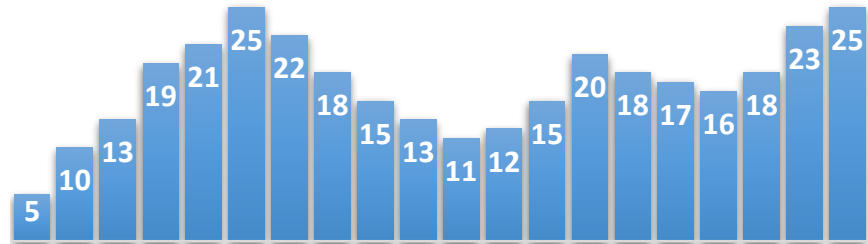
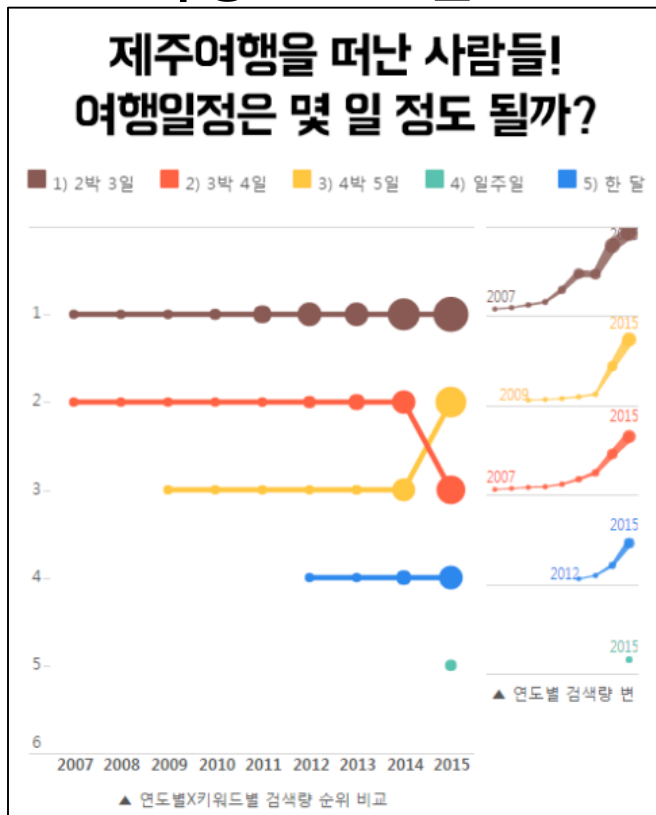
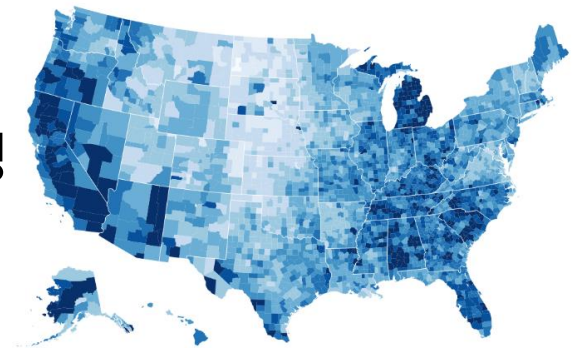


D3.js 들어가기 전

D3.js 들어가기 전

데이터 시각화?

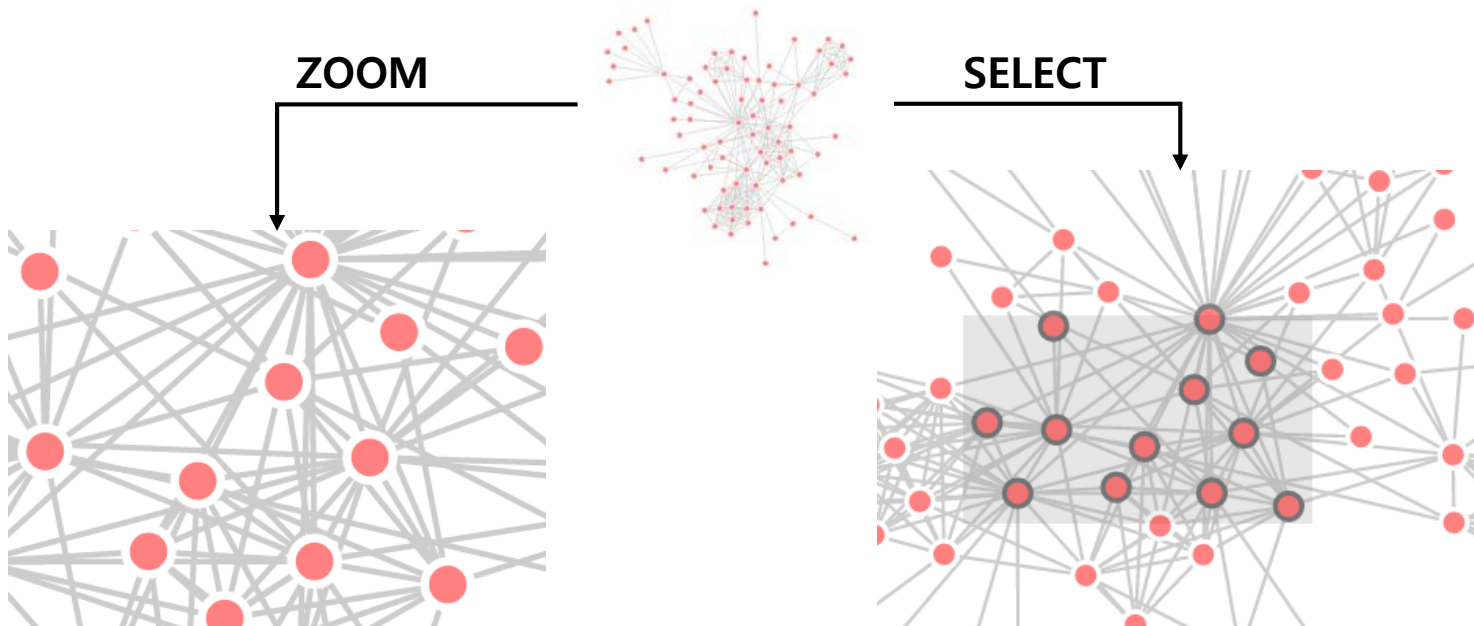
- 정보를 시각적 요소와 함께 매핑하는 과정
- 데이터를 이해한 후 그 값을 시각 요소 속성으로 표현



D3.js 들어가기 전

상호작용?

