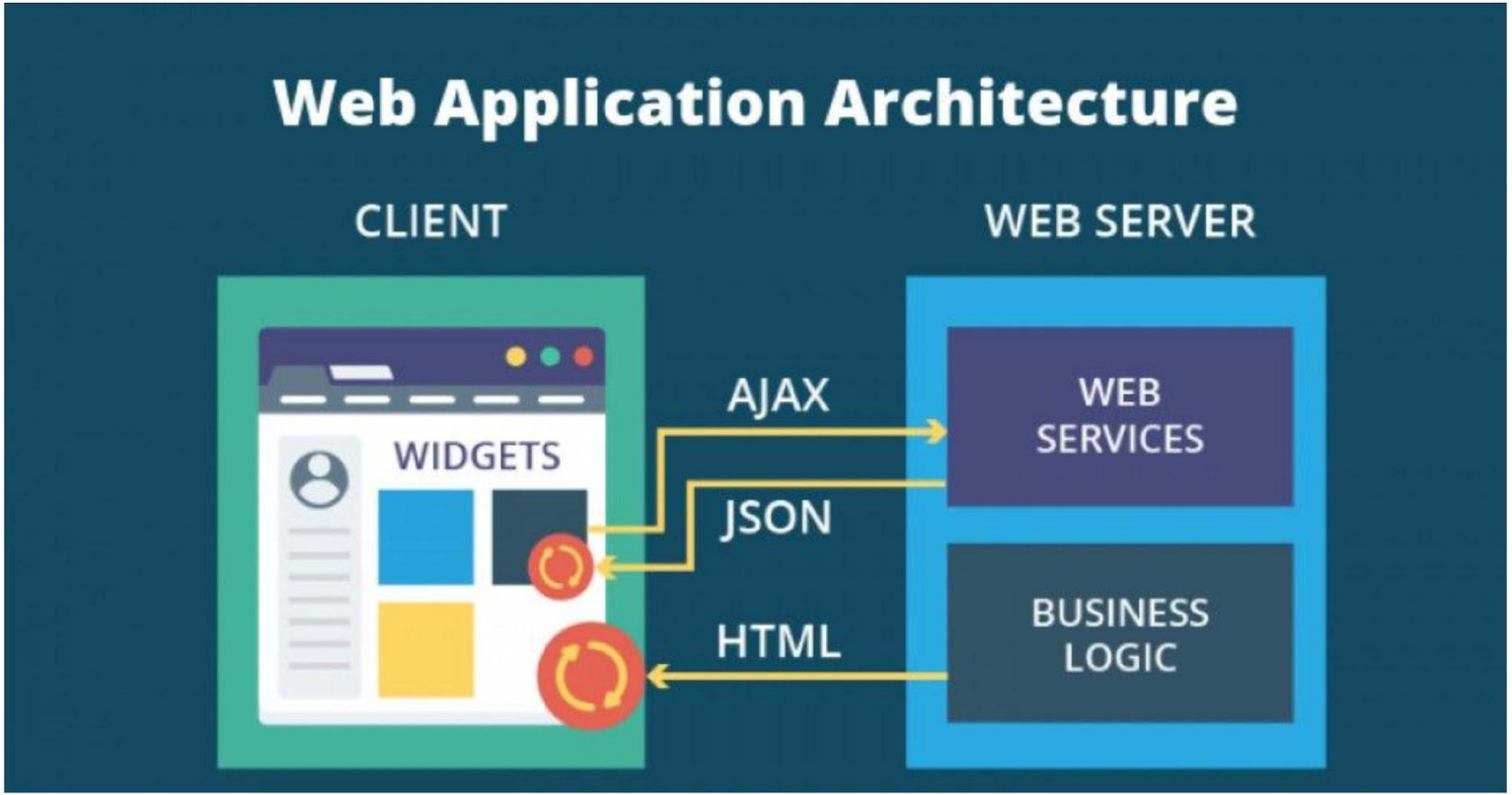
EECS E6893 Big Data Analytics
HW3: Data visualization

Srividya Inampudi, si2396@columbia.edu

Agenda

- Introduction of Web Application
  - HTML, CSS and JavaScript
  - 2 important things to know: SVG and DOM
- Using D3.js to do data visualization
- Introduction of Apache HTTP Server

Web Application



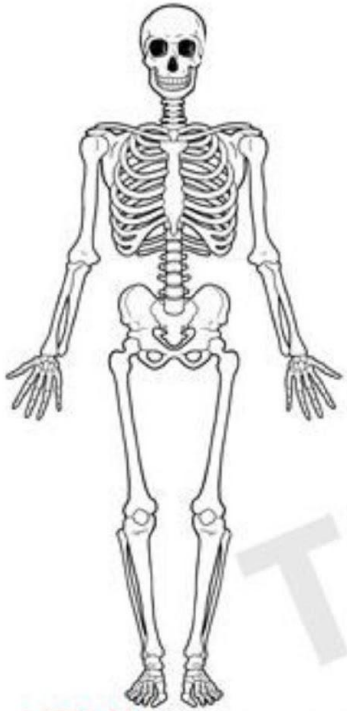
Reference: https://hackr.io/blog/web-application-architecture-definition-models-types-and-more

Client-side of Web

- HTML, CSS and JS are the parts of all websites that users directly interact with.
- HTML provides the *basic structure* of sites, which is enhanced and modified by other technologies like CSS and JavaScript.
- CSS is used to control *presentation, formatting, and layout*.
- JavaScript is used to control the *behavior* of different elements.

Reference: https://tutorial.techaltum.com/webdesigning.html

WEB DESIGNING



HTML (Structure)

CSS (Presentation)

Javascript (functionality)

tutorial.techaltum.com

Reference: https://tutorial.techaltum.com/webdesigning.html

What and Why?

- JavaScript is a programming language
- used by Web browsers to create a dynamic and interactive experience for the user.
- Most of the functions and applications that make the Internet indispensable to modern life are coded in some form of JavaScript.
- Some of the dynamic website enhancements performed by JavaScript are: Loading new content or data onto the page without reloading the page, Rollover effects and dropdown menus etc.
- Some of its most powerful features involve asynchronous interaction with a remote server.

Common Uses of JavaScript

- Form validation
- Page embellishments and special effects
- Navigation systems
- Basic math calculations
- Dynamic content manipulation
- Sample applications
    - Dashboard widgets in Mac OS X, Google Maps, Philips universal remotes, Writely word processor, hundreds of others…

JavaScript in Web Pages

- Embedded in HTML page as <script> element
  - JavaScript written directly inside <script> element
    - <script> alert("Hello World!") </script>
  - Linked file as src attribute of the <script> element

    <script type="text/JavaScript" src="functions.js"></script>
- Event handler attribute

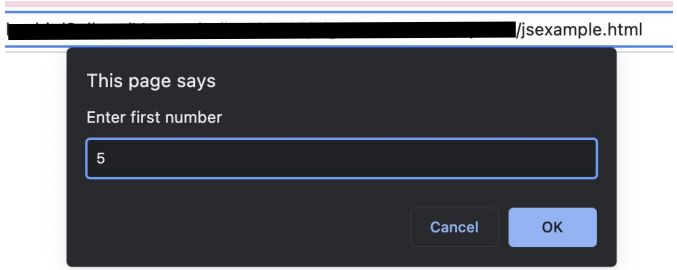
  <a href="http://www.yahoo.com " onmouseover="alert('hi');">
- Pseudo-URL referenced by a link

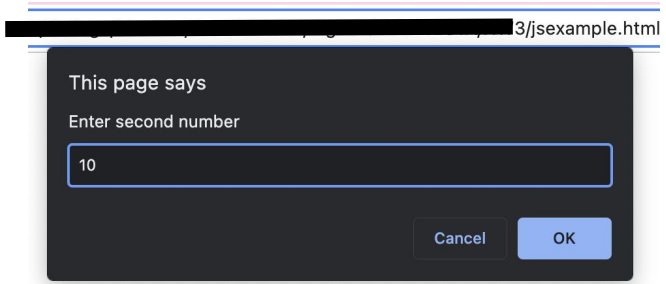
  Click me

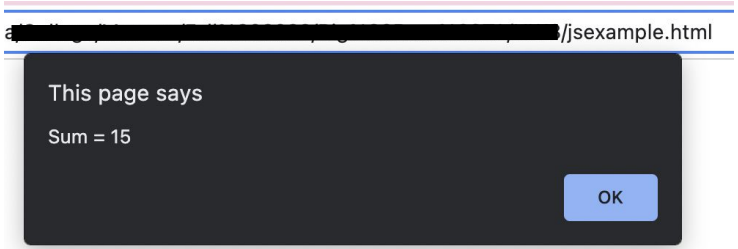
Example 1: Add Two Numbers

jsexample.html > ...

```
1   <!DOCTYPE html>
2   <html>
3       <head>
4           <title>Adding two numbers</title>
5       </head>
6   <body>
7       <script>
8           var num1, num2, sum
9           num1 = prompt("Enter first number")
10          num2 = prompt("Enter second number")
11          sum = parseInt(num1) + parseInt(num2)
12          alert("Sum = " + sum)
13      </script>
14
15  </body>
16  </html>
17
```



