캡스톤 디자인 프로젝트

IRMI

Interactive Real-time Motion Infographics

최예인

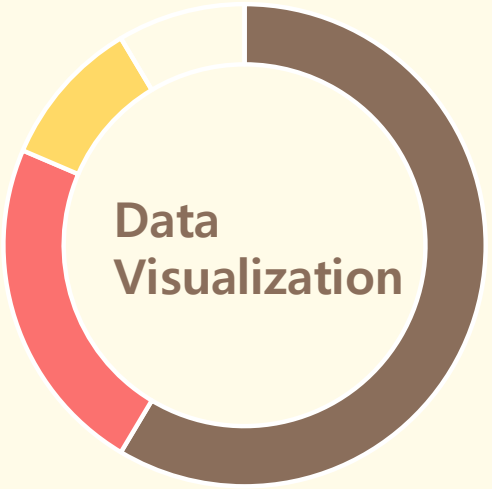
뉴로IRMI

컴퓨터공학부

20143108

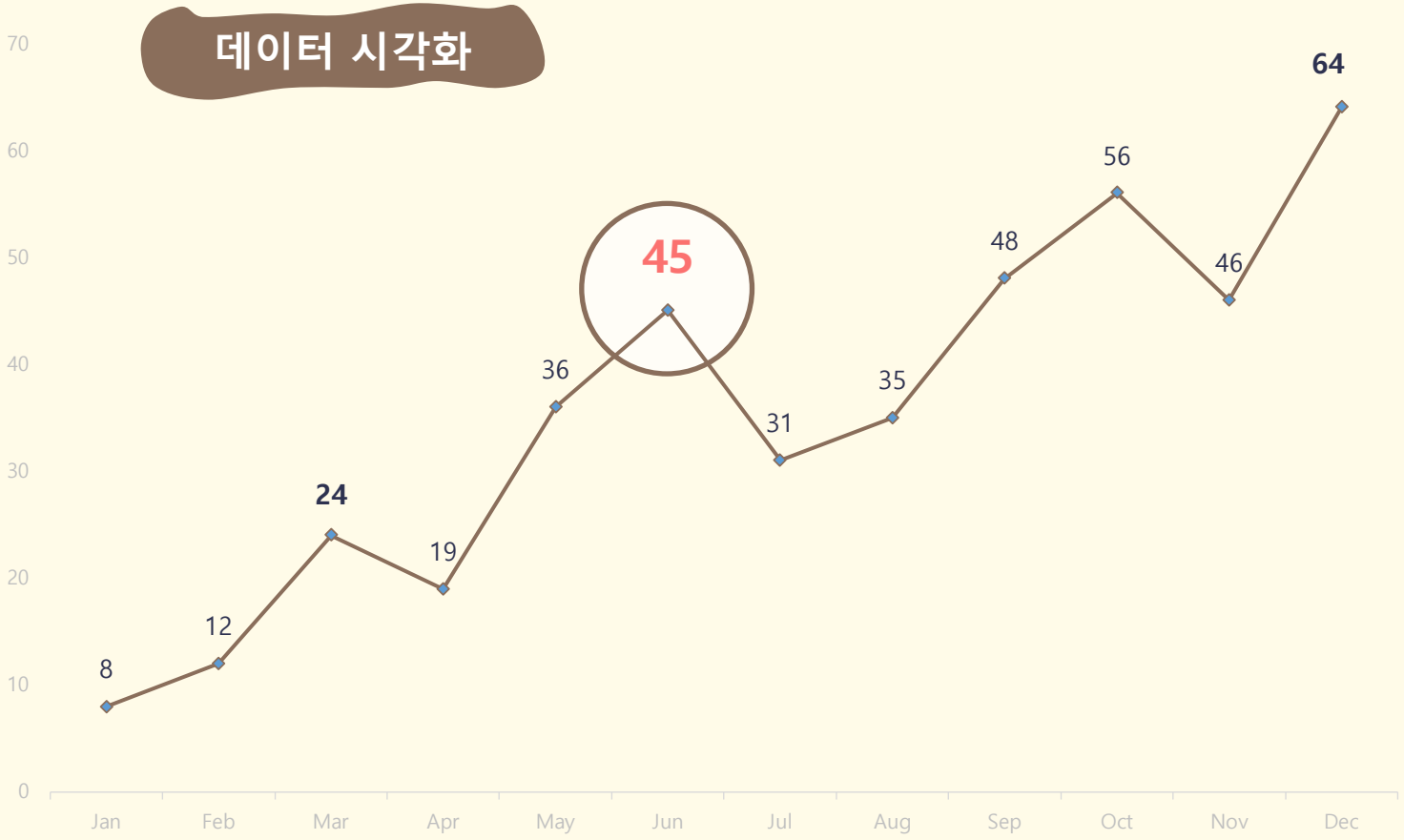
27조

목차



데이터 시각화 란?

-> 데이터 분석 결과를 쉽게 이해할 수 있도록 시각적으로 표현하고 전달하는 것.



20대 성인 대학생 남녀 각 100명 대상
– 2017. 12. 31

IRMI

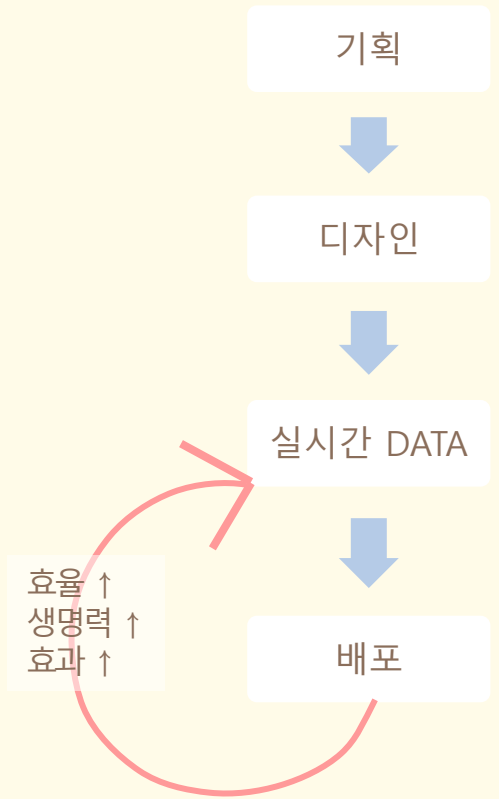