- 시각화 범위 한정 (모니터.. 스크린..)
> 표현해야 하는 데이터 제한
- 데이터 집합 ↑↑↑ 적절한 표현 어려움
- 키보드, 마우스, 터치 등 사람들이 스스로 데이터 탐색 가능
>> Zoom, Filter 를 통해 요구사항에 대한 자세히 보기 가능



D3.js 들어가기 전

웹?



- 웹 브라우저를 통해 전 세계를 대상으로 빠르게 전시
- 운영체제(Windows, Linux, OS X) 및 장비 종류(노트북, PC, 스마트폰, 태블릿)에 관계 없이 브라우저만으로 열람 및 경험
- 특정 업체 소프트웨어 / 플러그인을 사용하지 않아도 결과물을 어디서든지 확인 가능

D3.js 들어가기 전

무엇?

정 의

D3 (D3.js) 데이터 시각화 제작을 위한 자바스크립트 라이브러리

- 핵심 개발자 : 마이크 보스탁(Mike Bostock, <http://bost.ocks.org/mike>)

D3 : **D**ata - **D**riven **D**ocuments

데이터(Data)

HTML, SVG 처럼 웹 브라우저에서
렌더링할 수 있는 웹 기반 문서(Documents)

웹 문서를 데이터 중심으로 다룬다(Drive)

D3.js 들어가기 전

사용 범위?

<http://d3js.org>

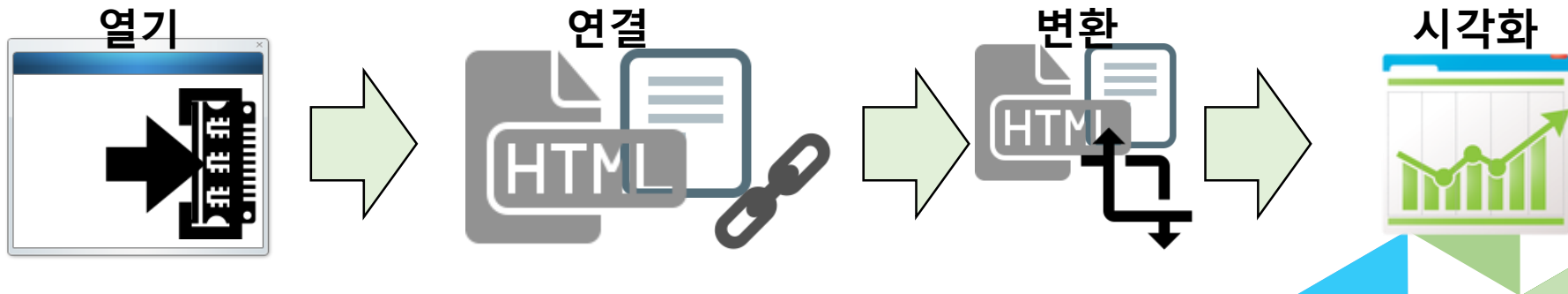
- 오픈소스 라이선스 채택
> BSD 라이선스

*BSD 라이선스란?

- 추가 비용 업싱 영리 목적이든 비영리 목적이든 코드를 마음대로 사용,
수정, 보강 가능.

- 동작 순서

1. 열기 : 브라우저 메모리로 데이터 부름
2. 연결 : 필요한 HTML 문서요소를 새로 만들어서 데이터를 엮음
3. 변환 : 각 문서요소에 역인 개별 데이터를 토대로 해당 문서요소를 변환
4. 시각화 및 상호작용 : 사용자 입력에 대한 반응으로 문서요소의 상태를 한 값에서 다른 값으로 전이



D3.js 들어가기 전

제한 요소

- “탐색용 시각화”가 아닌 “설명용 시각화” 제작을 지원할 목적

※ 탐색용 시각화: 데이터에서 의미 있는 유형 or 중요한 내용을 찾을 때 사용
(예시: Tableau)

설명용 시각화: 소통하고자 하는 핵심에만 집중

- 오래된 브라우저 지원 하지 않음 (예: IE 7, 8)

