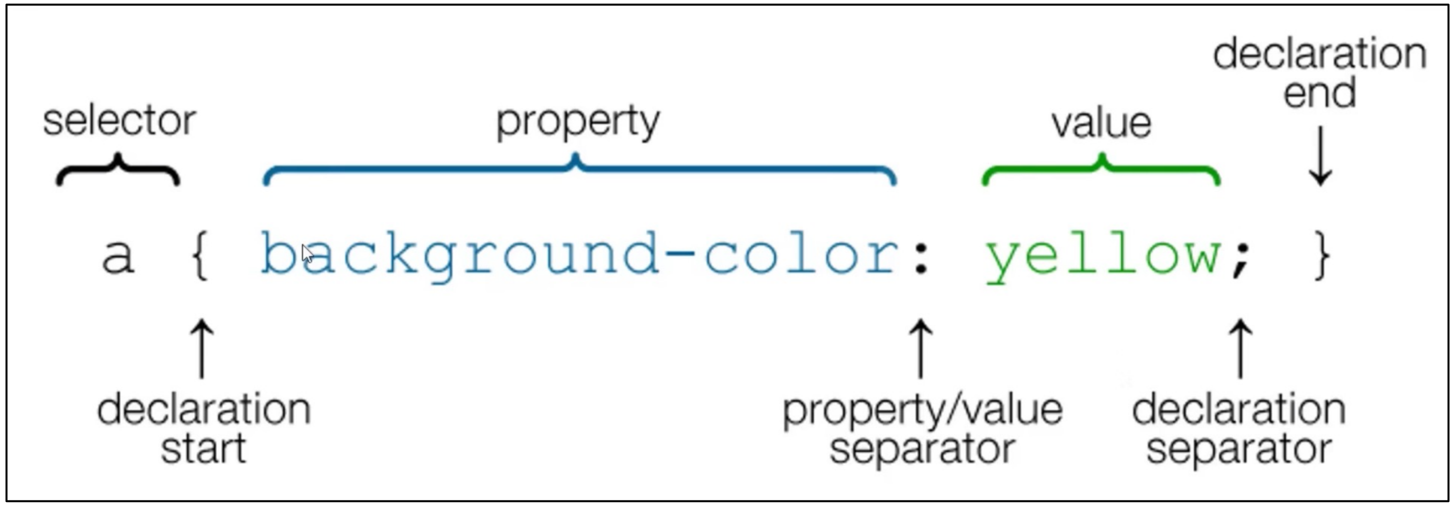
This is a paragraph.

How to add CSS

- Inline CSS: directly in the html element (No!)
- Internal CSS: using <style> tags within a single document
- External CSS: linking an external .css file
  - <link rel="stylesheet" href="style.css">

Demo

CSS Selectors

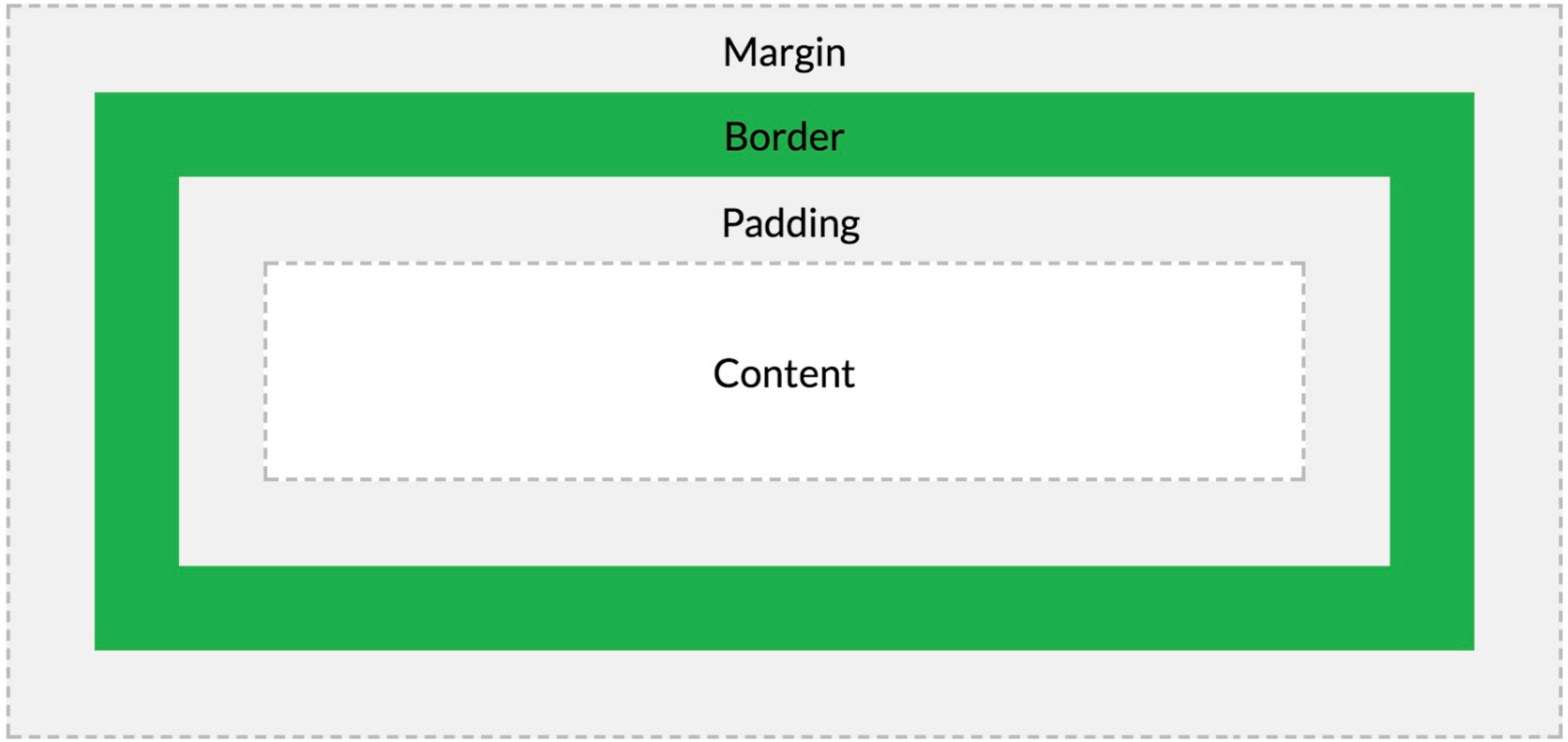


credit by: https://www.youtube.com/watch?v=yfoY53QXEnI

```
body {
  background-color: lightblue;
}

h1 {
  color: white;
  text-align: center;
}

p {
  font-family: verdana;
  font-size: 20px;
}
```

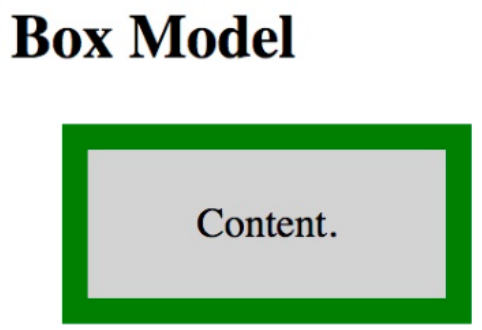
- select by
  - tag name
  - class attribute (.class)
  - id attribute (#id)