Interactive Real-time Motion Infographics

실시간 인포그래픽 제작 방식

JavaScript Web Framework

<기존의 정보디자인 제작 방식>

<실시간 인포그래픽 제작 방식 IRMI>

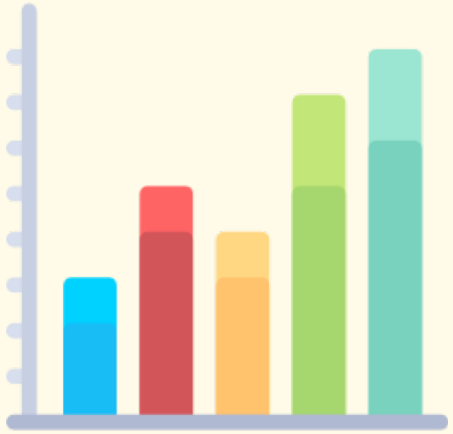


02 개발 목표

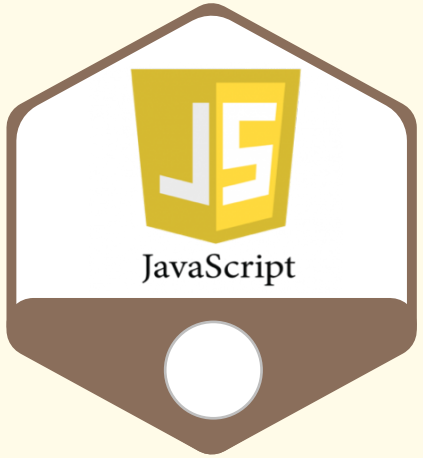
데이터 시각화 라이브러리

Chart.js

Data-Driven Documents



03 개발 내용



JavaScript

실시간 지하철 승하차 인원

(JavaScript만을 이용한 역별, 시간대별, 호선별 데이터 시각화물)

