

Billboard.js 보고서

2013011800 구장희

[D3.js]

- D3는 'Data Driven Document'(데이터 기반 문서)의 약자다. D3.js는 데이터에 기반을 둔 문서를 다룰 수 있는 자바스크립트 기반 라이브러리다. D3.js는 HTML, SVG(Scalable Vector Graphics), CSS를 사용해 데이터를 시각적인 결과물로 나타낸다. D3.js는 최신 표준 기술을 지원하는 웹브라우저에서 구동하며, '인터넷 익스플로러8' 이하 버전에서는 동작하지 않는다.
- 기능
 - 데이터를 불러온다(csv, json 등)
 - 문서 요소에 데이터를 엮고, 필요한 요소를 만들어낸다
 - 각 문서 요소에 엮인 개별 데이터를 토대로 해당 문서요소를 변환
 - 사용자의 입력에 대한 반응으로 문서요소의 상태를 한 값에서 다른 값으로 이동
- 기능 설명
 - 기본적으로 데이터를 불러오고, 이 데이터를 표현하는 데 적합한 HTML 문서 요소를 새로 만들어 데이터를 엮는다.
 - 이후 엮인 데이터를 토대로 해당 문서 요소에 시각적인 값을 부여하는 과정을 거쳐 구현된다. (예시: 막대 그래프라면 데이터를 '막대'에 엮고, 데이터를 막대의 길이 값으로 지정하는 식으로 그린다.
 - 이후 추가 조작이나 사용자와의 상호작용을 통해 문서 요소의 상태가 한 값에서 다른 값으로 변하는 과정이 더해진다.
 - D3.js는 클릭이나 마우스 오버 등의 작용이 있으면 해당 행위에 지정된 값으로 길이나 색상 값의 변화를 계산해서 바꿔주는 방식 등을 사용해서 동적 요소를 구현한다.
- 특징
 - D3.js는 바로 그래프를 구현할 수 있는 도구는 아니다. 태블로나 엑셀처럼 직관적으로 그래프 종류를 선택할 수 있는 것도 아니고, R의 'ggplot2' 패키지처럼 해당 데이터를 기반으로 그릴 수 있는 그래프의 기본 형태(막대, 스택, 분포도, 꺾은 선 그래프 등)를 제공하지도 않는다.
 - D3.js는 코딩을 통해 SVG 등의 시각적인 요소를 활용해 직접 그림을 그려 데이터를 시각화하는 방식이다. 텍스트 편집기 등을 이용해 코드를 작성해야 하므로 코딩을 전혀 배우지 않은 비전공자가 활용하기는 어렵다.
 - 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.

[배경]

- 그간 네이버에서의 차트
 - 서비스 마다 다른 라이브러리 사용
 - 기술적 노하우가 축적이 되지 않음
 - 매 개발마다 반복되는 리소스의 낭비
- 처음엔 자체 개발
 - 유지보수가 잘 되지 않음(개발자의 이직 및 다른 서비스 개발로 인해)
 - 타 라이브러리 대비 범용성이 부족
- 오픈소스를 선택
 - 지속적 업데이트 & 기술적 트렌드 반영, 안정성 기대 가능
 - fork를 통한 자체적인 유지 가능
 - 공통된 라이브러리 사용 -> 기술 경험 누적 가능
- C3.js 선정
 - 가장 인기있는 D3.js 기반 차트 라이브러리
 - popularity : 깃허브 스타, 서드파티 앱, 스택오버플로 질문수 등등
 - 간결한 인터페이스
 - 풍부한 레퍼런스
 - 네이버 서비스에 이미 많이 사용
 - 엔드 유저 대상이기 때문에 적합
- 난관
 - 커뮤니케이션 : 차트 내의 명칭에 대한 모호함
 - 디자인 가이드
 - SVG TEXT : 줄바꿈은 새로운 노드로 / 위치,여백등의 조정이 어려움
 - 모바일 환경 : C3.js는 모바일 환경을 지원하지 않음 (swipe 제스처를 통한 데이터 확인 UX필요)
 - 환경별 다른 이벤트 발생(iOS 11 과 Android 7)
 - y 축 기반의 따른 up/down 표현의 모호함
 - 데이터가 0인경우 표현이 되어야 하는지 안되어야 하는지
 - -> C3+(C3.js 확장 라이브러리)를 만들다.
- C3+.js 를 만들다. (C3.js를 확장한 테마 형태의 디자인 차트 생성)
 - 커스텀 축 지원
 - 범례 템플릿
 - 모바일 지원
 - 테마를 통한 차트 생성
 - 확장 옵션
 - 결과
 - 블로그/포스트 통계에 적용
 - 장점
 - 매번 다른 기술/라이브러리를 다루는 반복적인 비용 제거
 - 기본적 디자인(테마) 를 활용해 커스터마이징(디자인) 에 따른 비용 제거
 - 기술적 경험 축적(SVG , D3 , C3.js)
 - 현실적 문제
 - 장기적 관점에서 C3+ 발전을 위한 외부 공개 but
 - 래퍼/애드온 형태의 지속적 발전과 효용성 의문