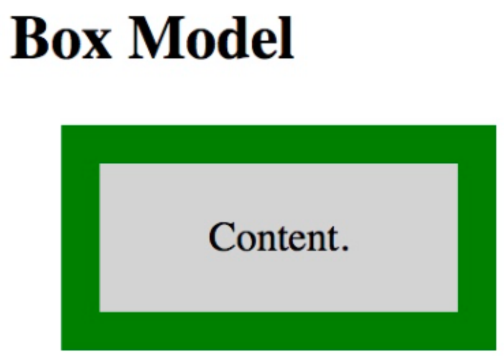
CSS – Box Model

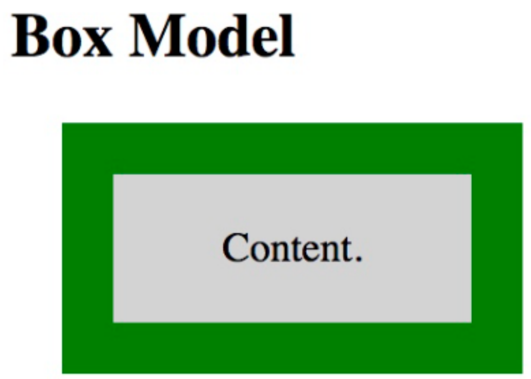


CSS – Box Model

- Border
- 10px 15px 20px

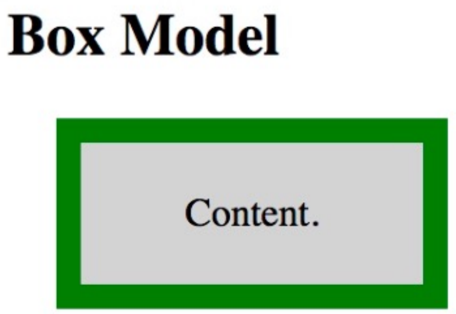


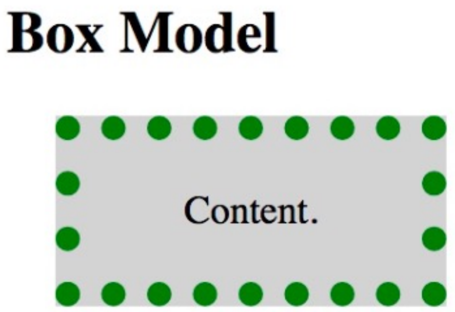


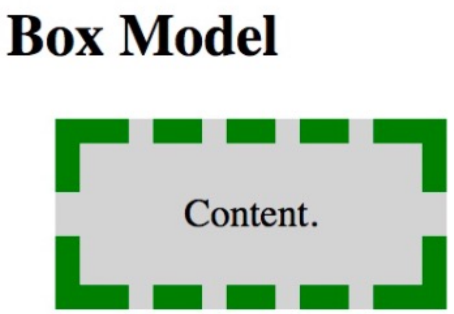


CSS – Box Model

- Border style
- solid dotted dashed







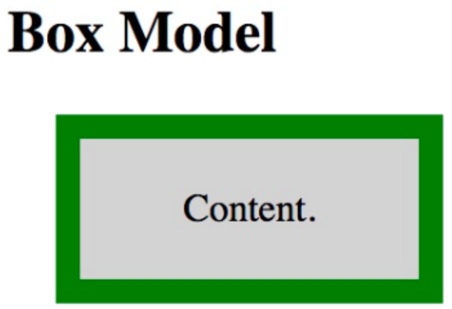
- Other styles
  - double, groove, ridge, insert, outset, none, hidden

CSS – Box Model

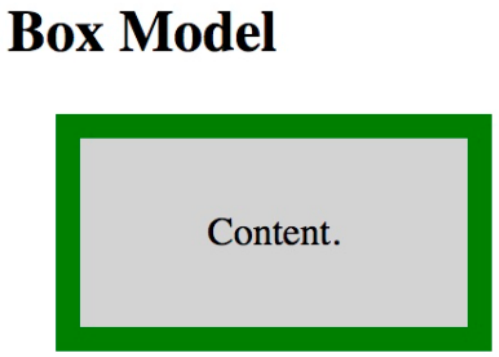
- Padding
- 10px

15px

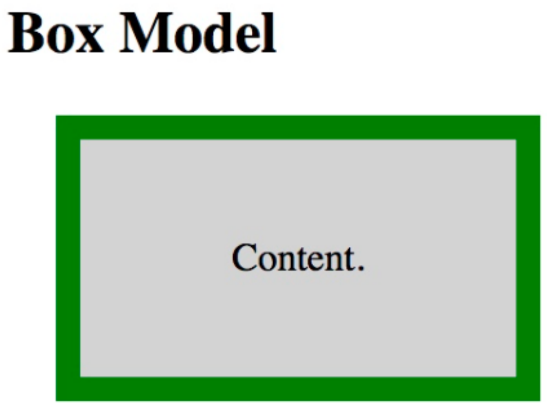
20px


Box Model

Content.


Box Model

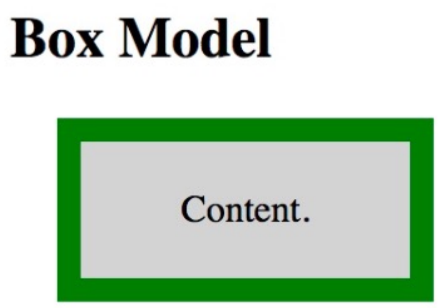
Content.


Box Model

Content.

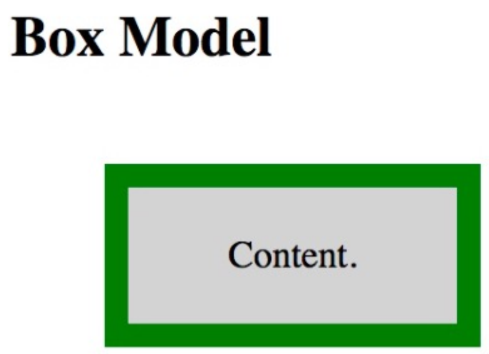
CSS – Box Model

- Margin
- 20px   40px   60px



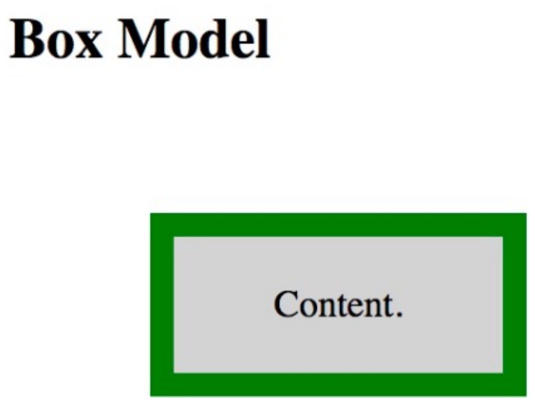
Box Model

Content.



Box Model

Content.

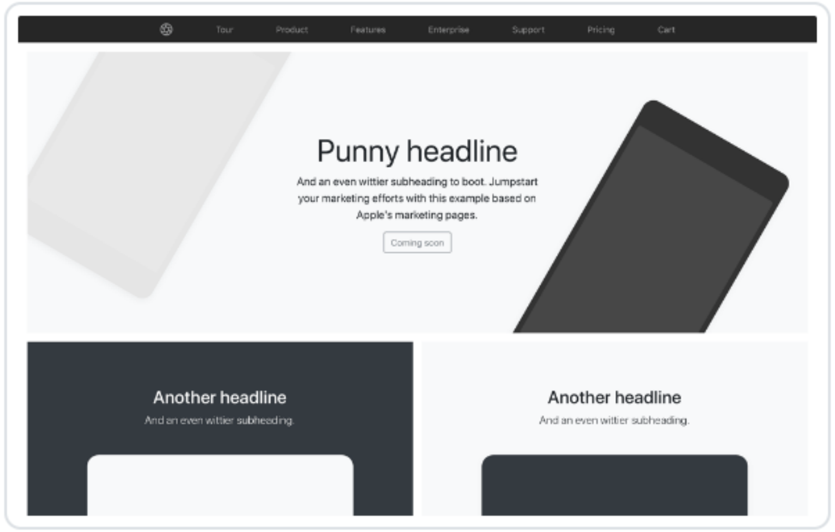


Box Model

Content.