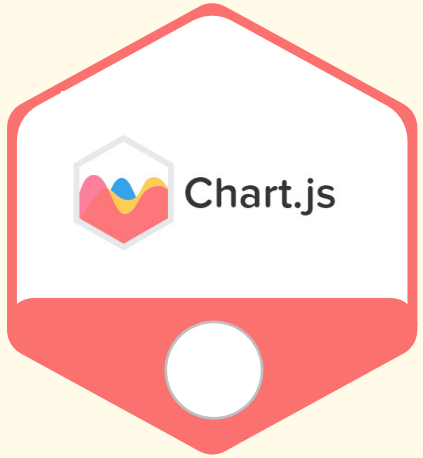


Chart.js

8가지 차트 구현

(Bar chart, Doughnut chart, Grouped bar chart, Horizontal bar chart, Line chart, Mixed chart, Pie chart, Radar chart)

Google Charts

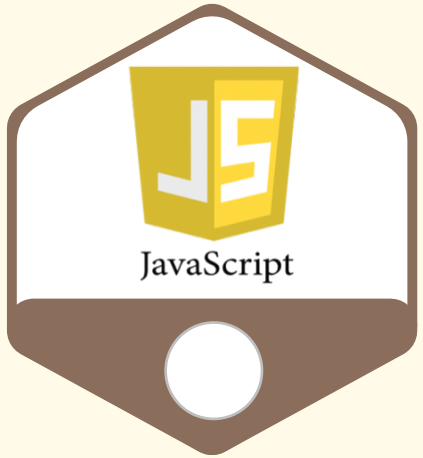
4가지 차트 구현

(Bar chart, Grouped bar chart, Horizontal bar chart, Pie chart)

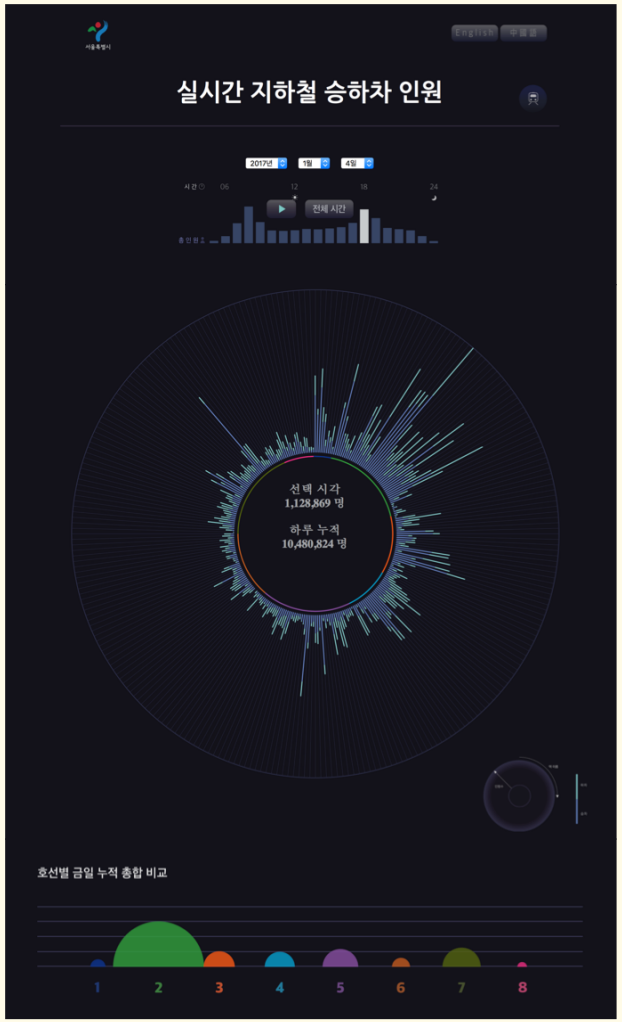
D3.js

5가지 차트 구현

(Bar chart, Doughnut chart, Horizontal bar chart, Pie chart, Doughnut chart)