- SVG or GeoJSON 같은 벡터 방식에 최적화
= 비트맵 같은 이미지 타일은 다루지 않음

예외 사항으로 d3.geo.tile 과 같이 플러그인 기능 추가로 변화되고 있음

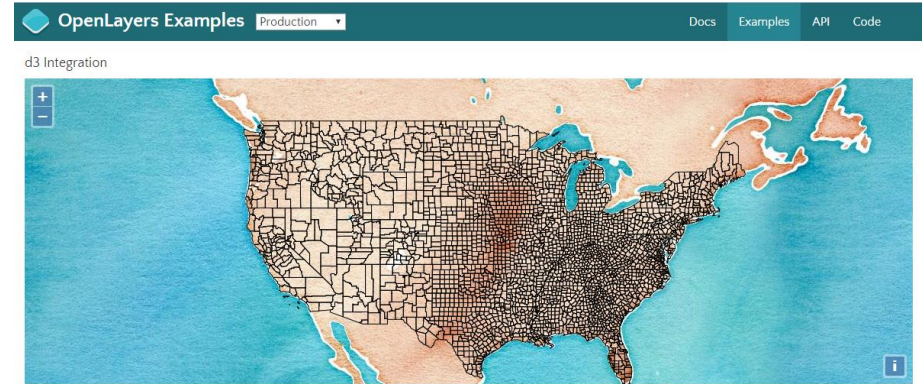
- 웹에서 시각화 함 = 데이터에 대한 숨김이 없음

∴ 공유하면 안 되는 데이터를 사용하면 안됨

D3.js 들어가기 전

비슷한 도구

- 그래프 시각화
 - Arbor.js (<http://arborjs.org>)
 - Sigma.js (<http://sigmajs.org>)
- 지오 매핑
 - Kartograph (<http://kartograph.org>)
 - Leaflet(<http://leaflet.cloudmade.com/>)
 - Modest Maps(<http://modestmaps.com>)
 - Polymaps(<http://polymaps.org>)
 - OpenLayers3(<http://openlayers.org>)



D3.js 들어가기 전

비슷한 도구

- 자유로운 그리기 도구
 - Processing.js (<http://processingjs.org>)
 - Paper.js(<http://paper.org>)
- 3차원 도구
 - PhiloGL (<http://www.senchalabs.org/philogl>)
 - Three.js (<http://mrdoob.github.com/three.js>)

D3.js 들어가기 전

D3 기반 제작 도구

- D3는 다소 복잡한 개발이 필요한 차트 시각화 라이브러리
- 편리하게 사용할 수 있도록 제작된 도구 사용 가능

- Crossfilter (<http://square.github.com/crossfilter>)

마이크 보스탁 주도로 개발된 대용량, 다변량 데이터 집합을 위한 라이브러리 상대적으로 작은 환경인 브라우저에서 '빅 데이터'를 쥐어짤 목적으로 사용하기 유용

- Cubism(<http://square.github.com/cubism>)

시계열 데이터를 시각화하는 d3 플러그인. 마이크 보스탁 개발

- Dashku (<http://dashku.com/>)

실시간 업데이트되는 데이터 대시보드와 위젯을 만드는 온라인 도구 폴 젠센 개발

D3.js 들어가기 전

D3 기반 제작 도구

- Polychart.js(<http://polychart.com/>)

사용 가능한 차트 종류 중 재사용 가능한 차트들의 모음이다. Polychart.js는 비영리 목적으로는 무료로 사용할 수 있음

- Tributary (<http://enjalot.com/>)

D3를 이용해서 라이브 코딩으로 실험해 볼 수 있는 멋진 도구이다. 랜 존슨 제작

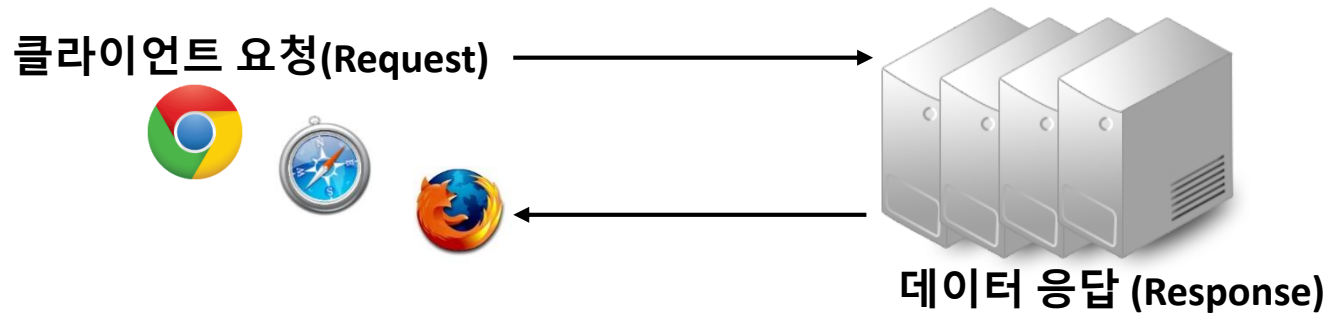
- C3 (<http://c3js.org/>)

D3를 편리하게 사용할 수 있는 확장 라이브러리

D3.js 들어가기 전

웹

- 웹 기본 요청 개념



- URL(Uniform Resource Locator) 과 URI(Uniform Resource Identifier) 구분

URL : 인터넷 상 올려진 자료들이 있는 주소 구성

URI : 인터넷에 있는 자원을 나타내는 유일한 주소



URI

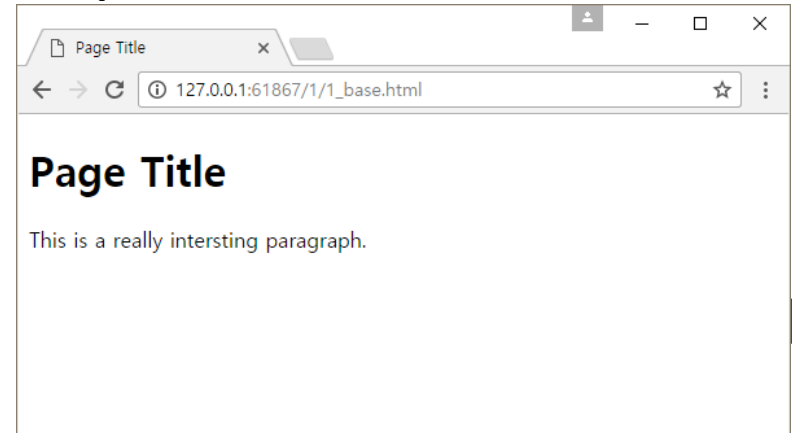
https://search.naver.com/search.naver → URL
?where=nexearch&query=한성대학교&sm=top_h ty&fbm=0&ie=utf8