CSS – Bootstrap (optional)

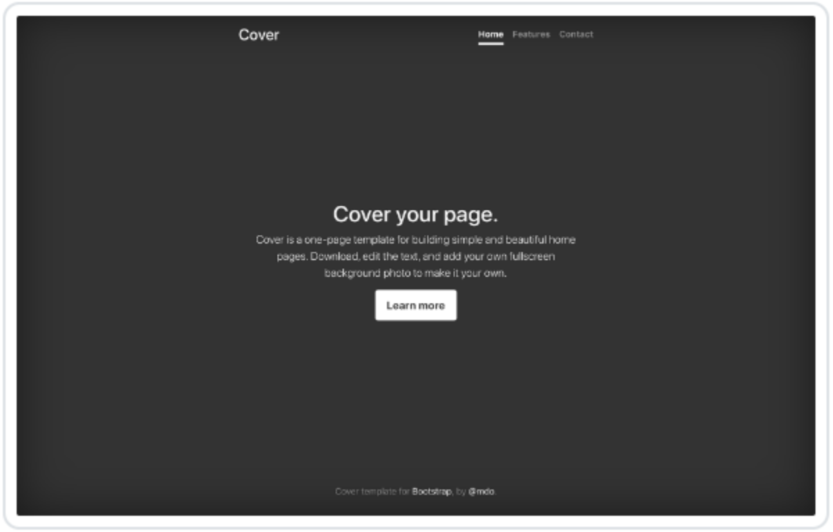
- A free and open-source CSS framework directed at responsive, mobile-first front-end web development.
- link



And an even wittier subheading to boot. Jumpstart your marketing efforts with this example based on Apple's marketing pages.

And an even wittier subheading.

And an even wittier subheading.



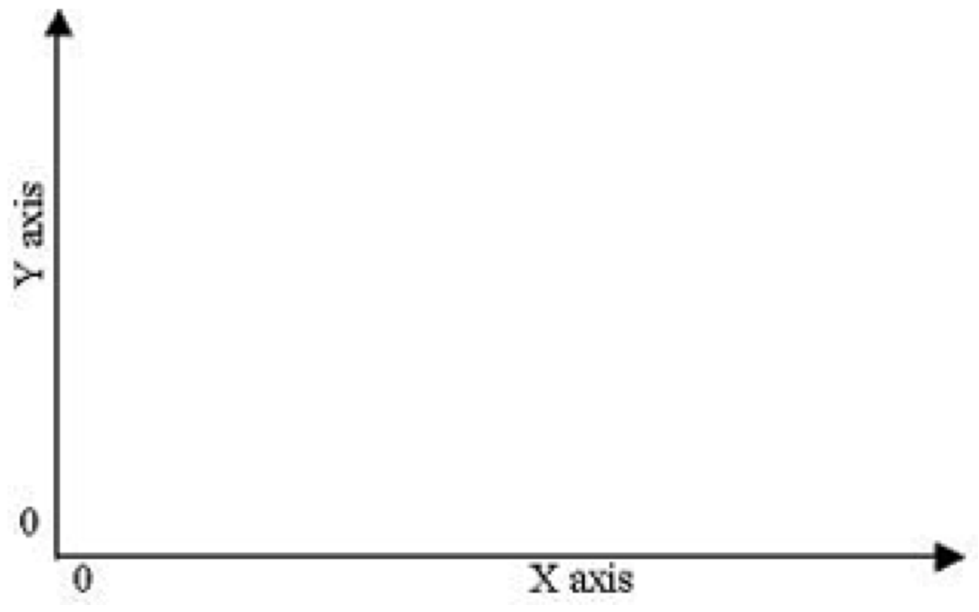
Cover is a one-page template for building simple and beautiful home pages. Download, edit the text, and add your own fullscreen background photo to make it your own.

Cover template for Bootstrap, by @mdo.

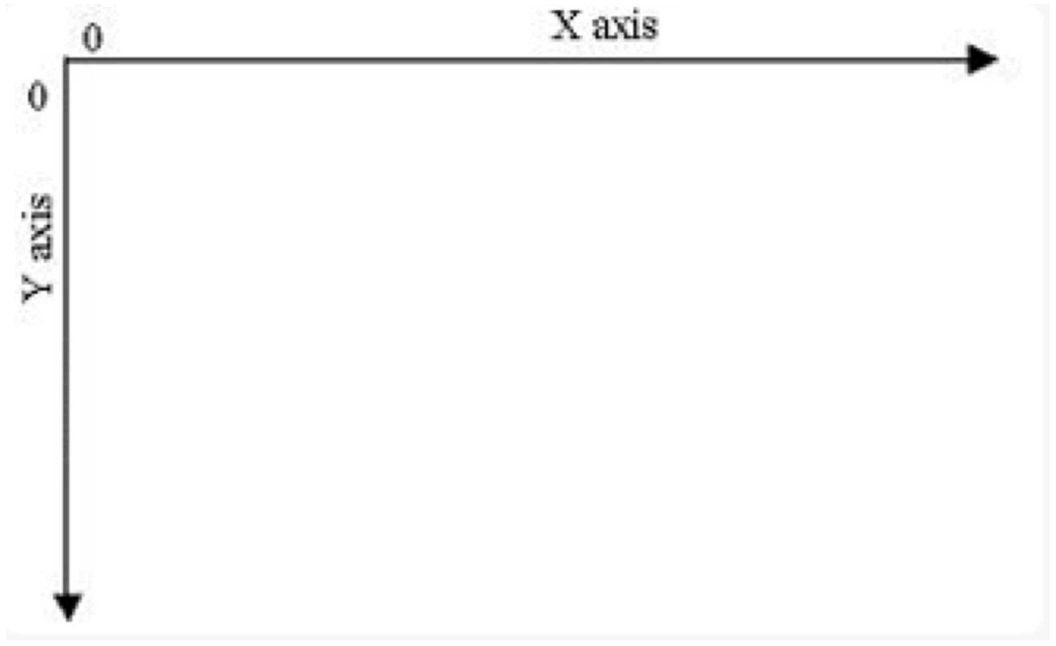
SVG (scalable vector graphics)

- SVG defines 2D vector-based graphics for Web
- SVG HTML tag

```
<svg width="500" height="50"> </svg>
```