JavaScript



< JavaScript만을 이용한 지하철 승하차 인원 시각화 >

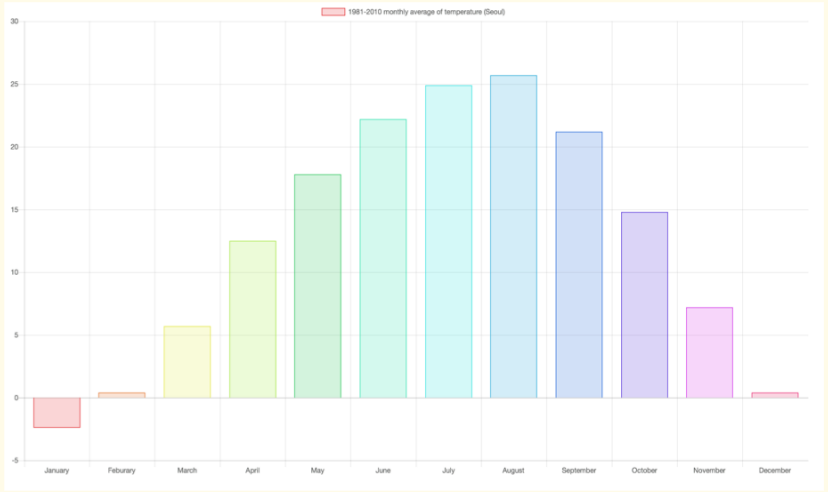
- 웹 상에 자바스크립트를 이용하여 2017년도 지하철 승하차 인원에 대한 데이터를 받아와 시각화 시킨 결과물

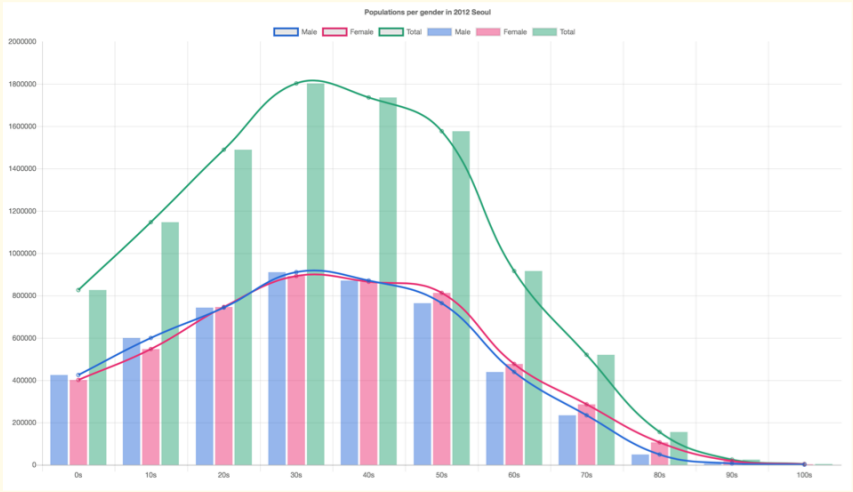
- 시간대별, 호선별, 역별로 구분하여 표시가 됨.

- 각 시간을 선택하면 그 시간에 해당하는 데이터만 표시가 됨.

- 하루 동안 각 역별 누적 승하차 인원 또한 표시 됨.