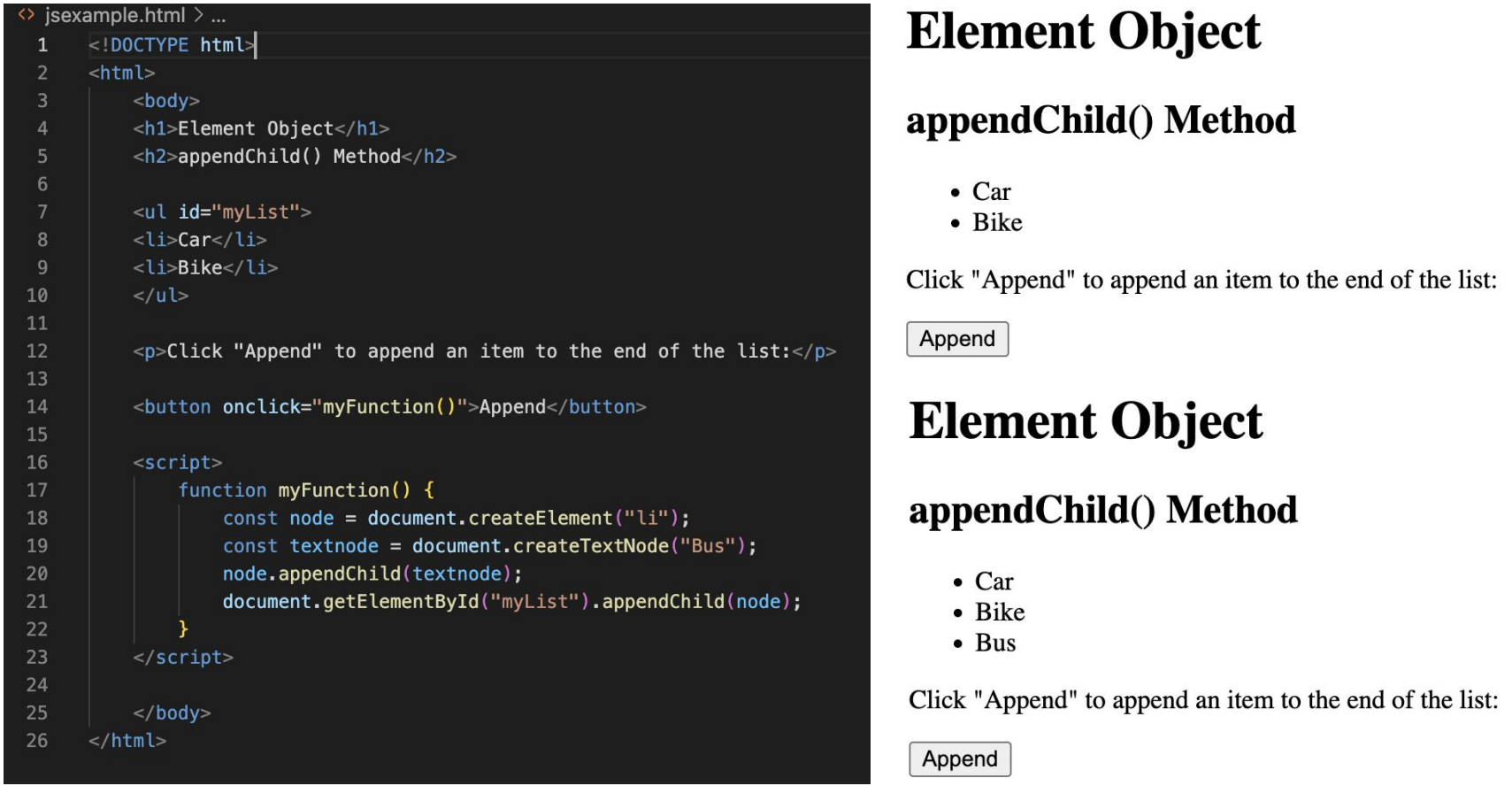




Example 2: Page Manipulation

jsexample.html > ...

```
1   <!DOCTYPE html>
2   <html>
3       <body>
4       <h1>Element Object</h1>
5       <h2>appendChild() Method</h2>
6
7       <ul id="myList">
8       <li>Car</li>
9       <li>Bike</li>
10      </ul>
11
12      <p>Click "Append" to append an item to the end of the list:</p>
13
14      <button onclick="myFunction()">Append</button>
15
16      <script>
17          function myFunction() {
18              const node = document.createElement("li");
19              const textnode = document.createTextNode("Bus");
20              node.appendChild(textnode);
21              document.getElementById("myList").appendChild(node);
22          }
23      </script>
24
25      </body>
26  </html>
```



Language Basics

- JavaScript is case sensitive
  - onClick, ONCLICK, … are HTML, thus not case-sensitive
- Statements terminated by returns or semi-colons
  - x = x+1; same as x = x+1
- “Blocks” of statements enclosed in { …}
- Variables
  - Define using the var statement
  - Define implicitly by its first use, which must be an assignment
    - Implicit definition has global scope, even if occurs in nested scope!

JavaScript Primitive Data types

- Boolean: true and false
- Number: 64-bit floating point
  - Similar to Java double and Double
  - No integer type
  - Special values NaN (not a number) and Infinity
- String: sequence of zero or more Unicode chars
  - No separate character type (just strings of length 1)
  - Literal strings using ' or " characters (must match)
- Special objects: null and undefined

Objects

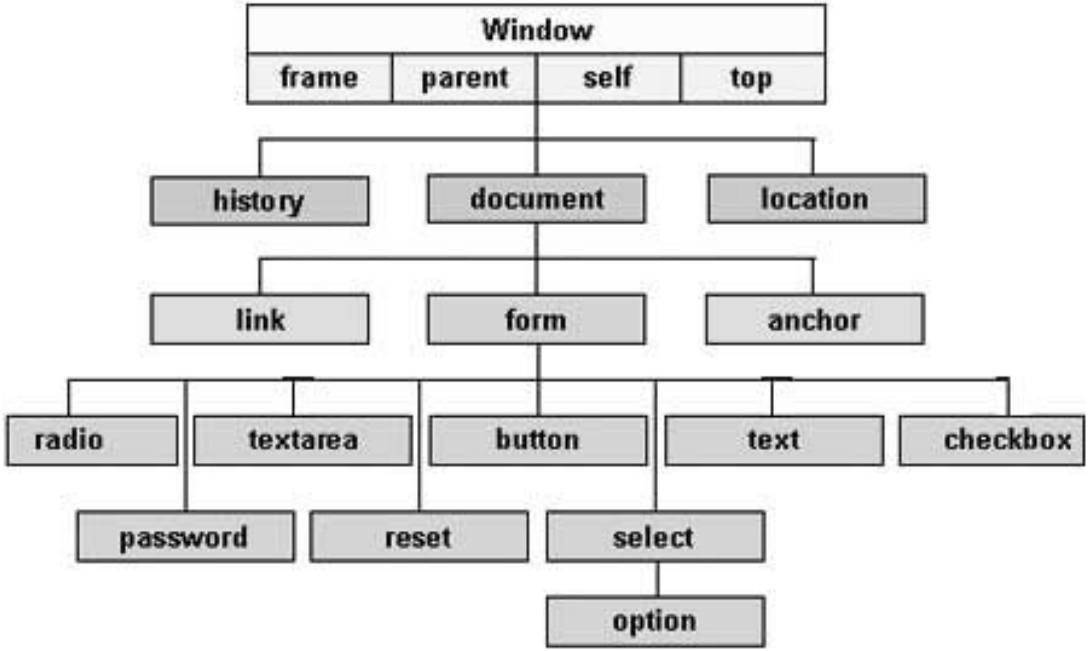
- An object is a collection of named properties
- Think of it as an associative array or hash table
  - Set of name:value pairs
    - objBob = {name: “Bob", grade: 'A', level: 3};
  - Play a role similar to lists in Lisp / Scheme
- New members can be added at any time
    - objBob.fullname = 'Robert';
- Can have methods

Functions

- Functions are objects with method called “( )”
  - A property of an object may be a function (=method)
    - function max(x,y) { if (x>y) return x; else return y;};
    - max.description = “return the maximum of two arguments”;
  - Local declarations may appear in function body
- Call can supply any number of arguments
  - functionname.length : # of arguments in definition
  - functionname.arguments.length : # arguments in call
  - Basic types are passed by value, objects by reference
- “Anonymous” functions
  - (function (x,y) {return x+y}) (2,3);

Document Object Model (DOM)

- HTML page is structured data
- DOM provides representation of this hierarchy
- Examples
  - Properties: document.alinkColor, document.URL, document.forms[ ], document.links[ ], document.anchors[ ], …
  - Methods: document.write(document.referrer)
    - These change the content of the page!
- Also Browser Object Model (BOM)
  - Window, Document, Frames[], History, Location, Navigator (type and version of browser)



JavaScript Document Object Model (DOM) hierarchy

Ref: https://www.tutorialspoint.com/javascript/javascript_html_dom.htm

Document Object Model (DOM)

The way a document content is accessed and modified is called the Document Object Model, or DOM. The Objects are organized in a hierarchy. This hierarchical structure applies to the organization of objects in a Web document.

- Window object − Top of the hierarchy. It is the outmost element of the object hierarchy.
- Document object − Each HTML document that gets loaded into a window becomes a document object. The document contains the contents of the page.
- Form object − Everything enclosed in the <form>...</form> tags sets the form object.
- Form control elements − The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.

Introduction to HTML

- HTML is a language for describing web pages.
- HTML stands for **H**yper **T**ext **M**arkup **L**anguage
- HTML is not a programming language, it is a **markup language**
- A markup language is a set of **markup tags**
- HTML uses **markup tags** to describe web pages

Objectives of HTML

- create, save and view a HTML document
- format a web page using section heading tags
- describe Ordered and Unordered lists
- explain graphics in HTML document
- describe hypertext links and making text/image link

World Wide Web

- The **World Wide Web** (abbreviated as **WWW** or **W3** and commonly known as **the Web)**is a system of interlinked hypertext documents accessed via the Internet. With a web browser, one can view web pages that may contain text, images, videos, and other multimedia and navigate between them via hyperlinks.

HTML Tools

a) HTML Editor: it is the program that one uses to create and save HTML documents. They fall into two categories:

- Text based or code based which allows one to see the HTML code as one is creating a document.e.g. Notepad.
- Netscape composer

b) Web Browser: program to view and test the HTML documents. They translate Html encoded files into text, image, sounds and other features user see. Microsoft Internet Explorer, Netscape, Chrome are examples of browsers that enables user to view text and images and many more other World Wide Web features.

HTML Terminology

- Tag: Tags are always written within angles brackets. it is a piece of text is used to identify an element so that the browser realizes how to display its contents.e.g.<HTML> tag indicates the start of an HTML document . HTML tag can be two types. They are:-
  - Paired Tags :A tag is said to be a paired tag if text is placed between a tag and its companions tag.In paired tag ,the first tag is referred to as opening tag and the second tag is referred to as closing tag.
  - Unpaired Tags: An unpaired tag does not have a companion tag .unpaired tag also known as singular or Stand-Alone tags.e.g:
,<hr> etc.

HTML Terminology

- Attribute: Attribute is the property of an tag that specified in the opening angle brackets. It supplies additional information like color,size,home font-style etc to the browser about a tag. E.g. most of the common attributes are height, color,width,src,border,align etc.

- DTD: Document Type Definition is a collection of rules written in standard Generalized Markup Language(SGML).HTML is define in terms of its DTDS. All the details of HTML tags, entities and related document structure are defined in the DTDS.

- ELEMENT: Element is the component of a document's structure such as a title, a paragraph or a list. It can include an opening and a closing tag and the contents within it.