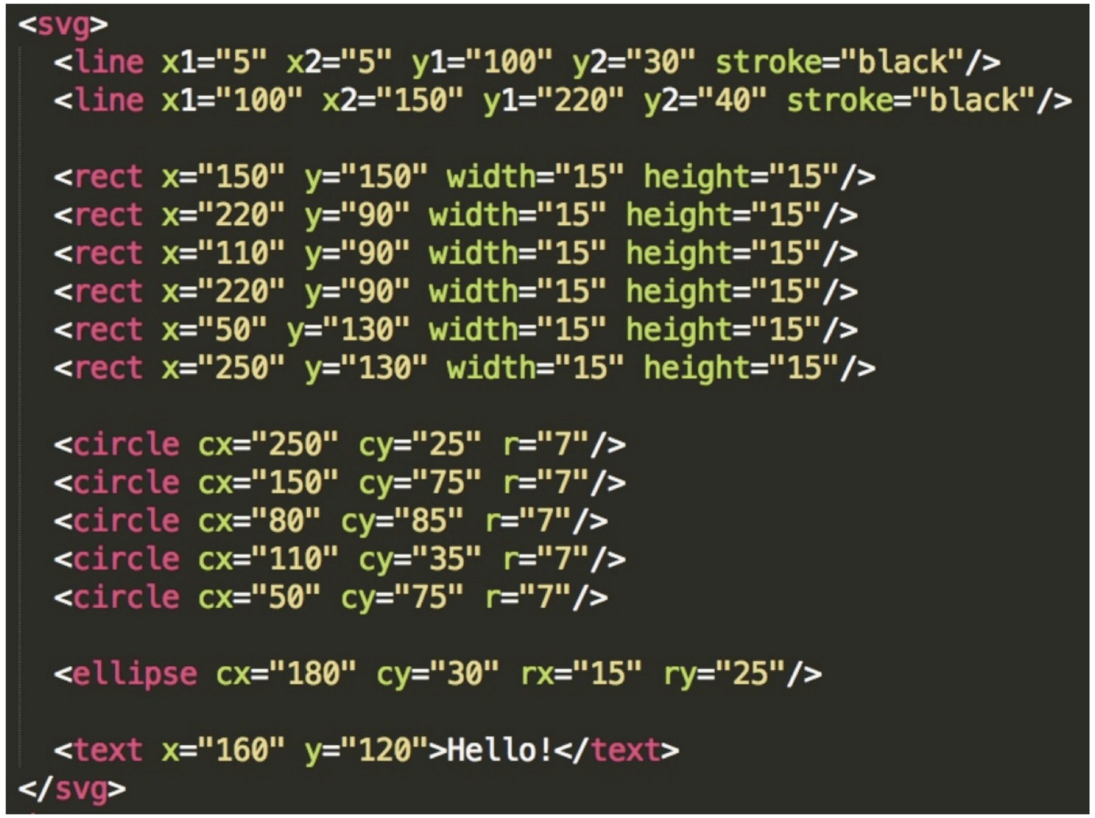
Mathematical coordinate



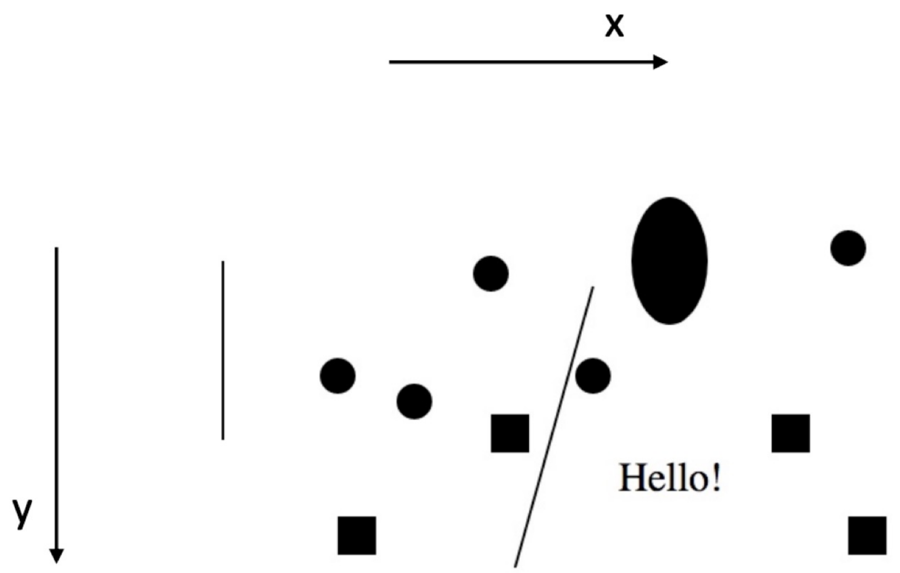
SVG coordinate space

SVG (scalable vector graphics)

- SVG is vector based and composed of shapes.

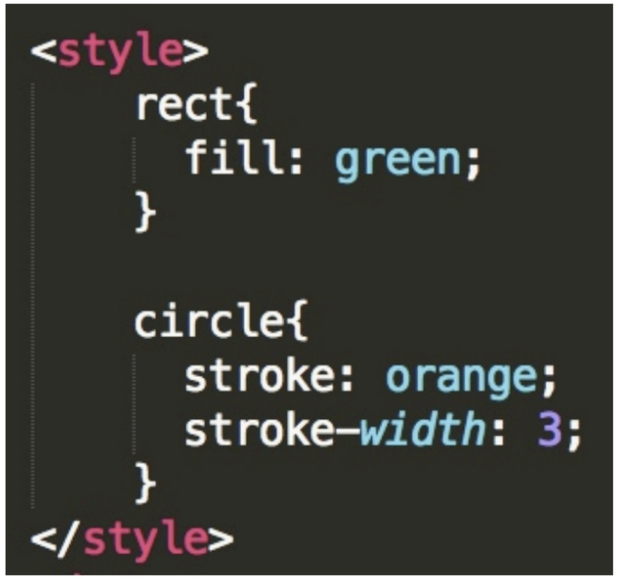


```
<svg>
  <line x1="5" x2="5" y1="100" y2="30" stroke="black"/>
  <line x1="100" x2="150" y1="220" y2="40" stroke="black"/>

  <rect x="150" y="150" width="15" height="15"/>
  <rect x="220" y="90" width="15" height="15"/>
  <rect x="110" y="90" width="15" height="15"/>
  <rect x="220" y="90" width="15" height="15"/>
  <rect x="50" y="130" width="15" height="15"/>
  <rect x="250" y="130" width="15" height="15"/>

  <circle cx="250" cy="25" r="7"/>
  <circle cx="150" cy="75" r="7"/>
  <circle cx="80" cy="85" r="7"/>
  <circle cx="110" cy="35" r="7"/>
  <circle cx="50" cy="75" r="7"/>

  <ellipse cx="180" cy="30" rx="15" ry="25"/>

  <text x="160" y="120">Hello!</text>
</svg>
```

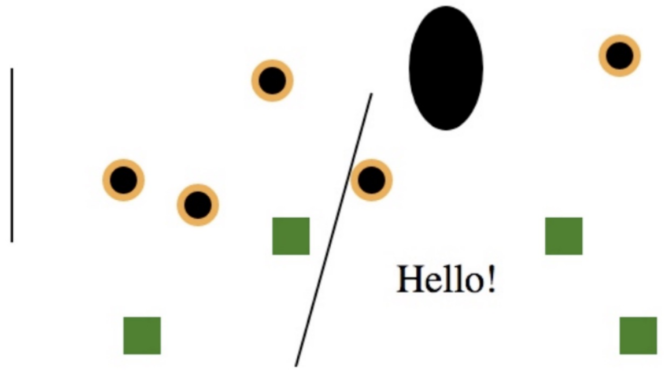


SVG (scalable vector graphics)

- SVG can be modified through script and CSS
- canvas can only be drawn/modified through script

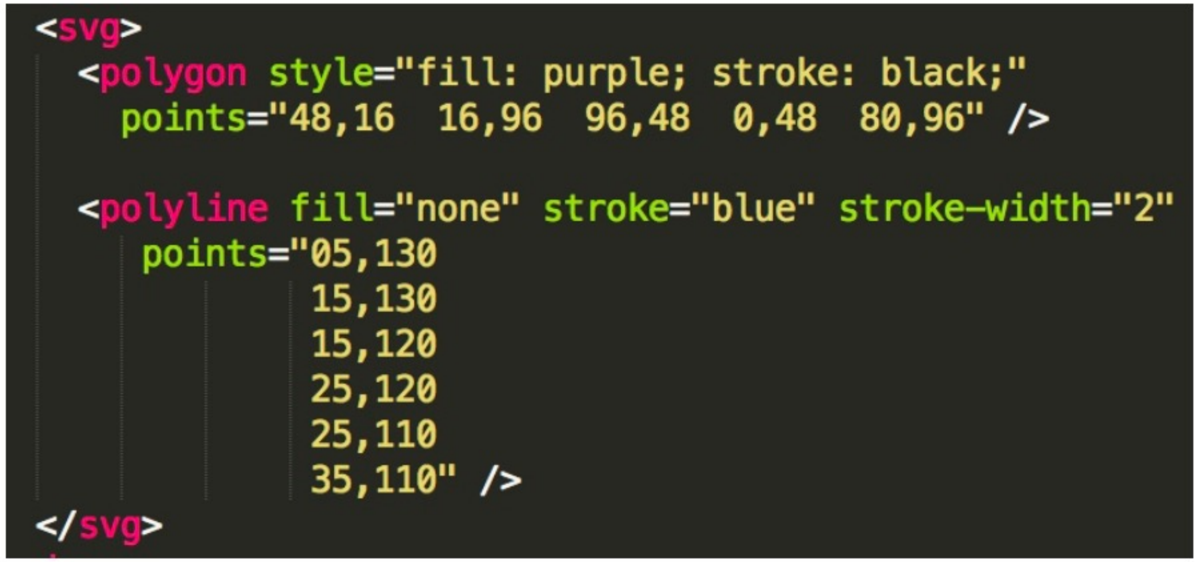


```
<style>
    rect{
      fill: green;
    }

    circle{
      stroke: orange;
      stroke-width: 3;
    }
</style>
```