- 기반 라이브러리 C3.js의 지속성 의문
- C3.js 의 더딘 발전으로 인해 , 오픈소스의 발전에 기댈 수 있을것이란 기대는 위기 직면

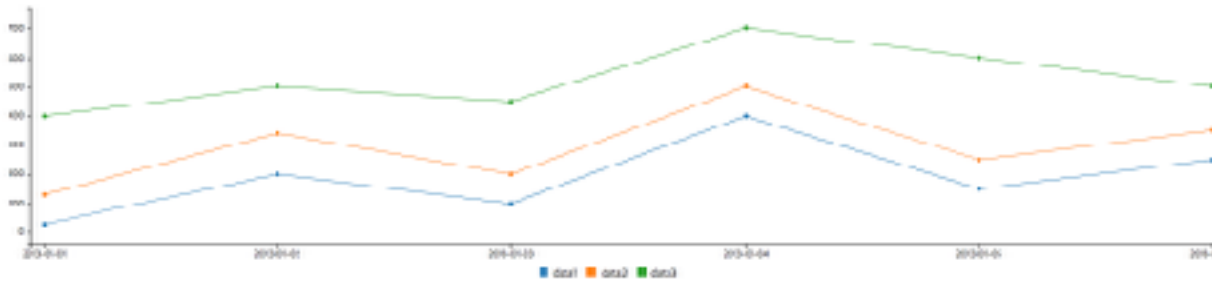
[Billboard.js]

- Billboard.js(re-usable , easy-interface tsp chart library based on D3 v4+.)
 - 커스터마이징
 - 150개 이상의 다양한 옵션 제공
 - SVG 노드 : 필요한 경우 , 직접 핸들링 가능
 - CSS로 스타일링 가능
 - 홍보전략
 - 다수의 ECHO 사이트 등록
 - 뉴스레터 소개
 - 유력 매체 소개
 - 결과
 - 깃허브 트렌딩 js 언어부분 3위 기록
 - 깃허브 스타 2주간 1000개 도달
 - third-party app의 등장
 - Angular , React , R , Web Components 등
 - 지속적 성장 : npm 다운로드 수
 - 충실한 문서 작성
 - C3.js에서 migration 가이드
 - API 문서
 - 왜 Fork하게 되었는지
 - README
 - 풍부한 예제 : 무엇이 가능한지, 가능 할 수 있는지를 보여준다.
 - PLAYGROUNG : 온라인에서 바로 옵션들을 수정하고 확인
- 개발 과정
 - C3.js 프로젝트 참여 시도
 - 원 개발자 및 커미터에게 메일을 통한 문의
 - PR 보내고
 - 이슈들에 대한 답변
 - 메일 회신이 없고 , 액티비티도 없는 상태
 - issue 등록으로 프로젝트 유지 문의 -> 지속적 업데이트 어렵다는 답변
 - fork (명제 : 오픈소스 업데이트 지속 안될 시 fork를 통한 유지)
 - 당면한 C3.js 미해결 과제들
 - D3 최신버전 v4+ 미지원
 - 모바일 환경 대한 지원 부족
 - 오래된 개발 스타일 코드(ES3)
 - SVG polypill 제거 등등
 - 합리성 , 당위성 & 신뢰
 - fork 한다고 해서 사용자가 오는것은 아니다.
 - 기존 커뮤니티에 당위성 제시 필요
- D3 v3 -> v4
 - breaking change