Chart.js

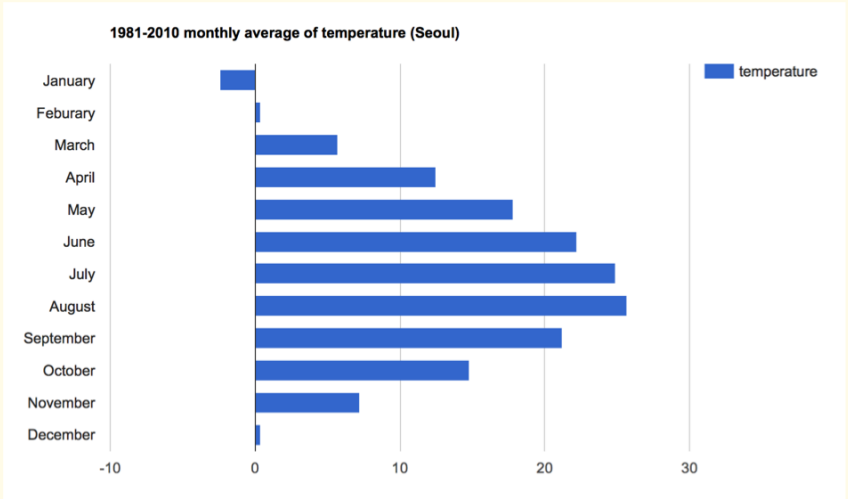


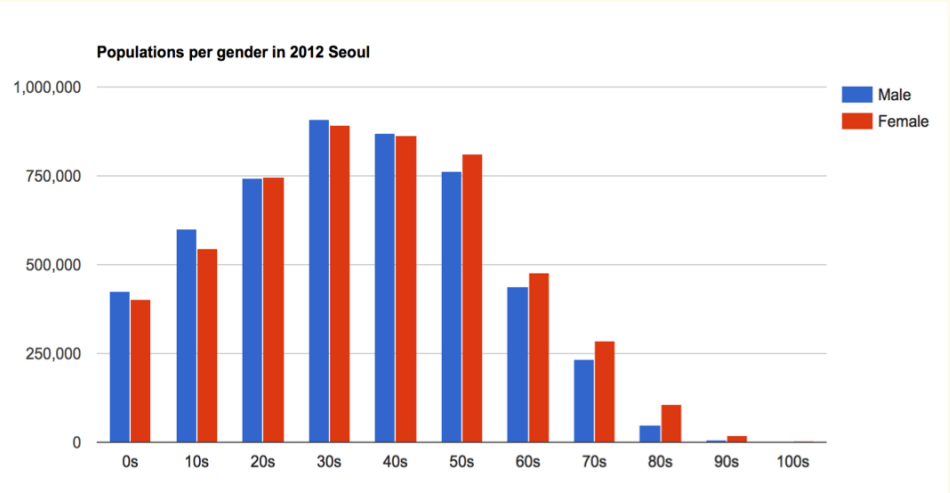


< Chart.js를 이용한 8가지 종류의 차트 >

- HTML5 Canvas 방식의 오픈소스 라이브러리
- Bar chart, Doughnut chart, Grouped bar chart, Horizontal bar chart, Line chart, Mixed chart, Pie chart, Radar chart
- Mouseover, Mouseclick 등 기본적인 간단한 브라우저 이벤트 지원
- HTML 색상코드 그대로 사용가능
- Type에 어떤 차트를 사용할 것인지 명시해주면 자동으로 차트 생성

Google Charts





< Google Charts를 이용한 차트 >

- 구글에서 제공하는 그래프 구현 서비스, 별도의 플러그인 설치없이 사용가능.

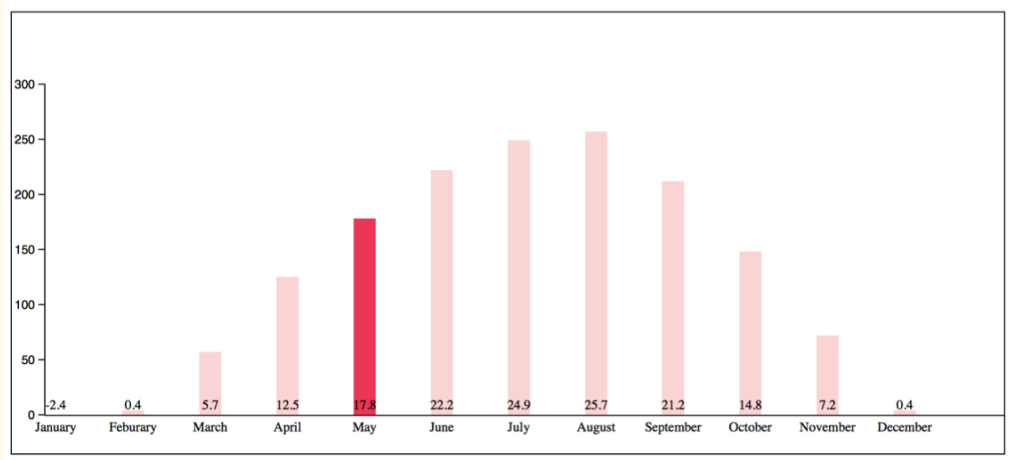
- Geo chart, Scatter chart, Column chart, Histogram, Pie chart, Tree map, Gauge, Timeline, Org chart, Bubble chart 등 다양한 차트구성