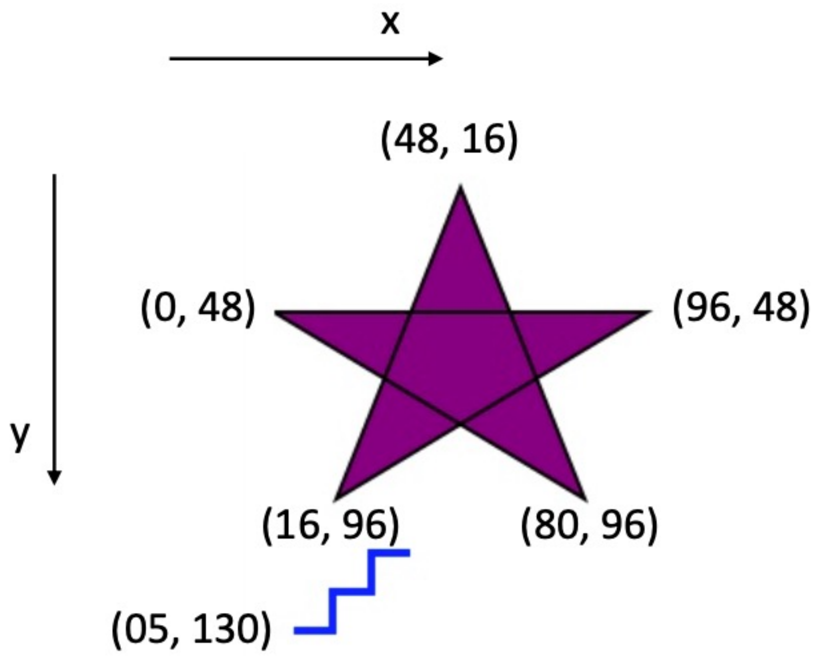


SVG - Polygon and Polyline

- Use coordinates to specify path



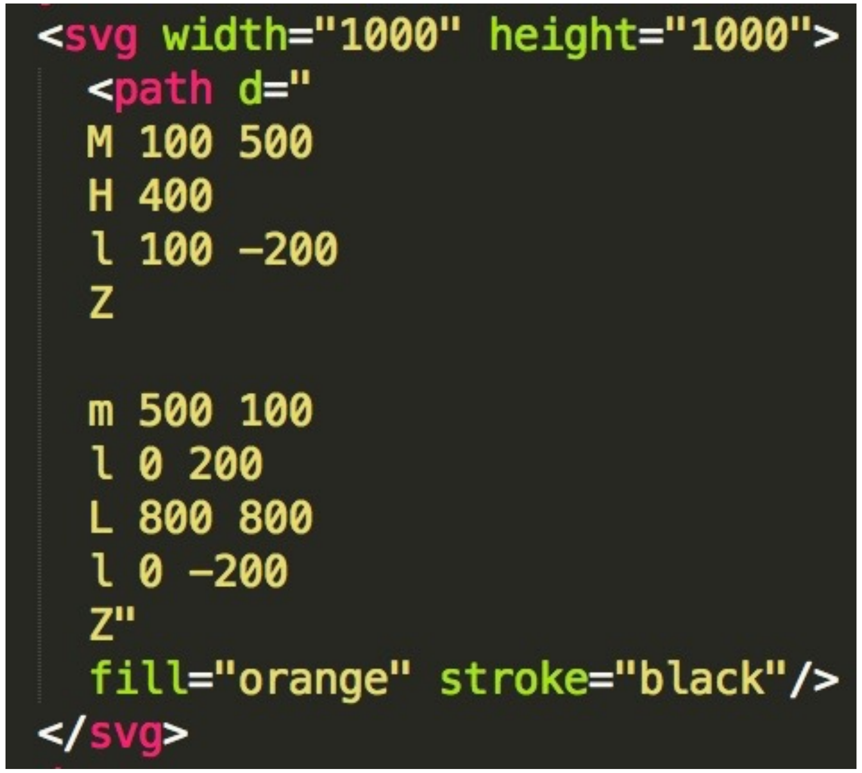
```
<svg>
  <polygon style="fill: purple; stroke: black;"
    points="48,16  16,96  96,48  0,48  80,96" />

  <polyline fill="none" stroke="blue" stroke-width="2"
     points="05,130
             15,130
             15,120
             25,120
             25,110
             35,110" />
</svg>
```



SVG - Path

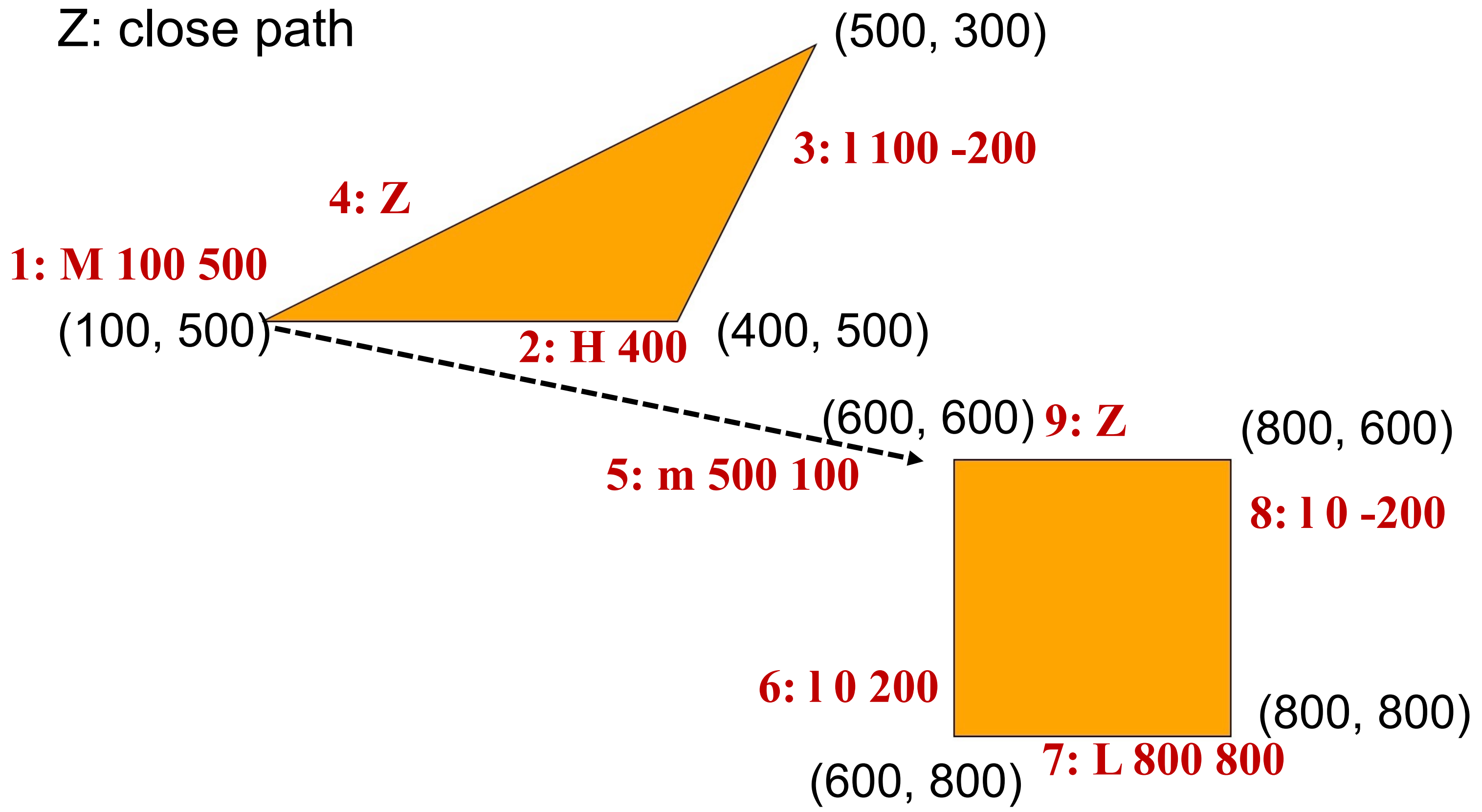
- M x y – Move to (x,y)
  - m dx dy – Move by (dx,dy)
- L x y – Line to (x,y)
  - l dx dy
- H x, V y – draw horizontal and vertical lines
  - h dx, v dy
- Z, z close path
- Curve commands (Bezier Curves and Arcs)
  - https://developer.mozilla.org/en-US/docs/Web/SVG/Tutorial/Paths?redirectlocale=en-US&redirectslug=SVG%2FTutorial%2FPaths#Curve_commands