D3.js 들어가기 전

HTML

하이퍼텍스트 마크업 언어(Hypertext Markup Language)
웹 브라우저를 위해 콘텐츠를 구조화하는데 사용

```
<!DOCTYPE html>
<html>
  <head>
    <title>Page Title</title>
  </head>
  <body>
    <h1>Page Title</h1>
    <p>This is a really intersting paragraph.</p>
  </body>
</html>
```



D3.js 들어가기 전

HTML (태그)

- 기본 문서 요소

<!DOCTYPE html>

표준 문서 타입 선언. 문서 최 상단 존재

<html>

문서의 전체 HTML 콘텐츠에 대한 부모

<head>

문서에 대한 모든 부가 정보 작성

<title>

문서 제목으로 브라우저 창 상단 표시 (head 내부 작성)

<body>

페이지에 시각적으로 표현되는 주 콘텐츠

D3.js 들어가기 전

HTML (태그)

- 많이 사용되는 문서 요소

`<h1>`, `<h2>`, `<h3>`, `<h4>`

단계별 제목 크기

`<p>`

문서 글 단락

``, ``, ``

목록 정의 (ul: 번호가 없는 목록, ol: 번호가 있는 목록)

``, ``

내용을 강조하기 위한 태그 (em: 강조, strong: 매우 강조)

`<a>`

링크 사용하지만 링크를 지정하지 않으면 밑줄과 파란색 텍스트로 렌더링

D3.js 들어가기 전

HTML (태그)

- 많이 사용되는 문서 요소

`<div>`

문서 안 임의로 분할한 영역으로 문서 요소를 묶고 담는 데 사용

``

p 태그와 비슷한 문서 글 단락, 더 좁은 문단

D3.js 들어가기 전

HTML (속성)

- 시작하는 태그 안 속성 삽입 가능

```
<tagname property="value"></tagname>
```

- class & id
 - 가장 중요한 속성으로 콘텐츠에서 구체적 한 부분 찾을 때 사용
 - 특히, 자바스크립트 코드로 문서 요소를 알아낼 때 사용
 - class의 경우 미리 정해놓은 다양한 스타일 적용 가능

Class

- 한 문서에 다중 적용 가능
- 한 태그에 여러 개 class 지정 가능

```
<p class="uplifting">brilliant paragraph</p>  
<p class="uplifting awesome">  
  Awe-inspring paragraph  
</p>
```

id

- 한 문서에 무조건 하나의 id
- 한 태그에 하나 지정

```
<div id="content">  
<div id="visualization"></div>  
<div id="button"></div>
```

D3.js 들어가기 전

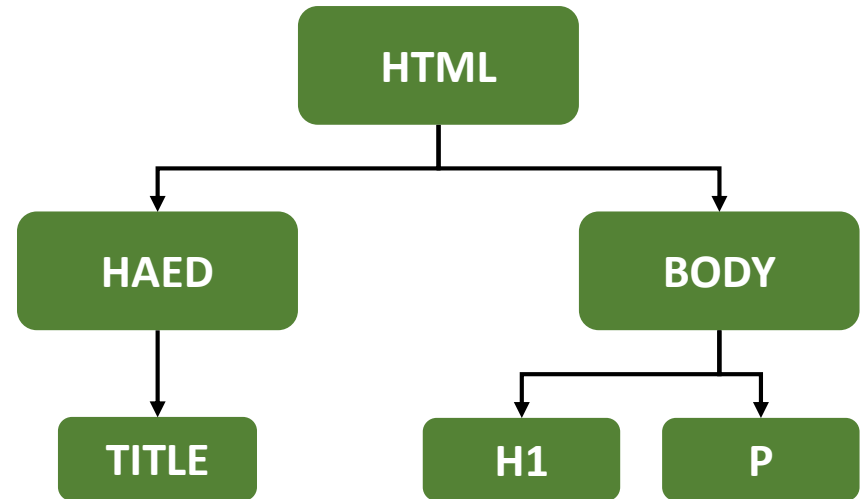
HTML (DOM)

- 문서 객체 모델 (DOM : document Object Model)

HTML과 XML 과 같이 구조화된 문서를 표현하는 형식

```
<html>
  <head>
    <title>HOME</title>
  </head>
  <body>
    <h1>Breaking News</h1>
    <p></p>
  </body>
</html>
```

계층적 구조



- H1과 p1은 서로 형제 문서 요소
- body는 h1, p의 부모 문서 요소
- 이 문서의 모든 문서 요소는 html의 자손

D3.js 들어가기 전

CSS

문서 객체 모델을 시각적 표현으로 꾸미는데 사용

```
Body{  
  background-color : white;  
  color : black;  
}
```

- 선택자 (selector) / 프로퍼티 (property)

```
selector{  
  property : value;  
  property : value;  
  property : value;  
  ...  
}
```

같은 프로퍼티 묶음은 콤마로 구분한 선택자에 한 번에 적용 가능

```
selectorA,  
selectorB,  
selectorC{  
  property : value;  
  ...  
}
```

D3.js 들어가기 전

CSS

- D3는 필요한 문서 요소를 찾을 때 , CSS 방식의 선택자를 자주 사용
- 선택자 종류
 - 타입 선택자: 가장 간단한 형태로 동명의 DOM 문서 요소를 찾음
 - 자손 선택자: 어떤 문서 요소를 감싸인 문서 요소들을 찾음
 - 클래스 선택자: 문서 요소 종류에 관계없이 지정한 클래스가 할당된 문서 요소를 모두 찾음
 - 아이디(id) 선택자: 주어진 id가 있는 문서요소 하나를 선택

h1
p1
strong
div
...

.caption
.label
...