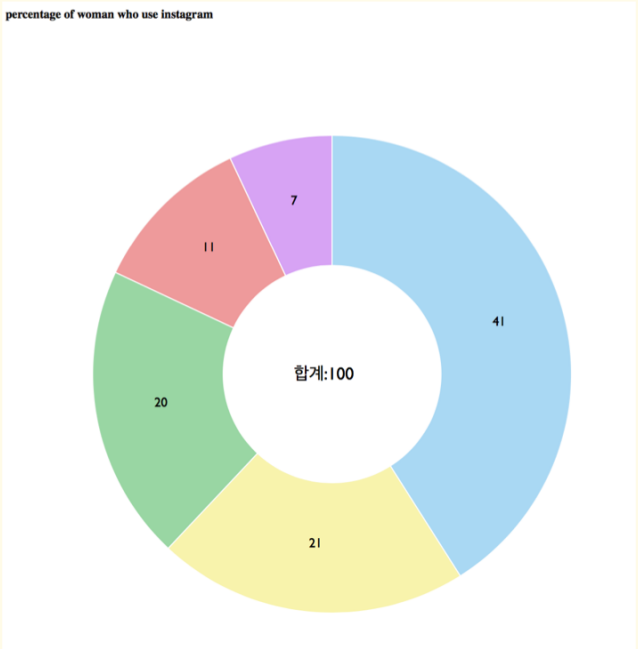
- 크게 2가지 구조로 나뉨. 차트를 생성하는 함수 정의, 나타나고자 하는 데이터 값 셋팅 및 옵션 설정.

- 데이터 구성을 위하여 그래프 구축 시 테이블 형태의 자료구조를 지원.

03 개발 내용

D3.js





< D3.js를 이용한 차트 >

- D3.js (Data-Driven Documents)

- 웹 브라우저 상에서 동적이고 인터렉티브한 시각화를 위한 JavaScript 라이브러리.

- 데이터 양이 방대할 경우에는 XML, CSV, JSON 등 다양한 데이터 형식을 취급.
- jQuery 사용법과 유사하면 DOM 제어 가능.
- 디자이너가 원하는 거의 모든 디자인 구현 가능.

기존의 데이터 시각화 라이브러리 비교, 분석 보고서

기존의 데이터 시각화 방법 비교 및 분석 보고서

1. 개요

현재 데이터 시각화를 위한 여러가지 방법이 존재하지만, 각각의 방법마다 한계가 있다. 따라서 이를 보완하기 위한 데이터 시각화용 자바스크립트 프레임워크 (IRMI.js)개발에 도움이 될 수 있도록, 기존에 있는 여러 데이터 시각화 방법들을 이용하여 여러 차트들을 만들어보고, 그 경험을 통해 비교, 분석을 위한 보고서를 작성한다.

A. 기존의 데이터 시각화 방법

i. JavaScript
ii. Google Charts
iii. Chart.js
iv. D3.js
v. 그 외 기타 등.. (Chartist.js, Dc.js, Plotly.js, Techan]S, Cola.js, VivaGraph 등 여러 종류가 있다.)