Steps to create a HTML file and view in browser

- Step-1: Open a text editor or notepad on your machine.
- Step-2: Enter the following lines of code:

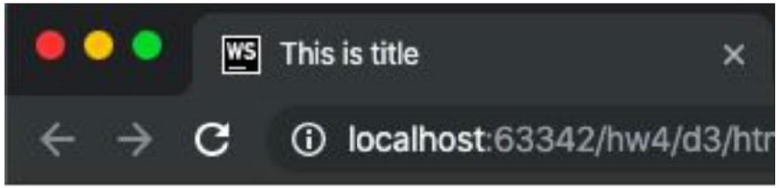
myfirstpage.html > html

```
1   <!DOCTYPE html>
2   <html>
3
4       <head>
5           <title>Page Title</title>
6       </head>
7
8       <body>
9           <h1>This is a Heading</h1>
10          <p>This is a paragraph.</p>
11      </body>
12  </html>
```

- Step-3: Save the file as myfirstpage.html (go to File-Save As give File name: myfirstpage.html-choose save as type: All Files-click save)
- Step-4: Viewing document in web browser (go to folder where file is saved and open it in any browser of your choice)

HTML

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>This is title</title>
</head>
<body>

<h1>Header 1</h1>
<h2>Header 2</h2>
<h3>Header 3</h3>


<p>Paragraph 1</p>

</body>
</html>
```



Header 1

Header 2

Header 3

Paragraph 1

Styles

Rough timeline of HTML / CSS history:

Early 90s:

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

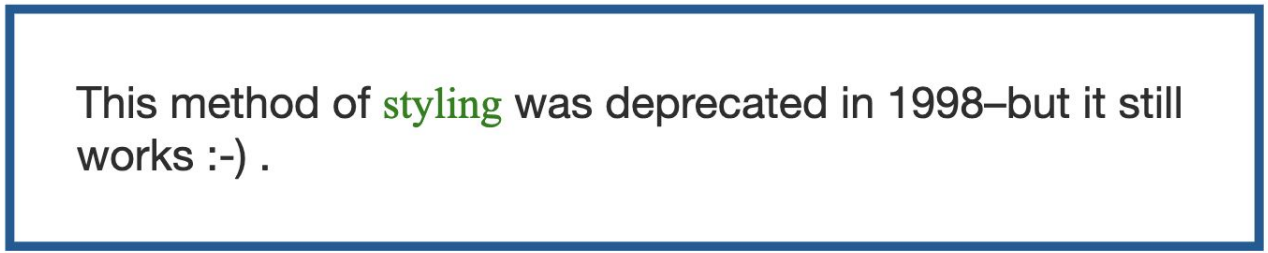
This is an h1 header.

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

Styles

Mid 1990s

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



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: 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: External style sheet

Preferred method of adding styles

Body of html file:

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

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

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>
```

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

SVG in HTML

- SVG stands for Scalable Vector Graphics. It is used to define vector-based graphics for the Web
- **Every element and every attribute in SVG files can be animated**
- SVG integrates with other W3C standards such as the **DOM** and XSL

```
<p id="p1">Paragraph 1</p>
<button type="button"onclick=myfunction()>Click here to change Paragraph 1</button>

<svg id='svg1' width="400" height="200">
    <rect id='r1'width="300" height="100" fill="red"/>
</svg>
```

SVG in HTML

Header 1

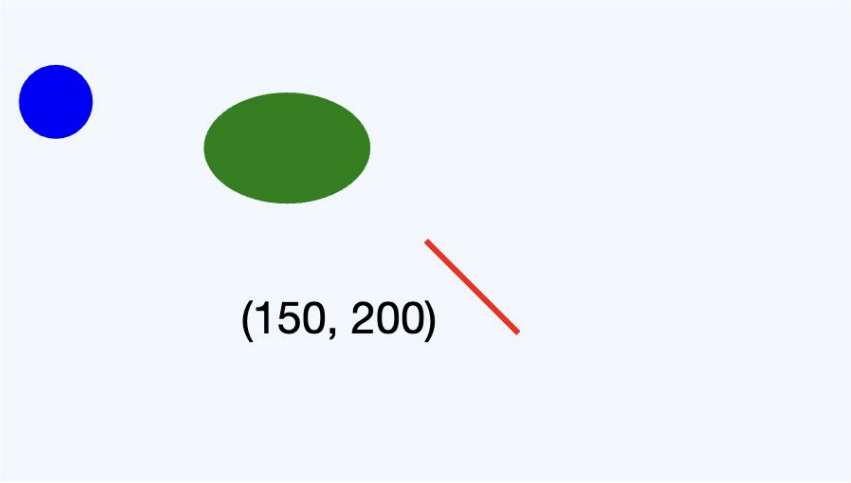
Header 2

Header 3

Paragraph 1

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>
```



What if we create a SVG elements and use DOM in Javascript to access its attributes?

```
var r1 = document.getElementById('r1');
r1.setAttribute('fill', 'blue');
```

If we draw a series of SVG and texts based on data, and use DOM to control their attributes, then we get a simple charts!

- D3.js, a library to do this in a simple way

What if we create a SVG elements and use DOM in Javascript to access its attributes?

Header 1

Header 2

Header 3

Paragraph 1

If we draw a series of SVG and texts based on data, and use DOM to control their attributes, then we get a simple charts!

- D3.js, a library to do this in a simple way

D3.js

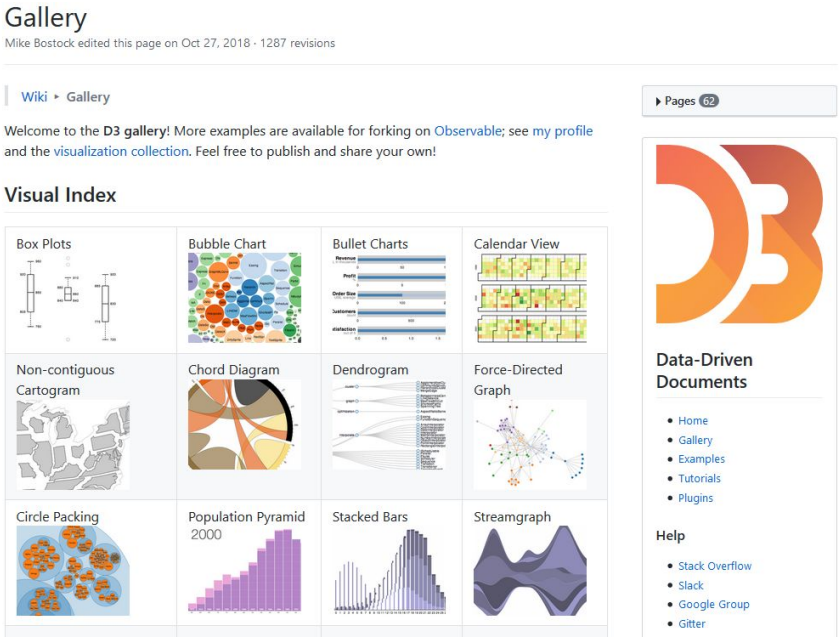
D3.js is a JavaScript library for manipulating documents based on data. It helps you bring data to life using HTML, SVG, and CSS. It provides a data-driven approach to DOM manipulation.

Visit https://d3js.org for more tutorials!

What is D3?

- **D3.js** is a JavaScript library for manipulating documents based on data. (https://d3js.org/)
- At its core, D3 is a graphics library for the web
- D3 is a targeted library – data visualization
- D3 visualizations can be embedded into any web page
- D3 Links:
  - Homepage: https://d3js.org/
  - Github: https://github.com/d3/d3
  - Latest Online Release: https://d3js.org/d3.v5.min.js

D3 Live Examples



- D3 Examples page: https://github.com/d3/d3/wiki/Gallery
- Can there be more content on web page than just a D3 visualization?
- Does D3 support interaction?
- Can there be multiple D3 visualizations on one web page?

Running D3

- What does D3 require to run?
  - A web browser
  - D3 source code
  - Valid HTML document
  - Server
- Most people probably have a web browser
- D3 source code can be added to any HTML file by including:
  - <script src="https://d3js.org/d3.v5.min.js"></script>
- Server:
  - Can use a remote setup
  - Host a local server

Starting D3

- “src” in script tag
- Sample code to get started. Save file as d3basic.html

d3basic.html > html

```
1   <!DOCTYPE html>
2   <html>
3
4       <head>
5           <script src="https://d3js.org/d3.v4.js"></script>
6       </head>
7
8       <body>
9           <p>Hello</p>
10      </body>
11  </html>
```

d3basic.html

- If we want our server to display that page in our browser we can go to: localhost:8000/d3basic.html
- We could also change d3basic.html to be called index.html, and our localhost:8000 will default to that page
- Key components of this file:
  - Valid HTML
  - Included script tag for D3 source

Example: Simple Bar Chart

```
<svg id="svg2"></svg>>
<script src="https://d3js.org/d3.v4.min.js"></script>
<script>
    var data = [10,20,30,40,50];
    var svgWidth = 640, svgHeight = 320;

    var svg = d3.select('svg')
            .attr("width", svgWidth)
            .attr("height", svgHeight);

    var barChart = svg.selectAll("rect")
            .data(data)
            .enter()
            .append("rect")
            .attr("class", "bar")
            .attr('x', 20)
            .attr('y', function(d,i){return i*30+100})
            .attr('height', 25)
            .attr('width', function(d){return d});
</script>>
```

Header 1

Header 2

Header 3

Paragraph 1

```
var data = [10,20,30,40,50];
var svgWidth = 640, svgHeight = 320;
```

Declaration of data and variables

```
var svg = d3.select('svg')
        .attr("width", svgWidth)
        .attr("height", svgHeight);
```

D3 provides operating on arbitrary sets of nodes called *selections.* You can manipulate individual nodes and set the attributes

```
var barChart = svg.selectAll("rect")
        .data(data)
        .enter()
        .append("rect")
        .attr("class", "bar")
        .attr('x', 20)
        .attr('y', function(d,i){return i*30+100})
        .attr('height', 25)
        .attr('width', function(d){return d});
cript>>
```

Tricky part of D3: Once you bound data with selection,each element in the data array is paired with the corresponding node in the selection. If there are fewer nodes than data, you can use enter() to appending nodes.

Again, please visit https://d3js.org for more tutorials!

Histogram using D3

```
<script>
    // setting the dimensions and margins of the graph
    var margin = {top: 10, right: 30, bottom: 30, left: 40},
        width = 460 - margin.left - margin.right,
        height = 400 - margin.top - margin.bottom;

    // appending the svg object to the body of the page
    var svg = d3.select("#histogram")
      .append("svg")
        .attr("width", width + margin.left + margin.right)
        .attr("height", height + margin.top + margin.bottom)
      .append("g")
        .attr("transform",
              "translate(" + margin.left + "," + margin.top + ")");

    // getting the data in csv format
    d3.csv("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/1_OneNum.csv", function(data) {

      // X axis: scale and draw:
      var x = d3.scaleLinear()
          .domain( [0,1000])
         // .domain( [0, d3.max(data, function(d) { return +d.price })])
         // this will set the scale of x from 0 to max of the price column
          .range([0, width]);
      svg.append("g")
          .attr("transform", "translate(0," + height + ")")
          .call(d3.axisBottom(x));