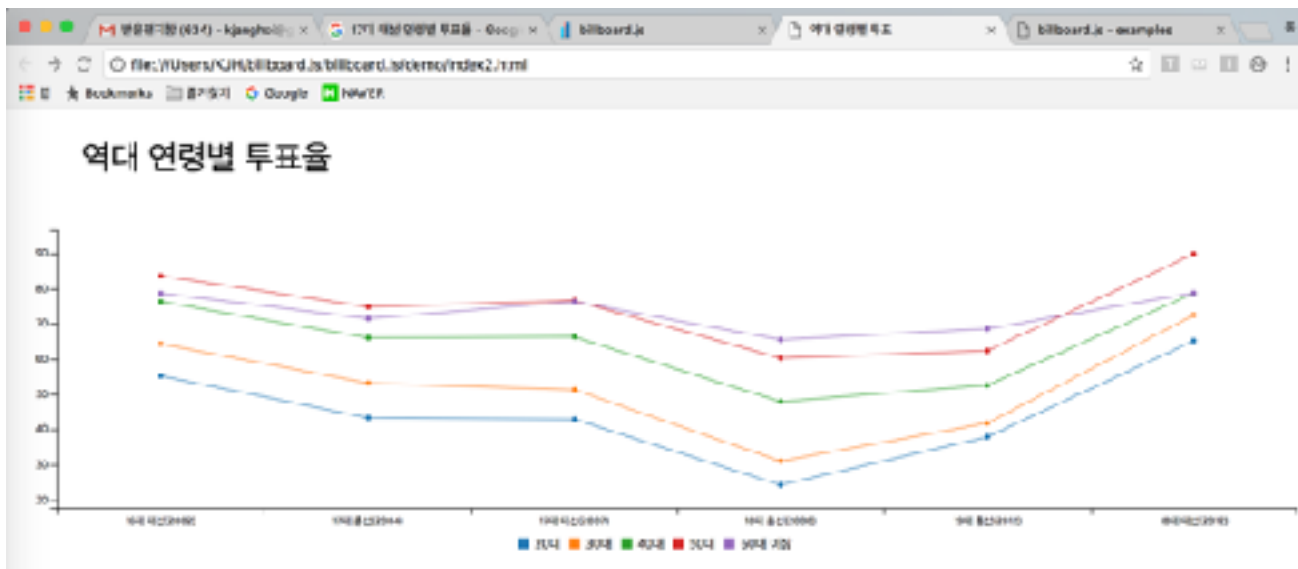
- 공식문서는 있으나 migration 만들 계획 없음.
- so,
 - 변경된 모듈들을 비교하고 테스트
 - 차트 생성 흐름에 따른 오류들의 순차적 해결/변환
 - ES3 -> ES6 의 전환 병행 및 개발 환경 변경
 - API의 문서화(JSDoc)
 - 테스트 코드 업데이트 및 커버리지 개선
- 릴리즈 직전 , 차기 릴리즈 계획과 새로운 커미터의 추가
 - 커미터 추가 후에도 활발한 활동이 없어 계획대로 릴리즈
 - billboard : 의미는 다르지만 차트를 연관. 오랫동안 친숙한 이름
 - 2017/06/08 v1.0.0 공개

[예제-TimeseriesChart이용, 데이터: 역대 연령별 투표율]

[TimeseriesChart Example]



[시연 - 역대 연령별 투표율(코드 첨부)]



[실행 법]

code/tmp/demo/index2.html 실행

[code]

code/tmp/demo/script3.js