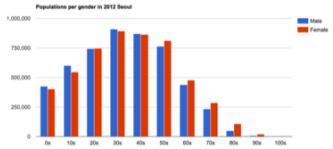
2. 기존의 데이터 시각화 방법 비교, 분석

- 기본적으로 데이터 시각화를 하는 과정에서는 데이터를 차트로 표현하는 경우가 대부분이다. 순수 자바스크립트만 사용하여 차트를 구현하려면 코드가 굉장히 길어지기 때문에, 이를 도와주기 위한 여러 라이브러리가 존재한다. 기본적으로 자바스크립트 라이브러리는 SVG 태그를 이용하는 것과 CANVAS 태그를 이용하는 것으로 나뉜다.

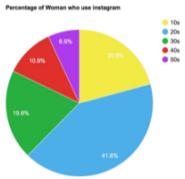
	SVG	CANVAS
장점	1) 벡터 그래픽 포맷으로 해상도에 독립적. (그리기 프로그램과 같다고 볼 수 있다.) 2) 높은 수준의 애니메이션 지원 O 3) 자바스크립트로 SVG DOM API 를 이용한 모든 요소를 자유자재로 컨트롤. 4) XML 파일 포맷이므로 모든 웹 브라우저에 지원, 웹 어플리케이션 UI 에 좀 더 좋은 솔루션으로 이용.	1) 픽셀 수준의 처리에 용이함. (래인트 프로그램과 같다고 볼 수 있다.) 2) .PNG 또는 .JPG 파일로 저장 가능. 3) 게임이나 도형 처리, 이미지 편집과 같이 작업을 할 때 유리.
단점	1) 문서 복잡도가 증가할수록 (DOM 이 많을수록) 느리다. 2) 게임을 만들때에는 부적합.	1) DOM 노드가 없고, 픽셀로만 컨트롤 가능. 2) 애니메이션에 대한 API 가 없어서 필요할때마다 이벤트를 업데이트하거나 타이머를 작동시켜야함. 3) 지원되지 않는 브라우저가 있음.

A. JavaScript 만으로 구현

- 자바스크립트는 웹을 위한 객체 기반의 스크립트 프로그래밍 언어이다. 주로 웹 페이지에 기능을 더해 HTML 웹 페이지를 동적이고 살아있게 만드는 기능을 한다. HTML 요소와 콘텐츠를 추가하고 제거할 수 있으며 여러 이벤트를 다룬다.
- 데이터 시각화를 위한 여러 라이브러리가 등장하기전에는 순수 자바스크립트만으로 여러 차트들을 구현해야 했다. 하지만 단순한 그래프 하나를 만들기 위해서도 불필요하게 많은 양의 코드가 필요했기 때문에 차차 여러 라이브러리가 등장하기 시작했다.
- 자바스크립트는 그래프나 차트를 위해 만들어진 언어가 아니라 웹 페이지 자체의 동적인 효과나 이벤트 등을 위해 만들어진 언어이기 때문에, 데이터 시각화를 하는데 있어 가장 널리 사용되는 차트를 만든다 할지라도, 차트로 인식을 하는게 아니라 하나의 도형으로 인식을 한다. 따라서 막대그래프의 선 하나하나를 하나의



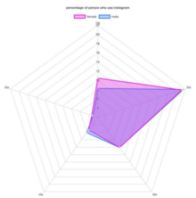
(Google Charts 를 이용한 Grouped bar chart 1)



(Google Charts 를 이용한 pie chart 1)

✔ 장점

1) 차트를 하나하나 직접 구현하는 것이 아니라, 차트의 이름만 지정해주면 해당 차트가 자동으로 구현된다.
2) 다른 차트 라이브러리에 비해 종류가 다양하다.
3) 데이터를 받아오는 부분, 색상 등 여러 옵션을 지정하는 부분, 차트를 불러오는 부분의 구분이 명확해서 처음 접하는 사람도 쉽게 사용할 수 있다.