      // setting the parameters for the histogram
      var histogram = d3.histogram()
          .value(function(d) { return d.price; })   // giving a vector of value
          .domain(x.domain())  // then setting the domain of the graphic
          .thresholds(x.ticks(30)); // then the numbers of bins
```

Histogram using D3

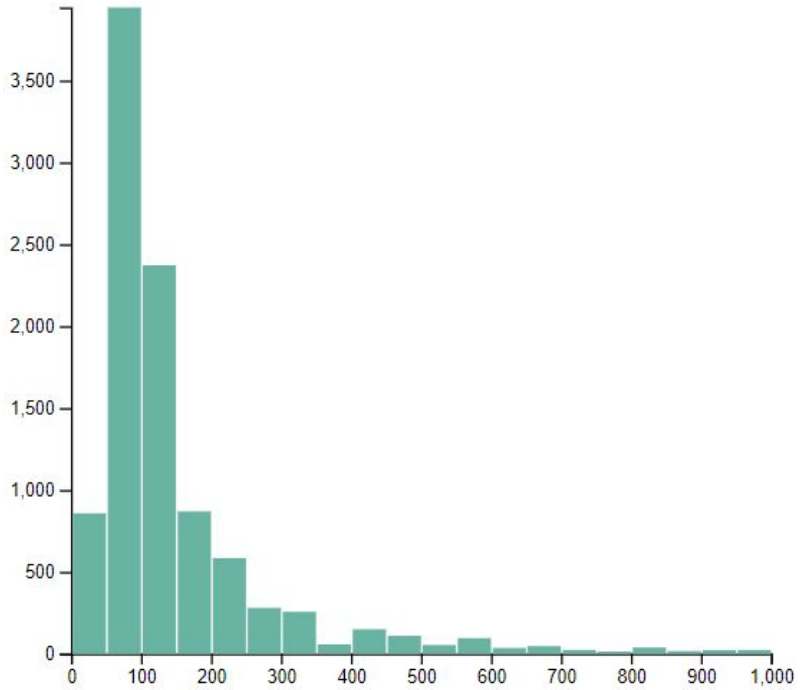
```
  // And apply this function to data to get the bins
  var bins = histogram(data);

  // Y axis: scale and draw:
  var y = d3.scaleLinear()
      .range([height, 0]);
      y.domain([0, d3.max(bins, function(d) { return d.length; })]);
  svg.append("g")
      .call(d3.axisLeft(y));

  // append the bar rectangles to the svg element
  svg.selectAll("rect")
      .data(bins)
      .enter()
      .append("rect")
        .attr("x", 1)
        .attr("transform", function(d) { return "translate(" + x(d.x0) + "," + y(d.length) + ")"; })
        .attr("width", function(d) { return x(d.x1) - x(d.x0) -1 ; })
        .attr("height", function(d) { return height - y(d.length); })
        .style("fill", "#69b3a2")

});
</script>
```

Histogram using D3

Output ->

Dataset ->

```
price
75.0
104.0
369.0
300.0
92.0
64.0
265.0
35.0
287.0
69.0
52.0
23.0
287.0
87.0
114.0
114.0
98.0
137.0
87.0
90.0
63.0
69.0
80.0
113.0
58.0
115.0
30.0
35.0
92.0
460.0
74.0
72.0
63.0
```



Line graph using D3

```
<script>
// setting the dimensions and margins of the graph
var margin = {top: 10, right: 30, bottom: 30, left: 60},
    width = 460 - margin.left - margin.right,
    height = 400 - margin.top - margin.bottom;

// appending the svg object to the body of the page
var svg = d3.select("#my_dataviz")
  .append("svg")
    .attr("width", width + margin.left + margin.right)
    .attr("height", height + margin.top + margin.bottom)
  .append("g")
    .attr("transform",
          "translate(" + margin.left + "," + margin.top + ")");

// getting the data in csv format
d3.csv("https://raw.githubusercontent.com/holtzy/data_to_viz/master/Example_dataset/3_TwoNumOrdered_comma.csv",

  // When reading the csv we will formate the date variable
  function(d){
    return { date : d3.timeParse("%Y-%m-%d")(d.date), value : d.value }
  },
```

Line graph using D3

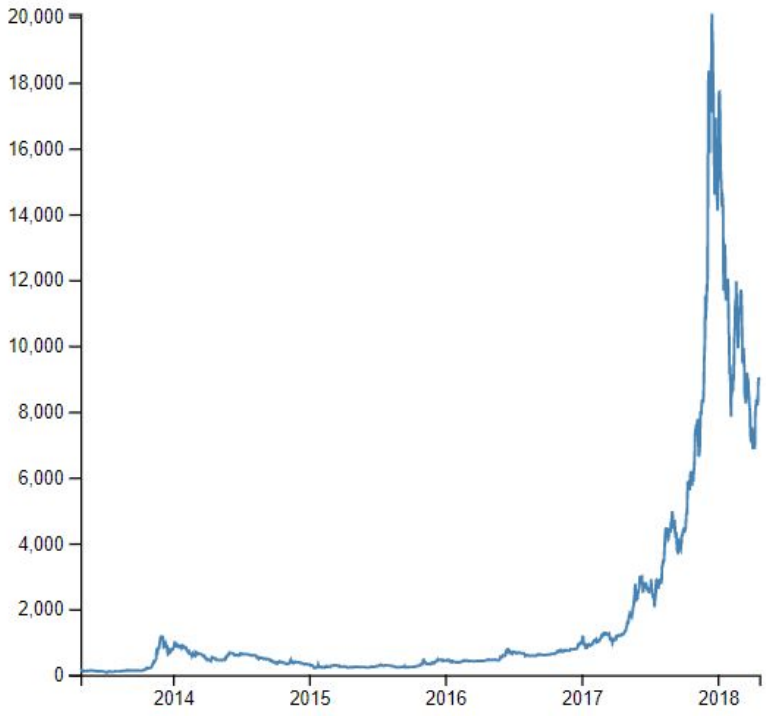
```
function(data) {
  // Setting the X axis
  var x = d3.scaleTime()
    .domain(d3.extent(data, function(d) { return d.date; }))
    .range([ 0, width ]);
  svg.append("g")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x));

  // Setting the Y axis
  var y = d3.scaleLinear()
    .domain([0, d3.max(data, function(d) { return +d.value; })])
    .range([ height, 0 ]);
  svg.append("g")
    .call(d3.axisLeft(y));

  // Adding the line
  svg.append("path")
    .datum(data)
    .attr("fill", "none")
    .attr("stroke", "steelblue")
    .attr("stroke-width", 1.5)
    .attr("d", d3.line()
      .x(function(d) { return x(d.date) })
      .y(function(d) { return y(d.value) })
      )
})
</script>
```

Line graph using D3

Dataset ->

```
date,value
2013-04-28,135.98
2013-04-29,147.49
2013-04-30,146.93
2013-05-01,139.89
2013-05-02,125.6
2013-05-03,108.13
2013-05-04,115
2013-05-05,118.8
2013-05-06,124.66
2013-05-07,113.44
2013-05-08,115.78
2013-05-09,113.46
2013-05-10,122
2013-05-11,118.68
2013-05-12,117.45
2013-05-13,118.7
2013-05-14,119.8
2013-05-15,115.81
2013-05-16,118.76
2013-05-17,125.3
2013-05-18,125.25
2013-05-19,124.5
2013-05-20,123.62
```

Output ->



Pie chart using D3

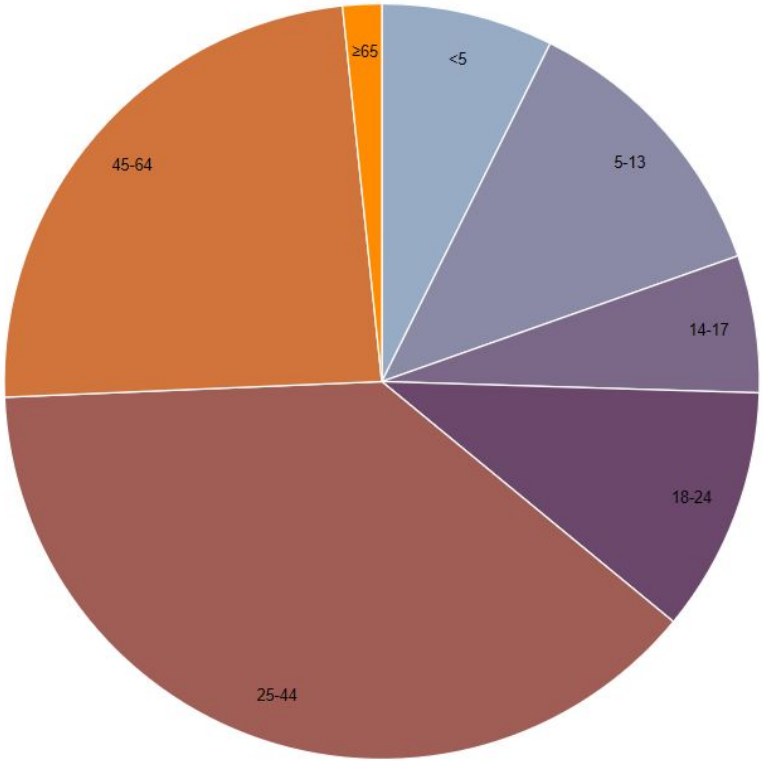
```
<svg width="960" height="500"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>

<script>

// appending the svg object to the svg id of the page
var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    radius = Math.min(width, height) / 2,
    g = svg.append("g").attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");

// setting color scale for the pie chart
var color = d3.scaleOrdinal(["#98abc5", "#8a89a6", "#7b6888", "#6b486b", "#a05d56", "#d0743c", "#ff8c00"]);

var pie = d3.pie()
    .sort(null)
    .value(function(d) { return d.population; });

var path = d3.arc()
    .outerRadius(radius - 10)
    .innerRadius(0);

var label = d3.arc()
    .outerRadius(radius - 40)
    .innerRadius(radius - 40);
```

Pie chart using D3

```
// getting the data in csv format
d3.csv("https://gist.githubusercontent.com/mbostock/3887235/raw/24e421a21e7f10da21db203944420b41561b7108/data.csv", function(d) {
  d.population = +d.population;
  return d;
}, function(error, data) {
  if (error) throw error;

  var arc = g.selectAll(".arc")
    .data(pie(data))
    .enter().append("g")
      .attr("class", "arc");

  arc.append("path")
      .attr("d", path)
      .attr("fill", function(d) { return color(d.data.age); });

  arc.append("text")
      .attr("transform", function(d) { return "translate(" + label.centroid(d) + ")"; })
      .attr("dy", "0.35em")
      .text(function(d) { return d.data.age; });
});