SVG - Path



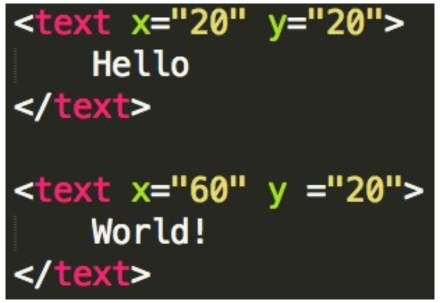
```
<svg width="1000" height="1000">
  <path d="
  M 100 500
  H 400
  l 100 -200
  Z

  m 500 100
  l 0 200
  L 800 800
  l 0 -200
  Z"
  fill="orange" stroke="black"/>
</svg>
```

M: move to
H: horizontal line
L: line to
Z: close path

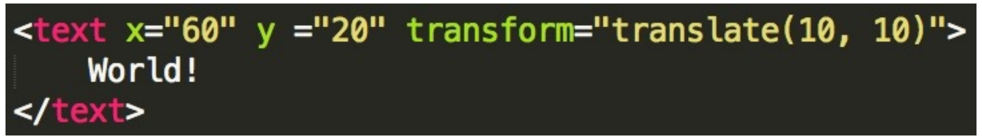


SVG - Transform

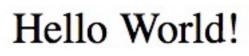
- translate(*dx, dy*)
  - move a shape by (*dx, dy*)



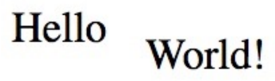
```
<text x="20" y="20">
    Hello
</text>

<text x="60" y ="20">
    World!
</text>
```



```
<text x="60" y ="20" transform="translate(10, 10)">
    World!
</text>
```



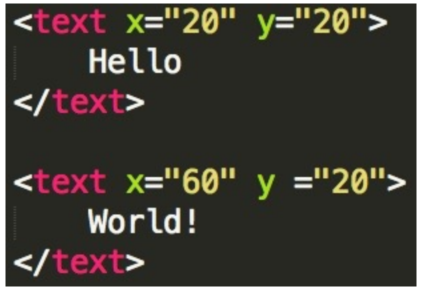
```
Hello World!
```



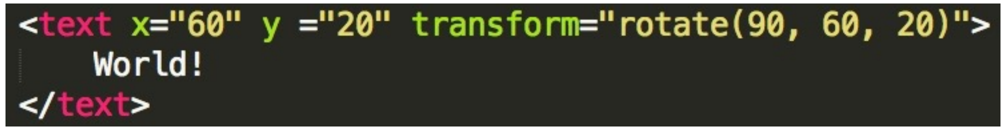
```
Hello
      World!
```

SVG - Transform

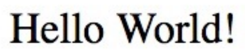
- rotate(*a, x, y*)
  - rotate a shape by *a* degrees about a given point (*x, y*)

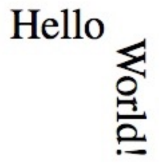


```
<text x="20" y="20">
    Hello
</text>

<text x="60" y ="20">
    World!
</text>
```



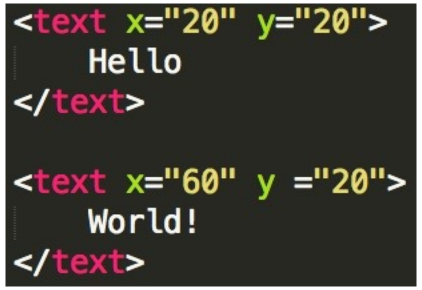
```
<text x="60" y ="20" transform="rotate(90, 60, 20)">
    World!
</text>
```



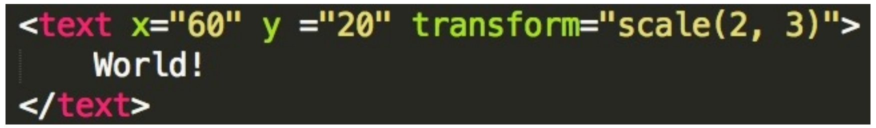


SVG - Transform

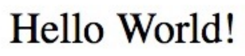
- scale(*x*, *y*)
  - scales both the shape's size and its coordinates

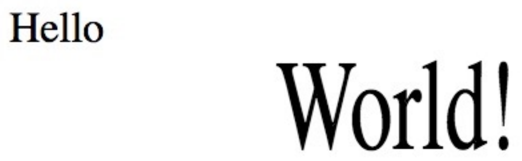


```
<text x="20" y="20">
    Hello
</text>

<text x="60" y ="20">
    World!
</text>
```



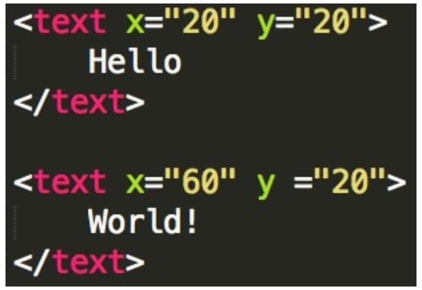
```
<text x="60" y ="20" transform="scale(2, 3)">
    World!
</text>
```



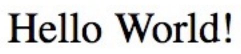


SVG - Transform

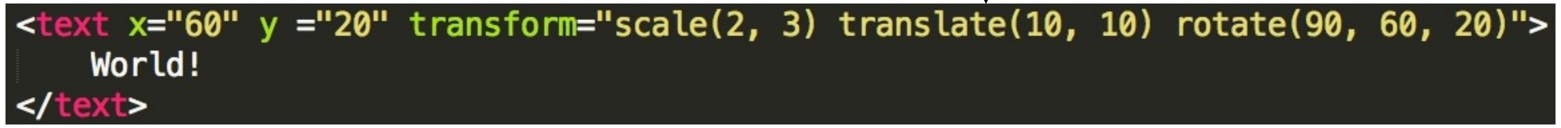
- Multiple functions



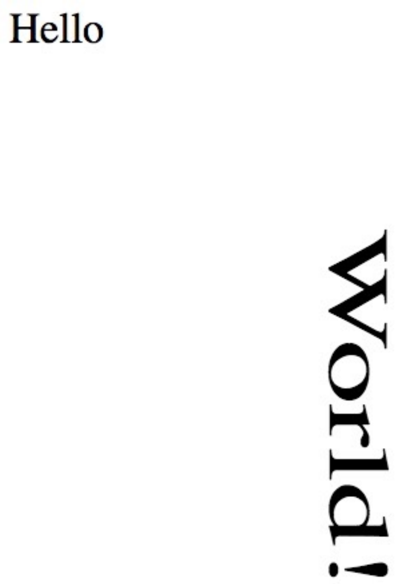
```
<text x="20" y="20">
    Hello
</text>

<text x="60" y ="20">
    World!
</text>
```