#header
#nav
#export

D3.js 들어가기 전

CSS

- D3는 필요한 문서 요소를 찾을 때 , CSS 방식의 선택자를 자주 사용
- 선택자 종류
 - 타입 선택자: 가장 간단한 형태로 동명의 DOM 문서 요소를 찾음
 - 자손 선택자: 어떤 문서 요소를 감싸인 문서 요소들을 찾음
 - 클래스 선택자: 문서 요소 종류에 관계없이 지정한 클래스가 할당된 문서 요소를 모두 찾음
 - 아이디(id) 선택자: 주어진 id가 있는 문서요소 하나를 선택

h1
p1
strong
div
...

.caption
.label
...

#header
#nav
#export

D3.js 들어가기 전

CSS

- 스타일 참조하기

HTML 문서 내 삽입

```
<!DOCTYPE html>
<html>
  <head>
    <style type="text/css">
      p{
        font-size: 24px;
        font-weight: bold;
        background-color: red;
        color: white;
      }
    </style>
  </head>
  <body>
    <p>If I were to ask you, as a mere paragraph, would you say that I have style?</p>
  </body>
</html>
```



D3.js 들어가기 전

CSS

CSS 파일을 삽입

```
<!DOCTYPE html>
<html>
  <head>
    <link rel="stylesheet" href="4_cssReference_atFile.css">
  </head>
  <body>
    <p>If I were to ask you, as a mere paragraph, would you say that I have style?</p>
  </body>
</html>
```

문서에 인라인으로 삽입

```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <p style="color: blue; font-size:48; font-style: italic;">If I were to ask you, as a mere
    paragraph, would you say that I have style?</p>
  </body>
</html>
```

D3.js 들어가기 전

자바스크립트

브라우저에 이미 떠 있는 페이지의 DOM을 조작해서 페이지를 동적으로 만들 수 있는 스크립트 언어

- 변수

- 데이터를 위한 공간으로 일반 변수는 다음과 같은 하나의 값을 가짐
 - > `var number = 5;`
- 초기 설정 값에 따라 타입이 자동으로 결정되므로 타입 별 변수를 정할 필요가 없음

```
var val = 5;  
var val = "a";  
var val = true;  
var val = 2.56;  
var val = "a value";
```

필요에 따라 타입 변경도 가능

D3.js 들어가기 전

자바스크립트

- 배열

- 단일 변수에 적절히 저장된 값을 나열한 형태

```
> var numbers = [5, 10, 15, 20, 25];
```

- 배열 인덱스는 0부터 시작
- 모든 데이터 타입을 담을 수 있음

※ 다른 타입의 값들을 저장할 수도 있지만 권장하는 형태는 아님

```
var string= ["a", "b", "c", "d", "e"];
```

```
var multi = [1, 2, 3, 4.6, 5.6, "oh boy", true]
```

D3.js 들어가기 전

자바스크립트

- 객 체
 - 배열보다 좀 더 복잡한 데이터 집합은 객체에 담는 게 더 적절
 - 자바스크립트 객체를 데이터 시각화를 위한 사용자 정의 데이터 구조체

```
var fruit = {  
  kind: "grape",  
  color: "red",  
  quantity : 12,  
  tasty : true  
}
```

중괄호로 표현, 괄호 안 프로퍼티 명과 값을 넣음

값을 참조하기 위해서는 구두점(.) 사용

fruit.kind	출력 →	"grape"
fruit.color		"red"
fruit.quantity		12
fruit.tasty		true

D3.js 들어가기 전

자바스크립트

- 객체와 배열
 - 객체가 담긴 배열 or 배열이 담긴 객체

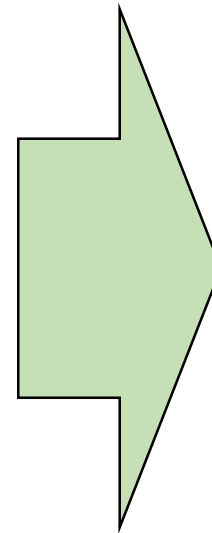
```
var fruits = [  
  {  
    kind: "grape",  
    color: "red",  
    quantity : 12,  
    tasty: true  
  },  
  {  
    kind: "kiwi",  
    color: "brown",  
    quantity: 98,  
    tasty : true  
  }  
];
```

fruits[1].kind

fruits[1].color

fruits[1].quantity

fruits[1].tasty



"kiwi"

"brown"

98

true

D3.js 들어가기 전

자바스크립트

- JSON (JavaScript Object Notation)
 - 데이터를 자바스크립트 객체로 최적화시키는 문법
 - 웹 기반 API에 많이 사용
 - XML보다 자바스크립트로 파싱하기 쉽고 빠름
 - 앞서 본 형태와 크게 다르지 않음

```
var fruits =  
{  
  "kind": "grape",  
  "color": "red",  
  "quantiry" : 12,  
  "tasty": true  
};
```

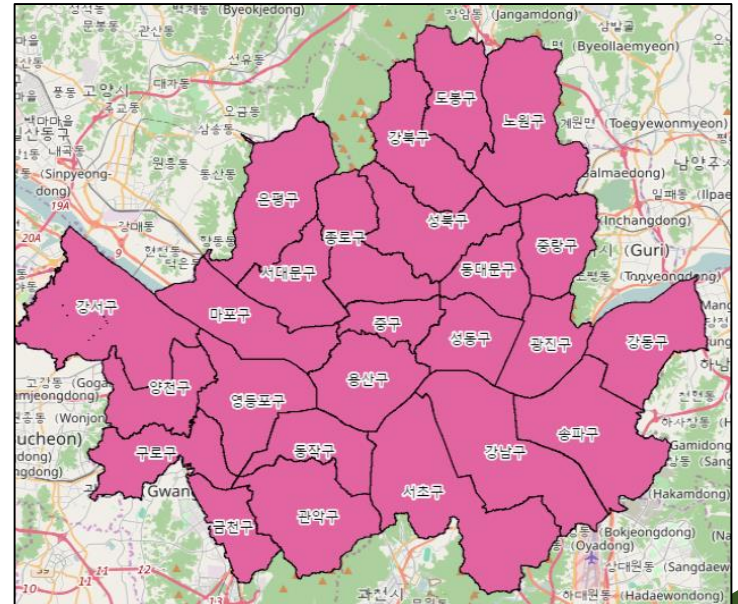
프로퍼티 명을 문자열로 만들기 위해 큰따옴표를 사용

D3.js 들어가기 전

자바스크립트

- GeoJSON (Geographic JavaScript Object Notation)
 - 지리정보를 저장할 목적으로 최적화된 JSON 객체의 특정한 형태
 - 선, 다각형 같은 도형, 다른 공간적인 특징도 저장 가능
 - 형태