</script>
```

Line graph using D3

Output ->



Dataset ->

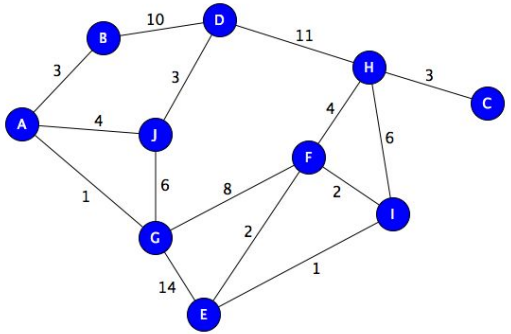
```
age,population
<5,2704659
5-13,4499890
14-17,2159981
18-24,3853788
25-44,14106543
45-64,8819342
≥65,612463
```

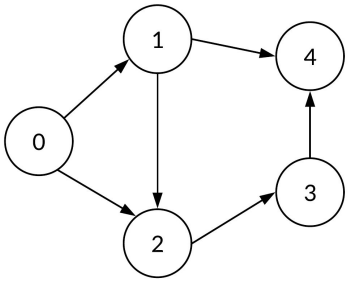
Graphical Analysis using D3

- Network analysis and network visualization are more common now with the growth of online social networks like Twitter and Facebook, as well as social media and linked data, all of which are commonly represented with network structures.
- In general, when dealing with networks you refer to the things being connected (like people) as nodes and the connections between them (such as being a friend on Facebook) as edges or links.
- Networks may also be referred to as graphs, because that’s what they’re called in mathematics.

Graphical Analysis using D3

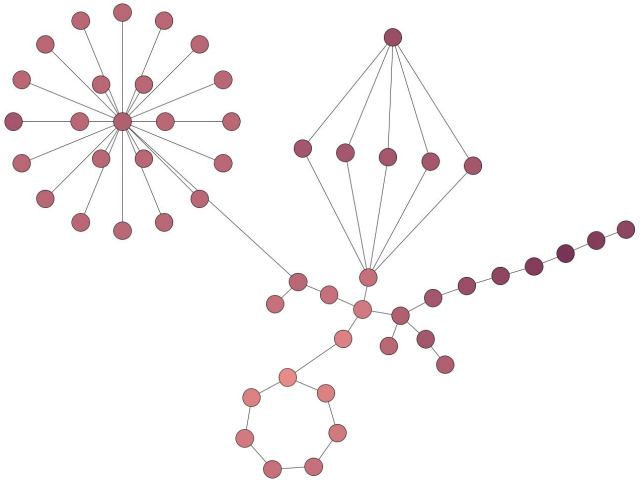
- Static Network diagrams
- Adjacency Matrices
- Arc Diagrams
- Force-directed Network diagrams

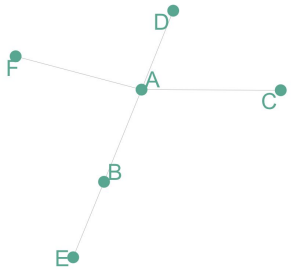


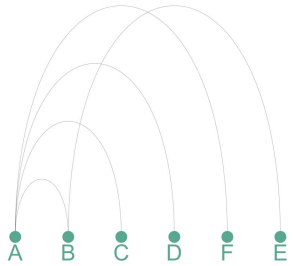


Adjacency Matrix

	0	1	2	3	4
0	0	1	1	0	0
1	0	0	1	0	1
2	0	0	0	1	0
3	0	0	0	0	1
4	0	0	0	0	0







Force-directed Network diagrams

- The force layout gets its name from the method by which it determines the most optimal graphical representation of a network
- The force() layout dynamically updates the positions of its elements to find the best fit. Unlike those layouts, it does it continuously in real time rather than as a preprocessing step before rendering.
- These forces push nodes away from each other, attract connected nodes to each other, and keep nodes from flying out of sight.

Force-directed Network diagrams

- The force layout gets its name from the method by which it determines the most optimal graphical representation of a network
- The force() layout dynamically updates the positions of its elements to find the best fit. Unlike those layouts, it does it continuously in real time rather than as a preprocessing step before rendering.
- These forces push nodes away from each other, attract connected nodes to each other, and keep nodes from flying out of sight.

Force-directed Network diagram example

```
<!DOCTYPE html>
<html>
  <head>
    <script type="text/javascript" src="https://d3js.org/d3.v3.js""></script>
    <style>
      body{ font: Arial 12px; text-align: center;}

      .link {
        stroke: #ccc;
      }

      .node text {
        pointer-events: none;
        font: sans-serif;
      }
    </style>
    <link rel="stylesheet" type="text/css" href="main.css">
  </head>
  <body>
  <script type="text/javascript">
  //Set margins and sizes
  var margin = {
    top: 20,
    bottom: 50,
    right: 30,
    left: 50
  };
  var width = 960 - margin.left - margin.right;
  var height = 700 - margin.top - margin.bottom;
  //Load Color Scale
  var c10 = d3.scale.category10();
  //Create an SVG element and append it to the DOM
  var svgElement = d3.select("body")
              .append("svg").attr({"width": width+margin.left+margin.right, "height": height+margin.top+margin.bottom})
              .append("g")
              .attr("transform","translate("+margin.left+","+margin.top+")");
```

Force-directed Network diagram example

```
//Load External Data
d3.json("https://gist.githubusercontent.com/mohdsanadzakirizvi/6fc325042ce110e1afc1a7124d087130/raw/ab9a310cfc2003f26131a7149950947645391e28/got_social_graph.json", function(dataset){
  //Extract data from dataset
  var nodes = dataset.nodes,
    links = dataset.links;
  //Create Force Layout
  var force = d3.layout.force()
          .size([width, height])
          .nodes(nodes)
          .links(links)
          .gravity(0.05)
          .charge(-200)
          .linkDistance(200);
  //Add links to SVG
  var link = svgElement.selectAll(".link")
        .data(links)
        .enter()
        .append("line")
        .attr("stroke-width", function(d){ return d.weight/10; })
        .attr("class", "link");
  //Add nodes to SVG
  var node = svgElement.selectAll(".node")
        .data(nodes)
        .enter()
        .append("g")
        .attr("class", "node")
        .call(force.drag);
  //Add labels to each node
  var label = node.append("text")
          .attr("dx", 12)
          .attr("dy", "0.35em")
          .attr("font-size", function(d){ return d.influence*1.5>9? d.influence*1.5: 9; })
          .text(function(d){ return d.character; });
```

Force-directed Network diagram example

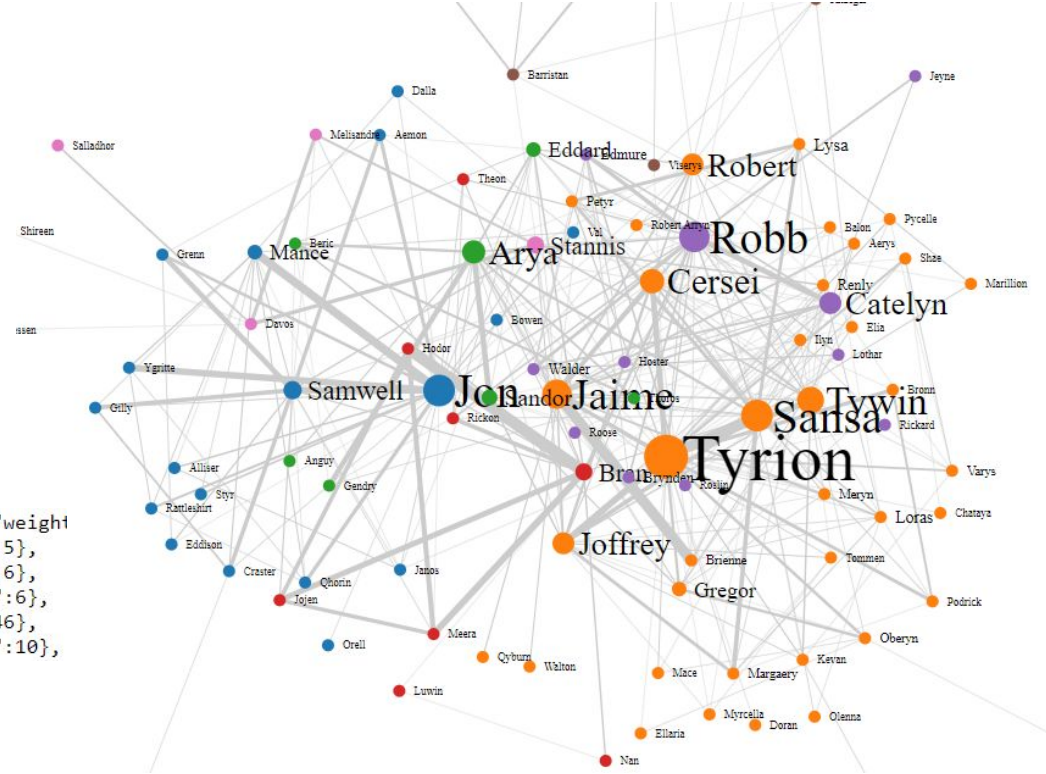
```
    //Add circles to each node
    var circle = node.append("circle")
                .attr("r", function(d){ return d.influence/2>5 ? d.influence/2 : 5; })
                .attr("fill", function(d){ return c10(d.zone*10); });
    //This function will be executed for every tick of force layout
    force.on("tick", function(){
      //Set X and Y of node
      node.attr("r", function(d){ return d.influence; })
        .attr("cx", function(d){ return d.x; })
        .attr("cy", function(d){ return d.y; });
      //Set X, Y of link
      link.attr("x1", function(d){ return d.source.x; })
      link.attr("y1", function(d){ return d.source.y; })
      link.attr("x2", function(d){ return d.target.x; })
      link.attr("y2", function(d){ return d.target.y; });
      //Shift node a little
        node.attr("transform", function(d) { return "translate(" + d.x + "," + d.y + ")"; });
    });
    //Start the force layout calculation
    force.start();
  });
  </script>
  </body>
</html>
```

Force-directed Network diagram example

Output ->

Dataset ->


```
{"nodes":[{"character":"Grenn","id":0,"influence":4,"zone":0},
{"character":"Robert","id":3,"influence":18,"zone":3},{"charac
{"character":"Mance","id":6,"influence":12,"zone":0},{"charact
{"character":"Bran","id":10,"influence":14,"zone":2},{"charact
{"character":"Gendry","id":13,"influence":4,"zone":5},{"charac
{"character":"Jon","id":16,"influence":26,"zone":0},{"characte
{"character":"Sandor","id":19,"influence":13,"zone":5},{"chara

"links":[{"source":77,"target":0,"weight":5},{"source":77,"target":1,"weight
{"source":100,"target":5,"weight":8},{"source":72,"target":6,"weight":5},
"source":65,"target":10,"weight":9},{"source":65,"target":11,"weight":6},
{"source":65,"target":2,"weight":11},{"source":65,"target":15,"weight":6},
source":65,"target":18,"weight":5},{"source":65,"target":19,"weight":46},
"source":54,"target":22,"weight":18},{"source":54,"target":23,"weight":10},
```

Hosting webpage on Apache web server using virtual host (MAC users)

Step 1: Install xcode

```
(base) srividyainampudi@Srividyas-Air ~ % xcode-select --install
xcode-select: error: command line tools are already installed, use "Software Update" to install updates
(base) srividyainampudi@Srividyas-Air ~ %
```

Step 2: Install Homebrew

Go to brew.sh in your browser and copy the command there to your terminal-
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"

```
(base) srividyainampudi@Srividyas-Air ~ % /bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
==> Checking for `sudo` access (which may request your password)...
==> This script will install:
/opt/homebrew/bin/brew
/opt/homebrew/share/doc/homebrew
/opt/homebrew/share/man/man1/brew.1
/opt/homebrew/share/zsh/site-functions/_brew
/opt/homebrew/etc/bash_completion.d/brew
/opt/homebrew
```

```
==> Installation successful!