Transform in the reverse order, i.e. the order of rotate, translate, and scale

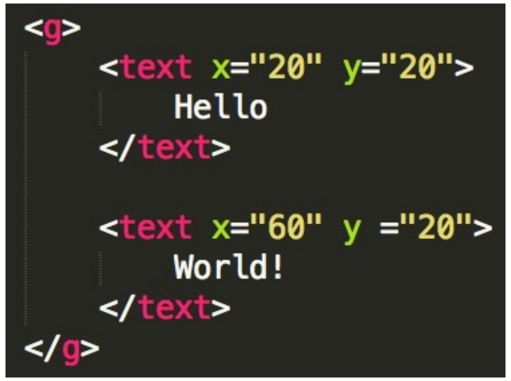


```
<text x="60" y ="20" transform="scale(2, 3) translate(10, 10) rotate(90, 60, 20)">
    World!
</text>
```

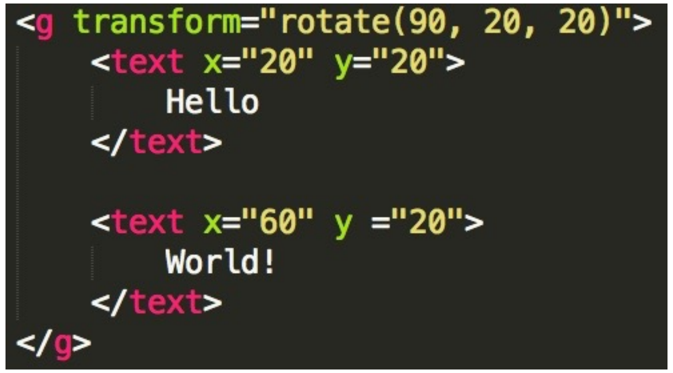


SVG - Group + Transform

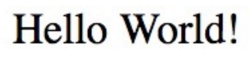
- Group multiple shapes
  - transformations applied to the <g> element are performed on its child elements
  - <g> tag

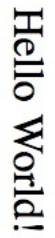


```
<g>
    <text x="20" y="20">
        Hello
    </text>

    <text x="60" y ="20">
        World!
    </text>
</g>
```



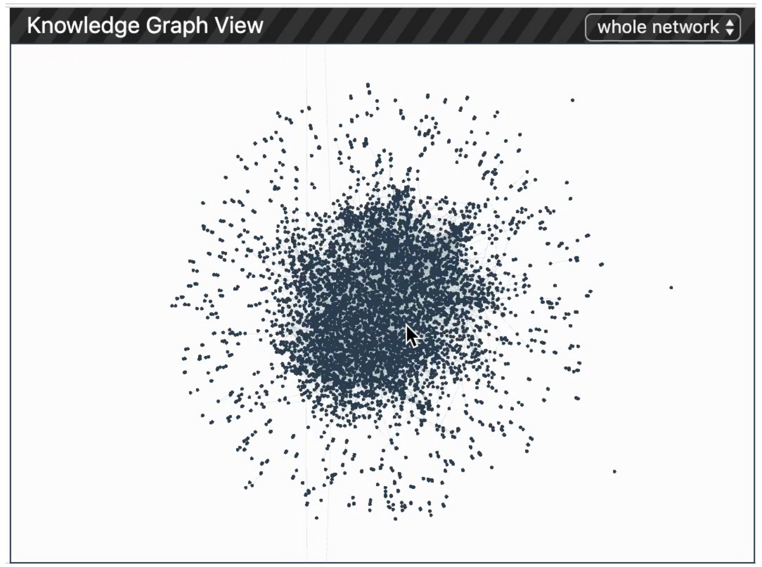
```
<g transform="rotate(90, 20, 20)">
    <text x="20" y="20">
        Hello
    </text>

    <text x="60" y ="20">
        World!
    </text>
</g>
```



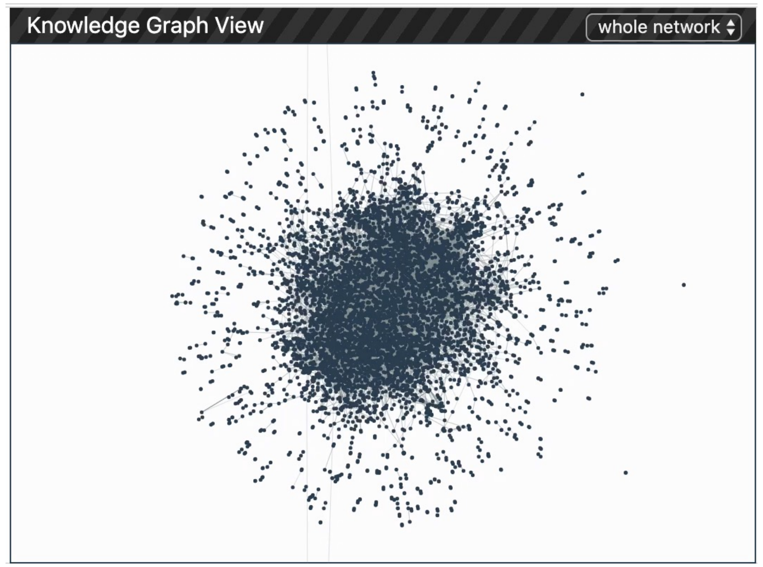


SVG performance

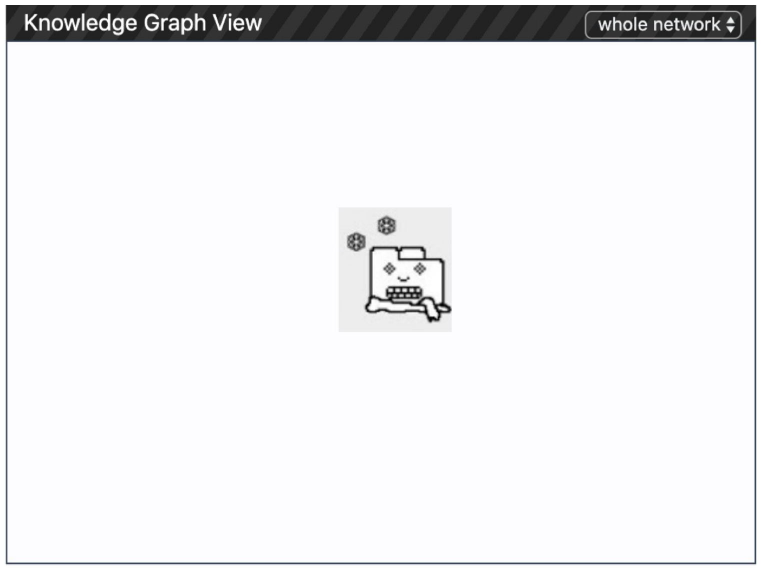
- SVG gives better performance with smaller number of objects or larger surface.