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [150, 1282427, -24.471803]  
      },  
      "properties": {  
        "type": "town"  
      }  
    }  
  ]  
};
```



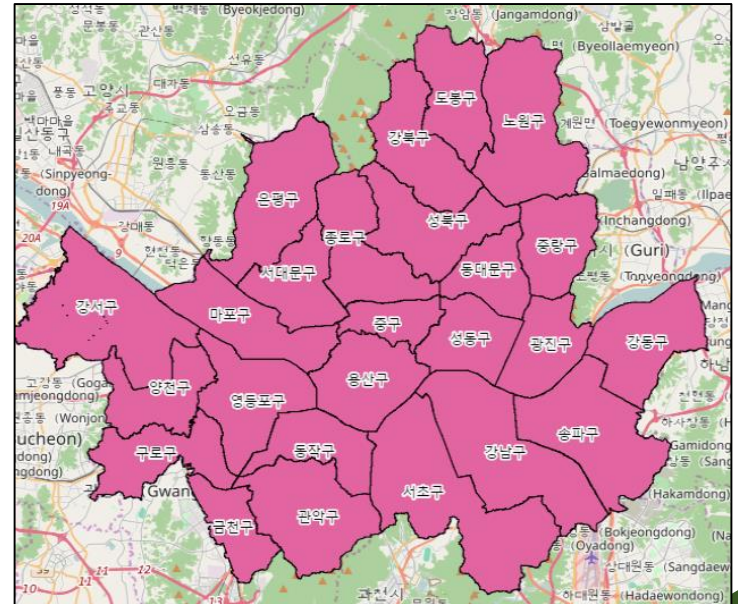
<http://geojson.org/geojson-spec.html>

D3.js 들어가기 전

자바스크립트

- GeoJSON (Geographic JavaScript Object Notation)
 - 지리정보를 저장할 목적으로 최적화된 JSON 객체의 특정한 형태
 - 선, 다각형 같은 도형, 다른 공간적인 특징도 저장 가능
 - 형태

```
{  
  "type": "FeatureCollection",  
  "features": [  
    {  
      "type": "Feature",  
      "geometry": {  
        "type": "Point",  
        "coordinates": [150, 1282427, -24.471803]  
      },  
      "properties": {  
        "type": "town"  
      }  
    }  
  ]  
};
```



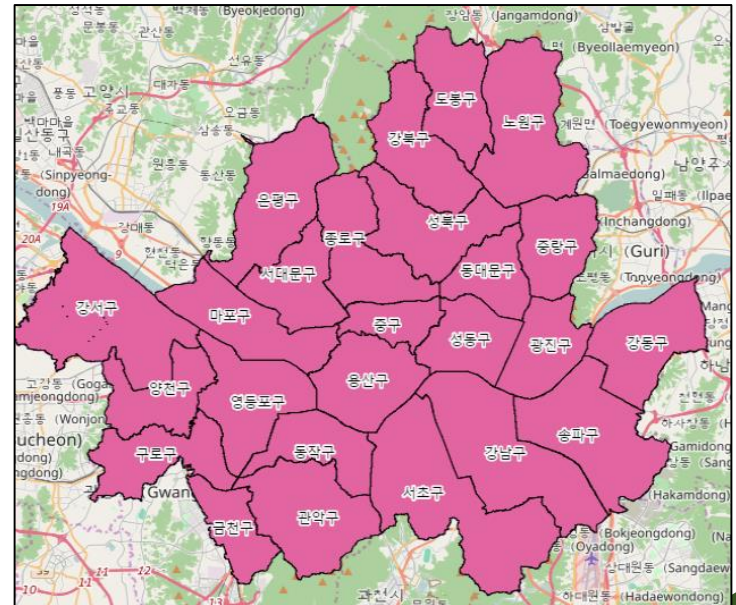
<http://geojson.org/geojson-spec.html>

D3.js 들어가기 전

자바스크립트

- GeoJSON (Geographic JavaScript Object Notation)
 - 지리정보를 저장할 목적으로 최적화된 JSON 객체의 특정한 형태
 - 선, 다각형 같은 도형, 다른 공간적인 특징도 저장 가능
 - 형태

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "type": "Feature",
      "geometry": {
        "type": "Point",
        "coordinates": [150, 1282427, -24.471803]
      },
      "properties": {
        "type": "town"
      }
    }
  ]
};
```