==> Homebrew has enabled anonymous aggregate formulae and cask analytics.
Read the analytics documentation (and how to opt-out) here:
  https://docs.brew.sh/Analytics
No analytics data has been sent yet (nor will any be during this install run).

==> Homebrew is run entirely by unpaid volunteers. Please consider donating:
  https://github.com/Homebrew/brew#donations

==> Next steps:
- Run these three commands in your terminal to add Homebrew to your PATH:
    echo '# Set PATH, MANPATH, etc., for Homebrew.' >> /Users/srividyainampudi/.zprofile
    echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/srividyainampudi/.zprofile
    eval "$(/opt/homebrew/bin/brew shellenv)"
- Run brew help to get started
- Further documentation:
    https://docs.brew.sh
```

Step 3: Add Homebrew to your PATH
Follow instructions after installation

```
(base) srividyainampudi@Srividyas-Air ~ % echo '# Set PATH, MANPATH, etc., for Homebrew.' >> /Users/srividyainampudi/.zprofile
(base) srividyainampudi@Srividyas-Air ~ % echo 'eval "$(/opt/homebrew/bin/brew shellenv)"' >> /Users/srividyainampudi/.zprofile
(base) srividyainampudi@Srividyas-Air ~ % eval "$(/opt/homebrew/bin/brew shellenv)"
(base) srividyainampudi@Srividyas-Air ~ %
```

Step 4: Install apache2

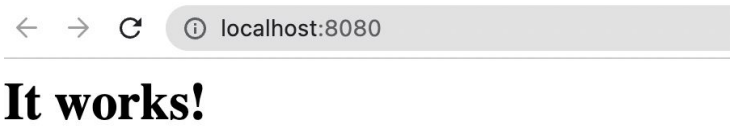
Command - brew install apache2

```
(base) srividyainampudi@Srividyas-Air ~ % brew install apache2
==> Downloading https://ghcr.io/v2/homebrew/core/apr/manifests/1.7.0_3
######################################################################## 100.0%
==> Downloading https://ghcr.io/v2/homebrew/core/apr/blobs/sha256:02e6b44b3284fa471cce15592a8666
==> Downloading from https://pkg-containers.githubusercontent.com/ghcr1/blobs/sha256:02e6b44b328
######################################################################## 100.0%
```

Step 5: Start apache server

```
(base) srividyainampudi@Srividyas-Air ~ % sudo apachectl start
(base) srividyainampudi@Srividyas-Air ~ %
```

This will start Apache HTTP server which can be tested by visiting localhost on the browser. The localhost 8080 gives the response as shown below:



Step 6: Open httpd config file

- If you're using Intel-based Mac: vim /usr/local/etc/httpd/httpd.conf
- If you're using Mac with Apple Silicon: vim /opt/homebrew/etc/httpd/httpd.conf

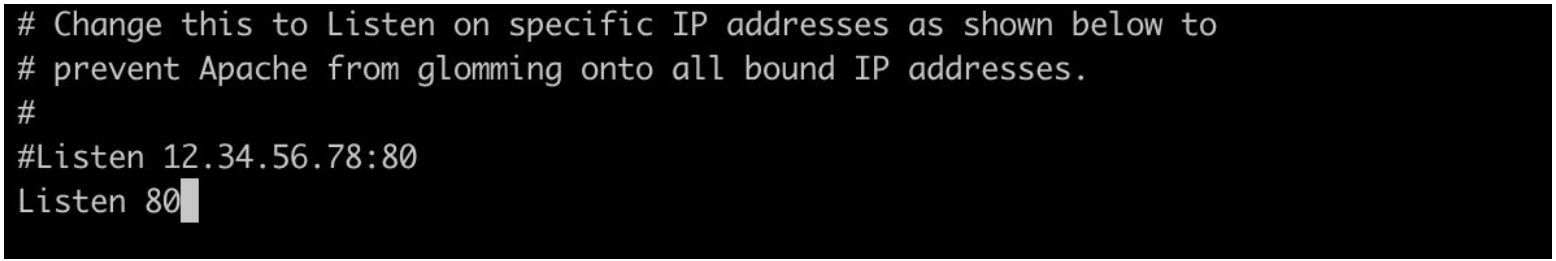
Step 7: Update these lines

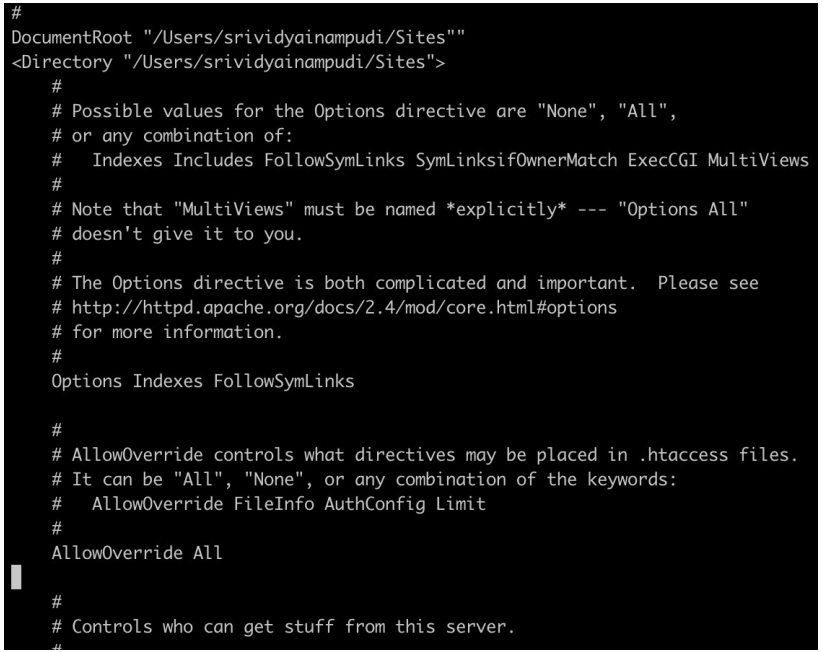
Listen 8080 to Listen 80

DocumentRoot "/usr/local/var/www" to DocumentRoot "/Users/your_account/Sites"

<Directory "/usr/local/var/www"> to <Directory "/Users/your_account/Sites">

AllowOverride None to AllowOverride All





Step 8: Uncomment this line
LoadModule rewrite_module lib/httpd/modules/mod_rewrite.so

Update these lines
User _www to User your_account
Group _www to Group staff

```
LoadModule alias_module lib/httpd/modules/mod_alias.so
LoadModule rewrite_module lib/httpd/modules/mod_rewrite.so

<IfModule unixd_module>
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# It is usually good practice to create a dedicated user and group for
# running httpd, as with most system services.
#
User srividyainampudi
Group staff
```

Step 9: Update this line
ServerName www.example.com:8080 to ServerName localhost

```
# If your host doesn't have a registered DNS name, enter its IP address here.
#
ServerName localhost

#
# Deny access to the entirety of your server's filesystem. You must
# explicitly permit access to web content directories in other
# <Directory> blocks below.
#
```

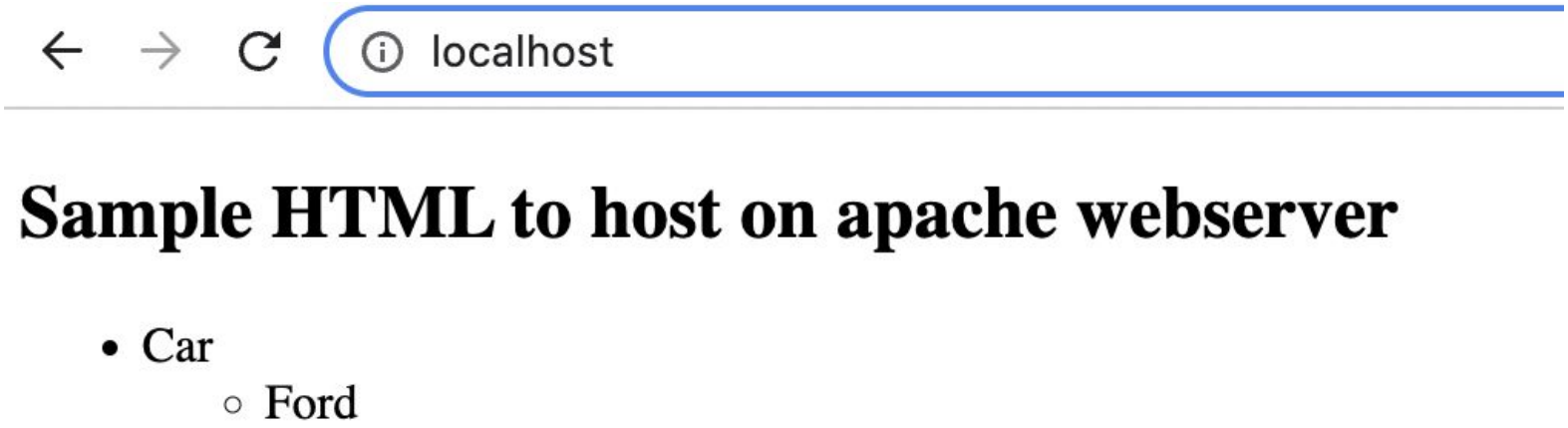
Step 10: Create Sites folder and add your html file to the folder

```
(base) srividyainampudi@Srividyas-Air ~ % mkdir Sites
(base) srividyainampudi@Srividyas-Air ~ %
(base) srividyainampudi@Srividyas-Air ~ %
(base) srividyainampudi@Srividyas-Air ~ % cd Sites
(base) srividyainampudi@Srividyas-Air Sites % ls
jsexample.html
```

Step 11. Restart apache