WebGL



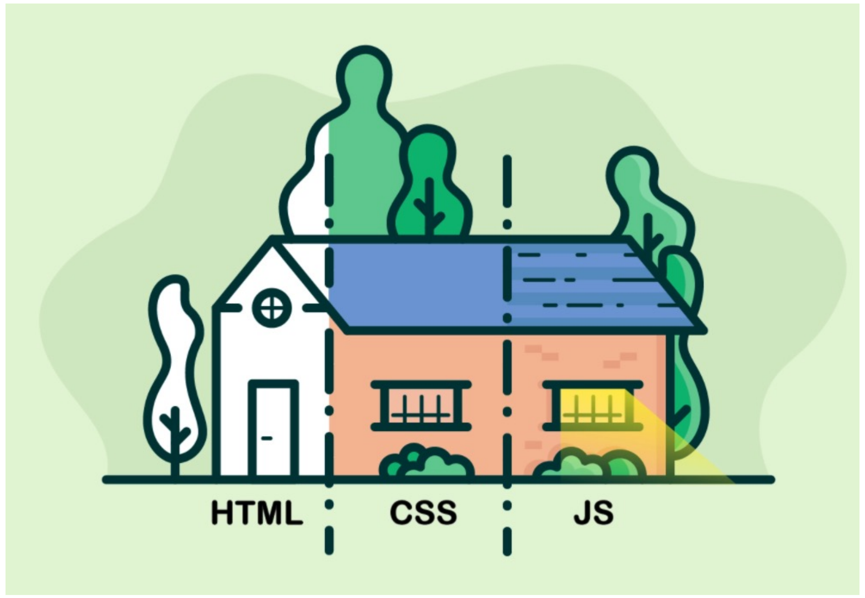
Canvas

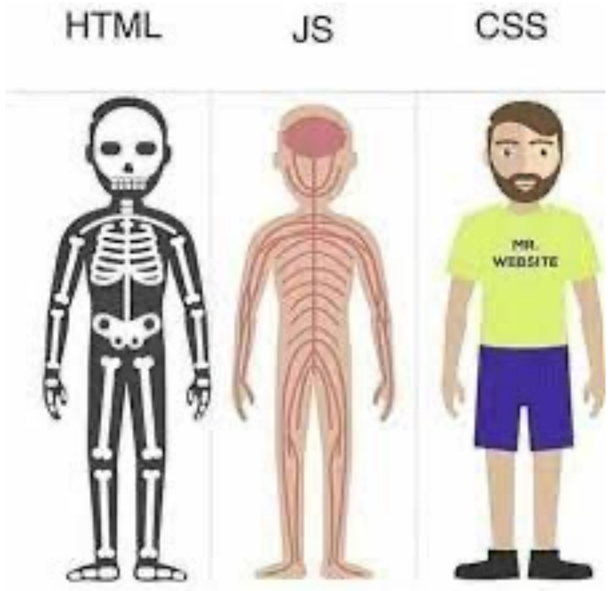


SVG

JavaScript

- *JavaScript* was initially created to “make web pages alive”.
  - HTML to define the content of web pages
  - CSS to specify the appearance of web pages
  - JavaScript to program the behavior of web pages

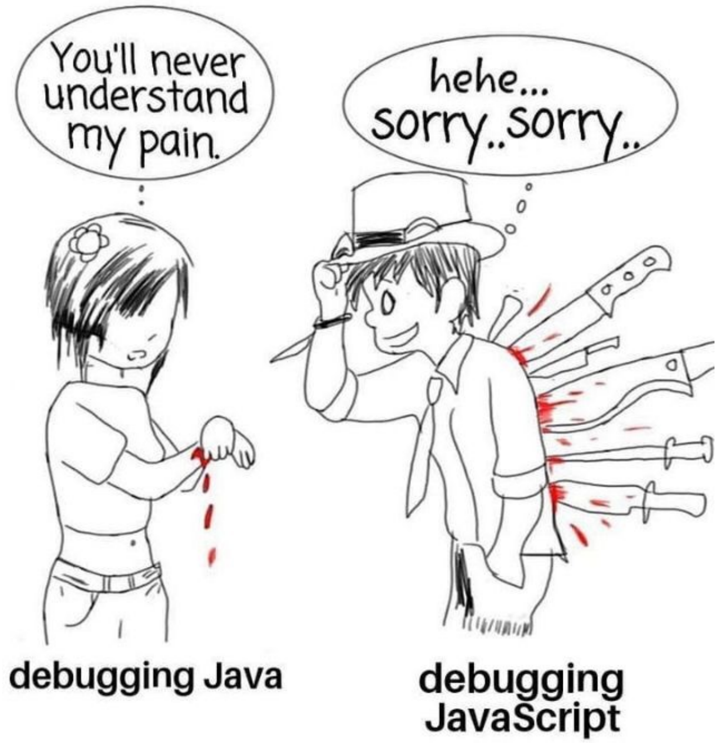




credit by https://www.google.com/url?sa=i&url=https%3A%2F%2Fdribbble.com%2Fshots%2F5299243-What-is-Front-end-Development&psig=AOvVaw0ubEorLkexK84YXVhnGonk&ust=1645589013701000&source=images&cd=vfe&ved=0CAwQjhxqFwoTCPi2p_y2kvYCFQAAAAAdAAAAABAU

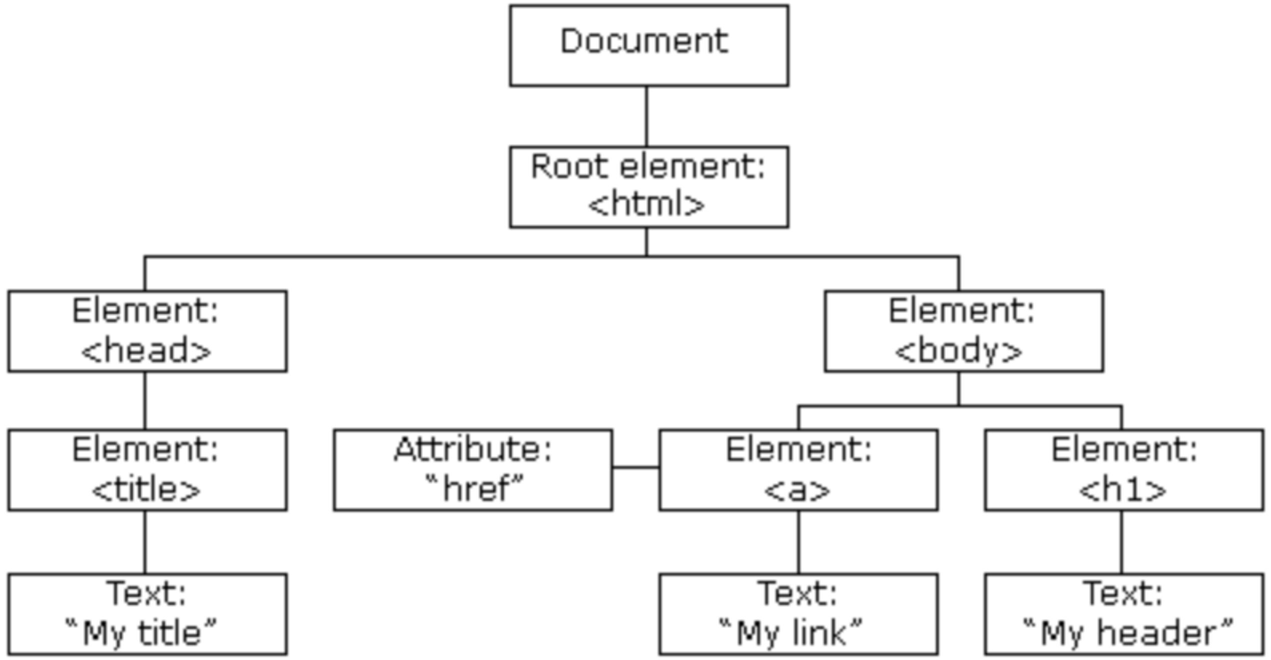
JavaScript

- JavaScript is the programming language with C/C++ style syntax
- *for, while, continue, break, if/else, switch* are similar to C/C++
- *operators (+,-,*,/,%)* are also similar (except ==,!=,||)
- weak typed language (similar to python)



JavaScript HTML DOM

- When a web page is loaded, the browser creates a **D**ocument **O**bject **M**odel of the page.
- The **HTML DOM** model is constructed as a tree of **Objects**:



JavaScript HTML DOM

The **HTML DOM** is an **API** (Programming Interface) for **JavaScript**:

- JavaScript can add/change/remove HTML elements
- JavaScript can add/change/remove HTML attributes
- JavaScript can add/change/remove CSS styles
- JavaScript can react to HTML events
- JavaScript can add/change/remove HTML events

We will learn how to use D3.js to manipulate DOM in a simple way

How to use Javascript

- Internal JS: using <script> tags within a single document
- External JS: linking an external .js file
    - <script type="text/javascript" src="myscripts.js"></script>

Demo

Developer tools - Console

- You can type JavaScript code directly into your browser in a web page
- The console accepts one line of code at a time
- Open Console
  - Chrome
    - Select View -> Developer -> JavaScript Console
- Safari
  - Safari -> Preferences -> Advanced -> Show Develop menu in menu bar
    - Develop -> Show JavaScript Console

Developer tools – Debug your code

- Open Console
    - Chrome
        - Select View -> Developer -> JavaScript Console