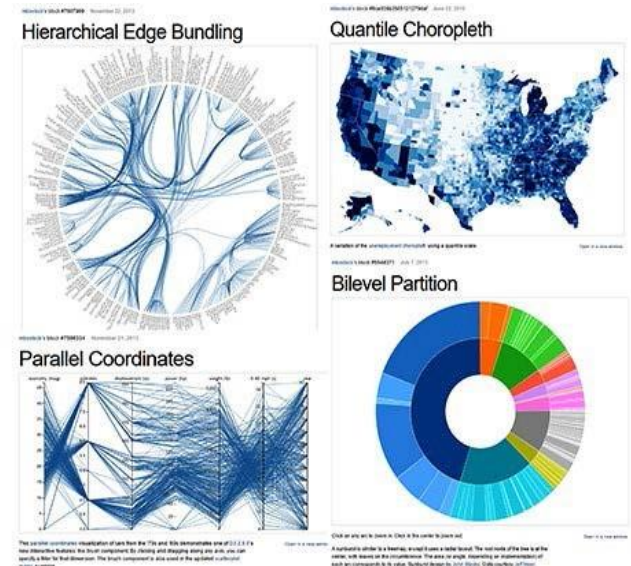
<http://geojson.org/geojson-spec.html>

D3.js 들어가기 전

D3 활용

- D3.js 는 바로 그래프를 구현할 수 있는 도구는 아니다. 엑셀처럼 직관적으로 그래프 종류를 선택할 수 있는 것도 아니고, R 언어의 'ggplot2' 패키지처럼 해당 데이터를 기반으로 그릴 수 있는 그래프의 기본 형태(막대, 스택, 분포도, 꺾은 선 그래프 등)를 제공하지도 않는다.

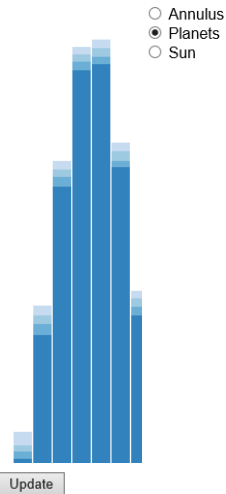
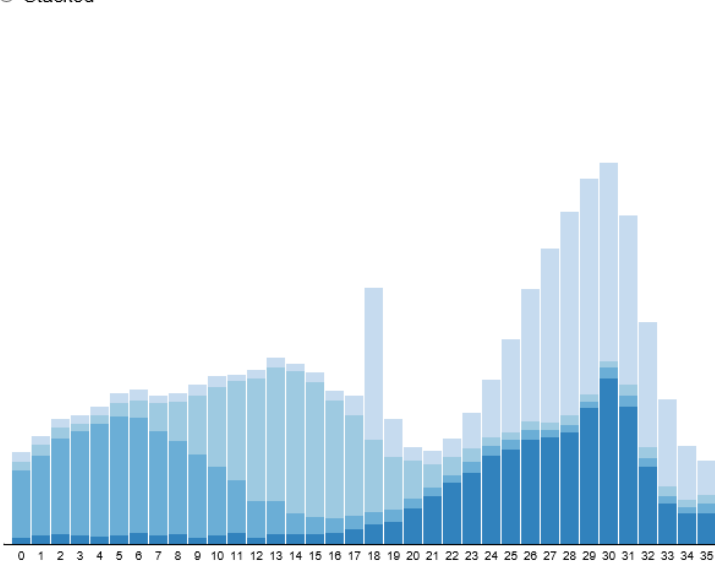
- 웹 프로그래밍에 활용되는 언어에 대한 기본적인 이해가 필요하다. 다만 직접 그리는 도구이기 때문에 그래프나 시각화는 활용 수준에 따라 폭넓게 구현할 수 있다. 원하는 대로 그릴 수 있다는 의미다. D3.js 개발자인 마이크 보스톡이 자신의 깃허브와 블로그를 통해 다양한 예제를 제공하고 있으므로, 이를 활용할 수 있다.



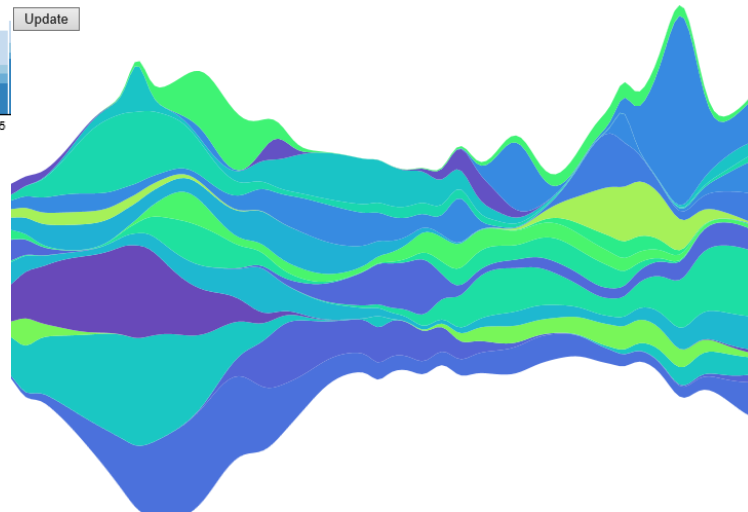
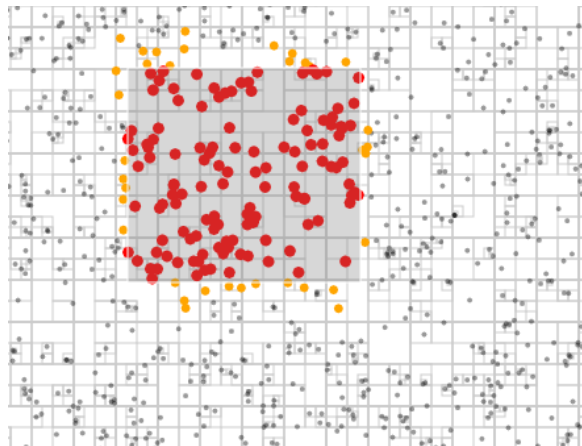
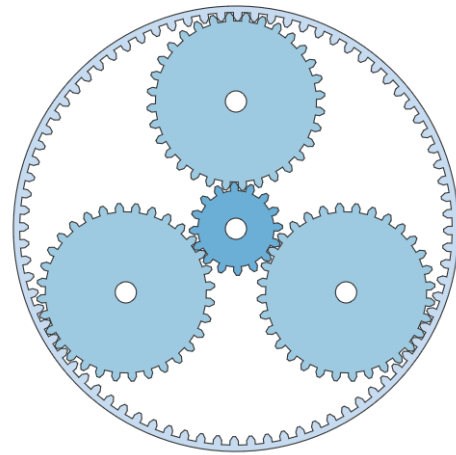
D3.js 들어가기 전