```
(base) srividyainampudi@Srividyas-Air Sites % sudo apachectl stop
(base) srividyainampudi@Srividyas-Air Sites %
(base) srividyainampudi@Srividyas-Air Sites %
(base) srividyainampudi@Srividyas-Air Sites %
(base) srividyainampudi@Srividyas-Air Sites % sudo apachectl start
```

Step 12: Test localhost to see your hosted website
http://localhost



Step 13: Stop apache

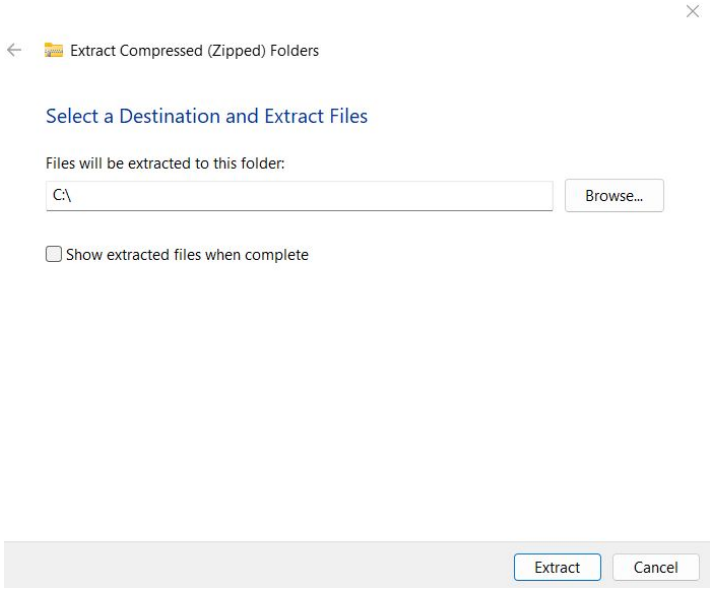
```
(base) srividyainampudi@Srividyas-Air Sites % sudo apachectl stop
Password:
(base) srividyainampudi@Srividyas-Air Sites %
```

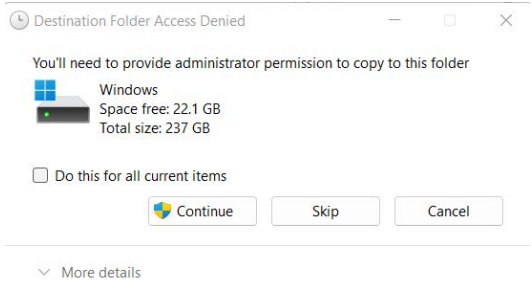
Hosting webpage on Apache web server using virtual host (Windows users)

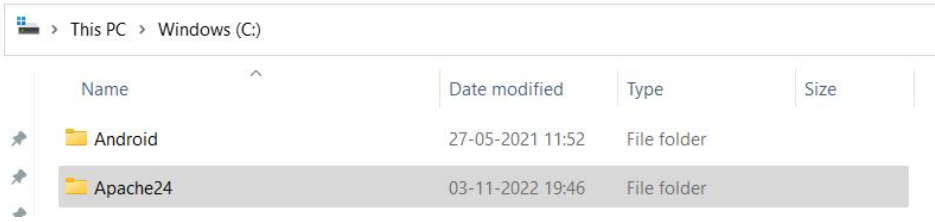
Step 1: Download Apache lounge zip file - https://www.apachelounge.com/download/

Download 64bit if your system is 64bit else download 32bit zip file

Step 2: Once zip file is finished downloading unzip it to C drive. Click Yes for administrative privileges.



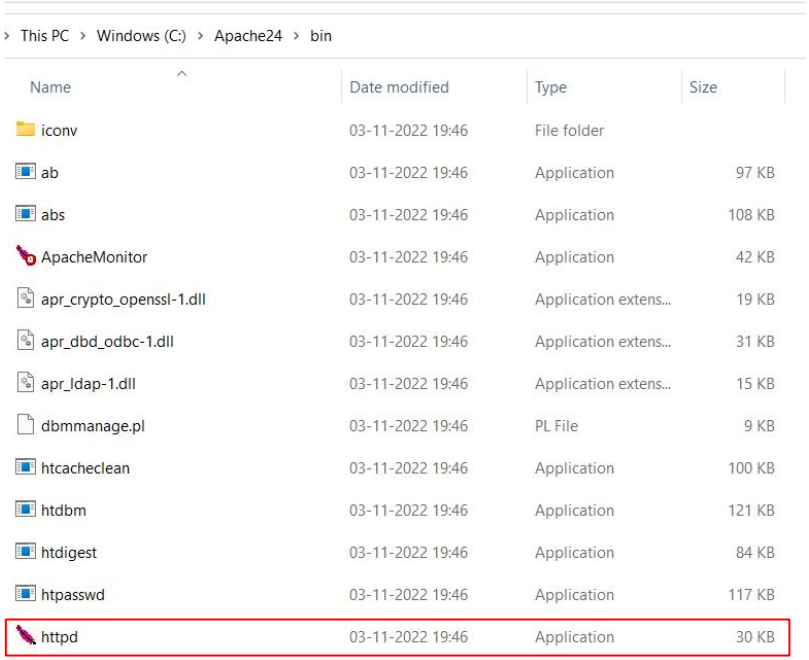


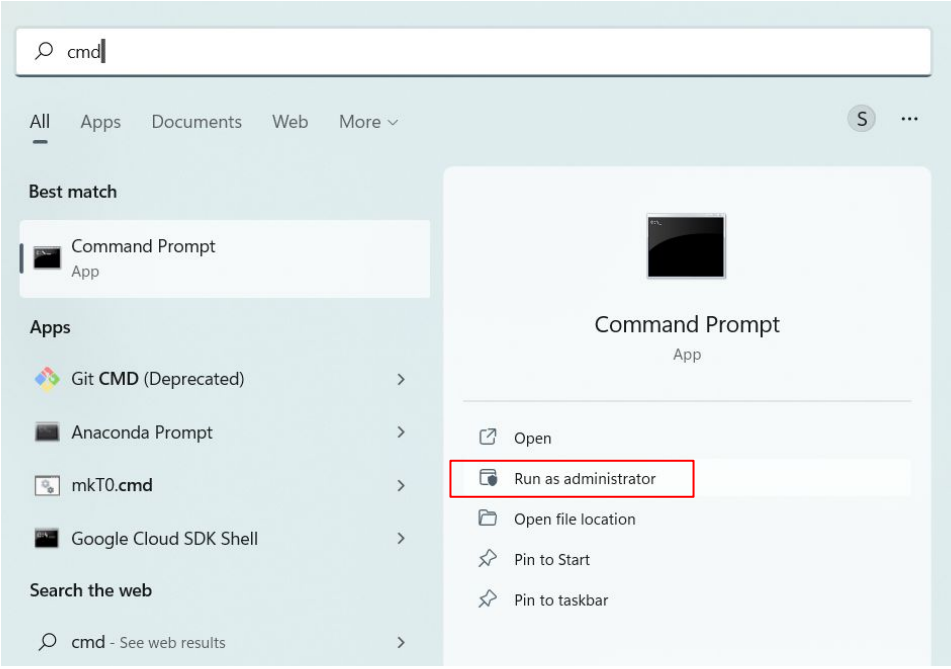


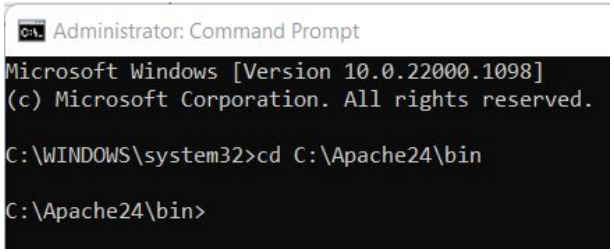
Step 3: Go to **"C:\Apache24\bin\"** and find httpd and we need to install this httpd. This is what is actually going to run apache. So we will go to start menu and type cmd to get command prompt. Right click as select **Run as administrator.** Click yes if it asks you for permission.

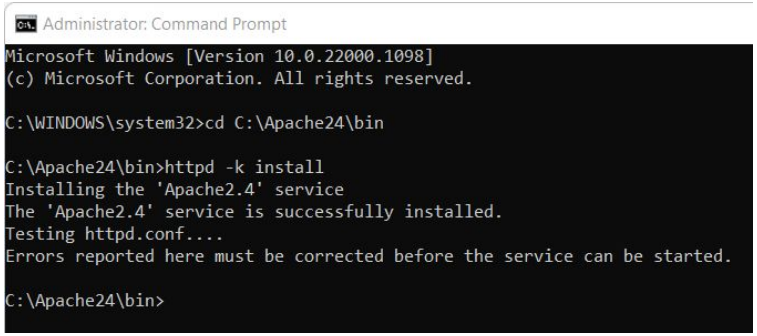
Step 4: Change directory to **C:\Apache24\bin** by typing **cd C:\Apache24\bin** in the terminal

Step 5: Now that we are in bin directory we will install httpd by typing the command **httpd -k install.** Allow access


This PC › Windows (C:) › Apache24 › bin
Name
Date modified
Type
Size
iconv 03-11-2022 19:46 File folder
ab 03-11-2022 19:46 Application 97 KB
abs 03-11-2022 19:46 Application 108 KB
ApacheMonitor 03-11-2022 19:46 Application 42 KB
apr_crypto_openssl-1.dll 03-11-2022 19:46 Application extens... 19 KB
apr_dbd_odbc-1.dll 03-11-2022 19:46 Application extens... 31 KB
apr_ldap-1.dll 03-11-2022 19:46 Application extens... 15 KB
dbmmanage.pl 03-11-2022 19:46 PL File 9 KB
htcacheclean 03-11-2022 19:46 Application 100 KB
htdbm 03-11-2022 19:46 Application 121 KB
htdigest 03-11-2022 19:46 Application 84 KB
htpasswd 03-11-2022 19:46 Application 117 KB
httpd 03-11-2022 19:46 Application 30 KB

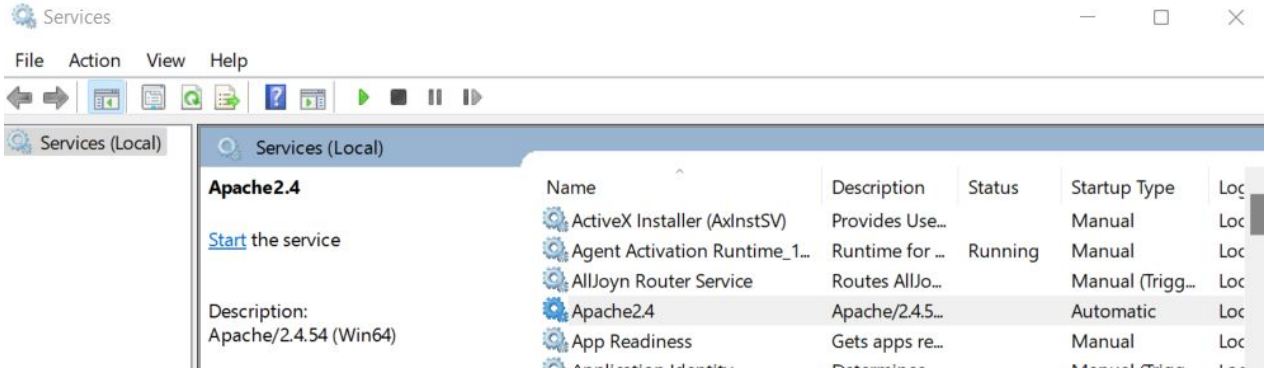

cmd
All
Apps
Documents
Web
More
S
Best match
Command Prompt
App
Apps
Git CMD (Deprecated)
Anaconda Prompt
mkT0.cmd
Google Cloud SDK Shell
Search the web
cmd - See web results
Command Prompt
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar

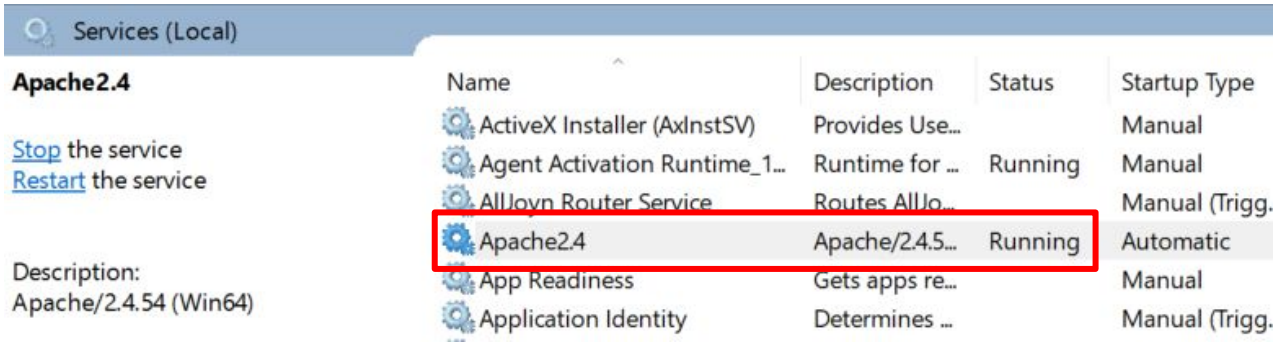

Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>cd C:\Apache24\bin
C:\Apache24\bin>


Administrator: Command Prompt
Microsoft Windows [Version 10.0.22000.1098]
(c) Microsoft Corporation. All rights reserved.
C:\WINDOWS\system32>cd C:\Apache24\bin
C:\Apache24\bin>httpd -k install
Installing the 'Apache2.4' service
The 'Apache2.4' service is successfully installed.
Testing httpd.conf....
Errors reported here must be corrected before the service can be started.
C:\Apache24\bin>

Step 6: Now Apache is installed as a service. If we go to Windows and type in Services in search bar and open Services we can see that Apache2.4 is installed.

Step 7: Now we will start the APache service by clicking start on left side. And then it will say that it is running. We can stop or restart the service on the left side if we need to.



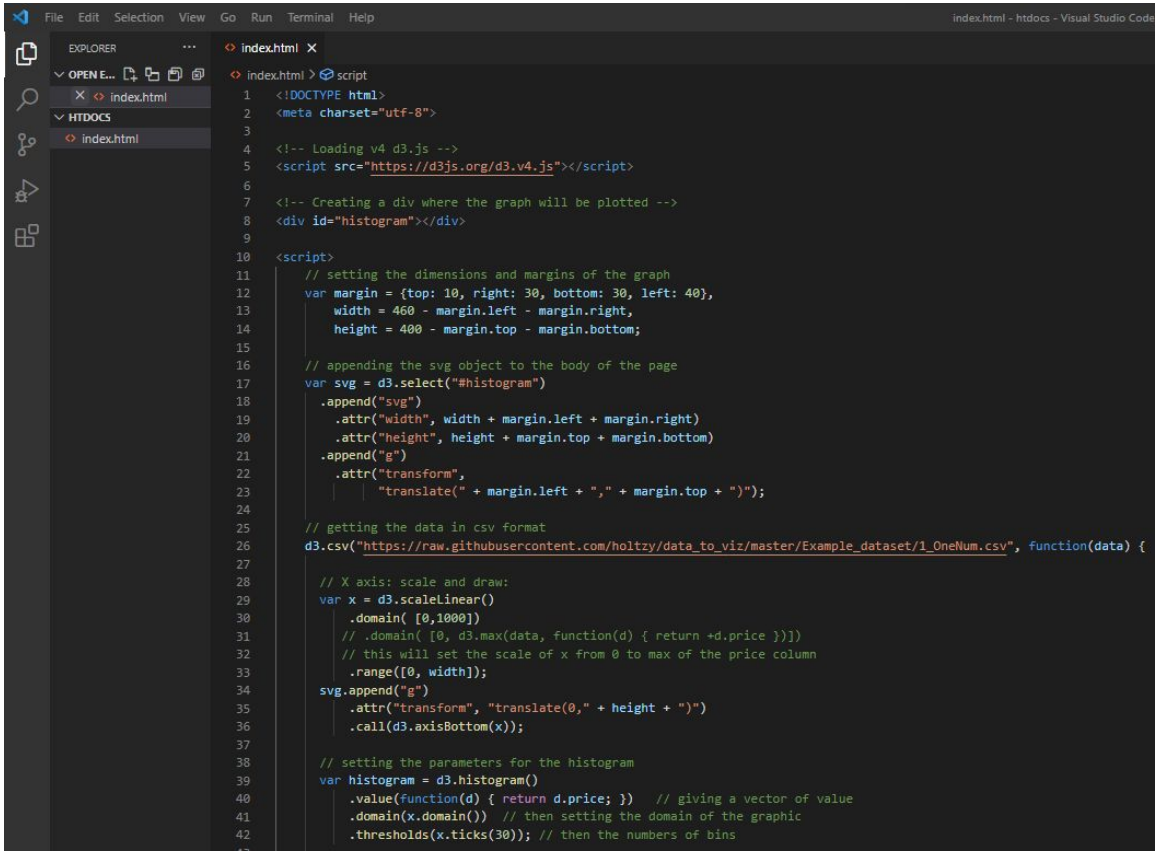


Step 8: Now to check if it is running. We will go to 127.0.0.1 in your browser it shows that it works!

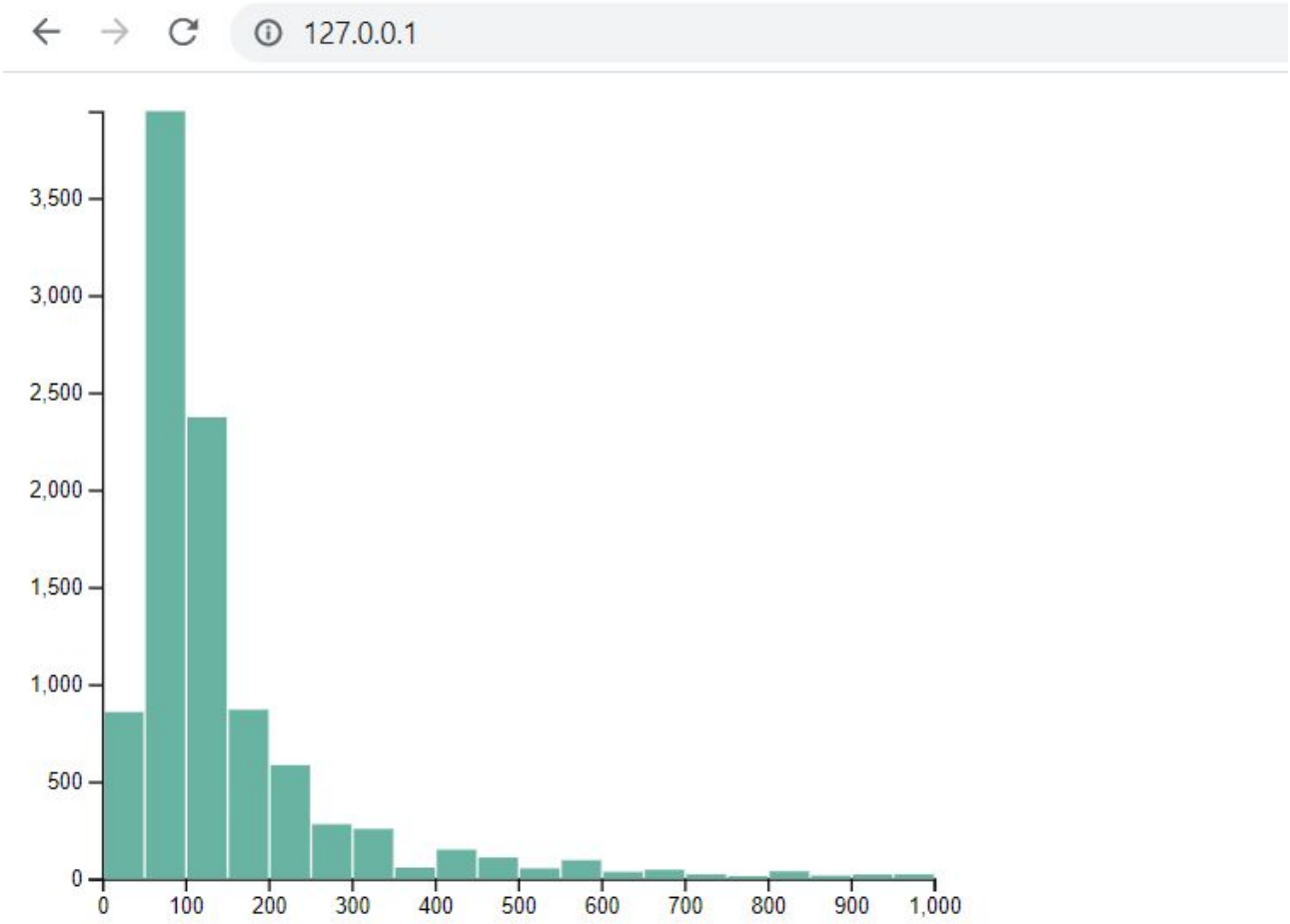
Step 9: If you have any errors you can check them in the logs folder in **C:\Apache24\logs** in errors.txt file

Step 10: Now to host our webpage we will go to **C:\Apache24\htdocs** and change the index.html to be your required html file and then again open 127.0.0.1 to see our webpage.

I have hosted the basic histogram html file for example.