(Chart.js 를 이용한 radar chart 1)

✔ 장점

1) 보통 예제 코드를 실행해보고 간단하게 바꾸어 보려면, 오픈소스의 메뉴얼이나 레퍼런스를 봐야하지만, chart.js 경우 따로 보지않아도 이 코드가 무슨 기능을 하는지 알기 쉽게 되어있어 초보자도 쉽게 접근할 수 있다.
2) type 에 무슨 차트를 사용할 건지만 명시해준다면, 차트를 직접 그리지 않아도 자동으로 차트가 생성된다.
3) 축은 입력된 데이터의 값에 맞춰서 자동으로 그려지며, 각 데이터의 라벨 또한 지정만 해주면 데이터 바로 밑에 자동으로 생성된다.
4) 다양한 색상 적용이 가능하다. Html5 색상코드를 그대로 사용가능하며, rgba 형태의 색상도 제공한다. 디자인적인 측면에서 다른 라이브러리에 비하여 색감이 예쁘게 나오는 편이다.
5) 각 데이터가 의미하는게 무엇인지 title 밑에 자동으로 명시해준다.
6) 따로 자바스크립트 이벤트 코드를 넣어주지 않아도, 마우스를 각 컨텐츠에 올리면 데이터 값을 표시해준다.

✔ 단점

1) 차트의 종류가 너무 제한적이다.

D. D3.js

- D3 란 Data-Driven Documents (데이터 기반 문서) 를 의미하는 단어로, 데이터에 기반을 둔 문서를 다룰 수 있는, 웹 브라우저 상에서 동적이고 인터랙티브한 시각화를 위한 자바스크립트 라이브러리이며, HTML, CSS, SVG 를 사용하여 데이터를 시각적인 결과물로 나타낸다.
- 자바스크립트 변수에 직접 데이터를 할당하여 사용할 수도 있지만, 데이터의 양이 방대할 때에는 XML, CSV, JSON 등 다양한 데이터 형식을 취급한다.
- jQuery 사용법과 매우 유사하며, 돔(DOM)을 제어할 수 있다.

```
d3.select("body").selectAll("div")
```

- 처음 d3.js 를 시작할 때, 위와 같이 html 파일에 선언된 body 문서요소를 선택하고 그 내부의 모든 div 를 선택하는 코드를 많이 볼 수 있는데, 실제 html 파일을 보면 body 만 선언해 주었을 뿐, div 는 어디에도 선언되지 않았다. 이것은 가상의 문서요소에 'data(dataSet)' 을 이용하여 우리가 정의하는 데이터를 바인딩 하고, 'enter()' 함수로 dataSet 배열의 각 요소를 순화하며 가상의 문서요소: div 를 만든 후 마지막으로 'append('div')' 함수를 통해 가상의 문서요소를 body 요소의 하위로 추가를 해주면 데이터를 기반으로 DOM 이 만들어진다. (화면상에서는 아무런 변화가 없지만 개발자도구로 보면 문서 요소가 추가된 것을 확인할 수 있다.)

(D3.js 를 이용한 pie chart 1)

Irmi 초기 프로토타입

```
<div class="irmi-ground" irmi-id="chart_00" irmi-source="./testdrive.irmi">
    <canvas id="bar-chart" width="800" height="450"></canvas>
    <script>show_irmi_chart("bar-chart", "irmi_chart_bar.irmi");</script>
</div>

<div class="irmi-ground" irmi-id="chart_01" irmi-source="./testdrive.irmi">
    <canvas id="doughnut-chart" width="800" height="450"></canvas>
    <script>show_irmi_chart("doughnut-chart", "irmi_chart_doughnut.irmi");</script>
</div>
```

```
{
    "type": "bar",
    "data": {
        "labels": ["0s", "10s", "20s", "30s", "40s", "50s", "60s", "70s", "80s", "90s", "100s"],
        "datasets": [
            {
                "label": "Male",
                "type": "line",
                "borderColor": "#296cd8",
                "data": [424685, 599453, 742981, 910258, 870688, 763881, 438938, 234147, 48952, 6357, 996],
                "fill": false
            },
            {
                "label": "Female",
                "type": "line",
                "borderColor": "#e83073",
                "data": [401302, 547056, 745963, 891984, 865091, 812455, 476794, 285892, 106184, 18331, 2930],
                "fill": false
            },
```

Q&A