D3 활용

- Grouped
- Stacked



- Annulus
- Planets
- Sun



D3.js 들어가기 전

D3 활용

index.html

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>
```

```
body {
  font-family: "Helvetica Neue", Helv
  width: 960px;
  height: 500px;
  position: relative;
}
```

```
form {
  position: absolute;
  top: 1em;
  left: 1em;
}
```

```
path {
  fill-rule: evenodd;
  stroke: #333;
  stroke-width: 2px;
}
```

```
.sun path {
  fill: #6baed6;
}
```

```
.planet path {
  fill: #9ecae1;
}
```

```
.annulus path {
  fill: #c8dbef;
}
```

```
</style>
<form>
  <input type="radio" name="reference" id="ref-a">
  <label for="ref-annulus">Annulus</label><br>
  <input type="radio" name="reference" id="ref-p">
  <label for="ref-planet">Planets</label><br>
  <input type="radio" name="reference" id="ref-s">
  <label for="ref-sun">Sun</label>
</form>
<script src="https://d3js.org/d3.v4.min.js"></script>
</script>
```

```
var width = 960,
    height = 500,
    radius = 80,
    x = Math.sin(2 * Math.PI / 3),
    y = Math.cos(2 * Math.PI / 3);
```

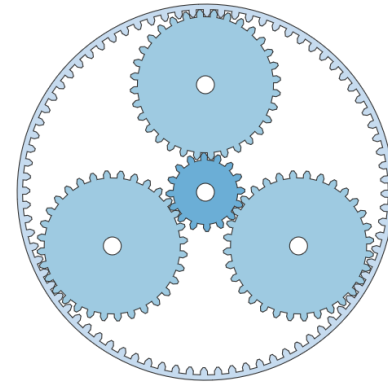
```
var offset = 0,
    speed = 4,
    start = Date.now();
```

```
var svg = d3.select("body").append("svg")
  .attr("width", width)
  .attr("height", height)
  .append("g")
  .attr("transform", "translate(" + width / 2 + "," + height / 2 + ")scale(.55)")
  .append("g");
```

```
var frame = svg.append("g")
  .datum({radius: Infinity});
```

```
frame.append("g")
  .attr("class", "annulus")
  .datum({teeth: 80, radius: -radius * 5, annulus: true})
```

○ Annulus
● Planets
○ Sun



추가 데이터 시각화 언어 R

<http://www.r-project.org>



[CRAN](#)
[Mirrors](#)
[What's new?](#)
[Task Views](#)
[Search](#)

[About R](#)
[R Homepage](#)
[The R Journal](#)

[Software](#)
[R Sources](#)
[R Binaries](#)
[Packages](#)
[Other](#)

[Documentation](#)
[Manuals](#)
[FAQs](#)
[Contributed](#)



[Home]

Download

[CRAN](#)

R Project

[About R](#)

[Logo](#)

[Contributors](#)

[What's New?](#)

[Mailing Lists](#)

[Bug Tracking](#)

[Development Site](#)

[Conferences](#)

[Search](#)

R Foundation

The R Project for Statistical Computing

Getting Started

R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).

If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News

- **R version 3.3.0 (Supposedly Educational)** has been released on 2016-05-03.
- **R version 3.2.5 (Very, Very Secure Dishes)** has been released on 2016-04-14. This is a rebadging of the quick-fix release 3.2.4-revised.
- **Notice XQuartz users (Mac OS X)** A security issue has been detected with the Sparkle update mechanism used by XQuartz. Avoid updating over insecure channels.
- The **R Logo** is available for download in high-resolution PNG or SVG formats.
- **useR! 2016** will take place at Stanford University, CA, USA, June 27 - June 30, 2016.

http://cran.nexr.com/	NexR Corporation, Seoul
http://healthstat.snu.ac.kr/CRAN/	Graduate School of Public Health, Seoul National University, Seoul
http://cran.biodisk.org/	The Genome Institute of UNIST (Ulsan National Institute of Science and Technology)

추가 데이터 시각화 언어 R

R 개요

- 1976년 발표된 S, 1988년 발표된 S-Plus를 기반으로 하여 1993년 오픈소스 라이선스 R 언어 발표
- 주요 특징
 - ✓ 인터프리터 방식 스크립트 언어
 - ✓ 데이터 분석기능이 요구되는 시스템 통합 용이 (기타 개발언어, Excel, DB 인터페이스(ODBC), 개발 환경)
 - ✓ 데이터 분석에 적합한 자료구조 지원
 - ✓ 다양한 단계의 통계 계산 및 분석 기능
 - ✓ 시각화 그래픽 지원
- R언어 발전 배경
 - ✓ 빅데이터 정보처리 언어로 대중적 인식
 - ✓ 향후 유망 직업으로 데이터 과학자 (Data Scientist), R은 데이터 과학 언어로 소개
 - ✓ 전 세계 사용자, 개발자 급격히 확대, 4000개 이상의 무료 오픈소스 패키지 공개
 - ✓ 상업적 분석 도구나 sw 기능 대체 가능성
 - ✓ 인터넷에 공개된 소개서, 설명서, 사용법 등 풍부한 R언어 자료
 - ✓ 기존에 개발언어와 유사하지만 쉬운 난이도(여러 가지 평가 존재)와 독특한 문법 체계
- 단점 (제한사항)
 - ✓ 메모리 로딩 방식으로 TB급 이상 정보처리는 부담
 - ✓ 불필요한 데이터 저장으로 메모리 소모 증가
 - ✓ R 2버전 단일 코어 사용