```
var bins = histogram(data);

// Y axis: scale and draw:
var y = d3.scaleLinear()
    .range([height, 0]);
    y.domain([0, d3.max(bins, function(d) { return d.length; })]);
svg.append("g")
    .call(d3.axisLeft(y));

// append the bar rectangles to the svg element
svg.selectAll("rect")
    .data(bins)
    .enter()
    .append("rect")
      .attr("x", 1)
      .attr("transform", function(d) { return "translate(" + x(d.x0) + "," + y(d.length) + ")"; })
      .attr("width", function(d) { return x(d.x1) - x(d.x0) -1 ; })
      .attr("height", function(d) { return height - y(d.length); })
      .style("fill", "#69b3a2")

});
</script>
```



We can see that our d3 html file is hosted on Apache web server when we go to 127.0.0.1 on browser

References

- https://www.w3schools.com/js/js_htmldom.asp
- Vitaly Shmatikov CS 345 Introduction to JavaScript
- Sarbjit Kaur Introduction to HTML
- D3.js
- https://d3js.org/
- https://developer.mozilla.org/en-US/docs/web/SVG
- https://livebook.manning.com/book/d3js-in-action-second-edition/chapter-7/70
- https://d3-graph-gallery.com/index.html

Acknowledgement

Part of the slides is credited to Juncai Liu and Yiwen Fang

THANK YOU