

[ 목 차 ]

* 과제를 시작하면서
* 과제 개요
* 과제 내용
* 과제를 마무리하면서

학과 : 소프트웨어전공

학번 : 2012003073

이름 : 박 경세

* **과제를 시작하면서**

우선, 과제에 대한 리포트를 작성하기에 앞서 이번 과제를 시작하면서 느낀 점에 대해 간단하게 설명 드리고 시작하도록 하겠습니다.

사실 저는 Open-Source란 것에 대해서 이전에 많이 들어보긴 했지만 직접 사용해본적도, 사용하는 것을 본적도 없습니다. 대부분의 학생들이 그럴 것 같습니다.

또한, 저는 주로 학교에서 배우는 이론적인 내용이나, 이론적 내용 기반 과제 위주로만 공부해왔고 그리고 관심 있는 분야가 Linux Kernel, Embedded System Programming분야 쪽이라 Front-End에서 개발해본 경험이 거의 없습니다.

다행히 이번 학기를 시작하면서 해커톤에 경험을 위해 참가해보았는데, 그 때 Java -script가 무엇이고 어떤 것인지에 대해서는 대충 감을 얻을 수 있었습니다.

이번 해커톤의 경험을 통해 앞으로 “Front-End, Back-End가리지 않고 많은 개발경험을 쌓아보자!” 라고 생각했는데 마침 이번 소프트웨어스튜디오2과목을 수강하게 되었고 또, 이번 과제를 통해 조금이나마 접할 수 있게 되어 많은 배움을 얻을 수 있을 것이라 생각했습니다.

그래서 전체적인 js, css, html의 구성이나 모든 것이 아직은 미숙해 보일 수 있지만 공부를 하면서, 나름대로 최대한 많은 예제를 접해보고 작성한 코드이므로 귀엽게 봐주셨으면 좋겠습니다 ☺

* 과제 개요

저는 학교포탈홈페이지에 학기 별 성적 조회를 차트로 만들어보고자 하였습니다.

매번 학기가 끝날 때나, 기업에 성적에 관련된 사항을 입력하거나, 성적 조회를 위해서 자주들어가보곤 하는 곳입니다.

여기에는 모든 정보가 표로 정리되어 있고, 숫자로 명시되어 있어 사실 한눈에 쉽게 눈에 들어오지는 않습니다.

그래서 한눈에 쉽게 들어오도록 Billboard.js를 통해 차트로 만들어보면 좋을 것 같다고 생각하였고, 이번 과제를 위해 같은 형식의 차트를 **D3, C3, Billboard.js**로 작성해보기로 하였습니다.

이렇게 함으로써 각각의 차이점에 대해 학습하고, 어떤 발전이 있었는지를 Report로 작성해보면 배울 점도 있고, 사용법을 익히는 데에 좋을 것 같다고 생각하여 과제를 진행하였습니다.

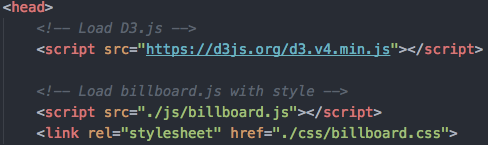
* 과제 내용

우선 우리의 과제의 중심인 billboard.js에 대해서 공부를 시작하였습니다.

Billboard.js는 naver에서 C3.js를 fork하여 만든 새로운 차트 라이브러리입니다.

D3 -> C3 -> billboard의 과정에 따라 작성하며 올라오는 방식으로 진행하려 했었는데, 차라리 billboard.js에 대한 이해도를 높여 차근차근 진행하고, 아래로 내려가는 방식을 선택하였습니다.

우선 java script를 사용하기 위해 html파일의 hea에 사용을 명시해주어야 했습니다.



저에게는 js, css모두 생소했지만 간편한 사용을 위해서 D3, C3, billboard의 압축파일을 모두 다운받고 workspace안에 css, js폴더를 만들어 분류해 놓았습니다.

그러고 나서 해당 폴더에 있는 js, css의 경로를 맞게 설정해주니 setting은 끝났습니다.

우선 html의 body부분에 간단하게 내용을 구성하기 위한 제목, 전공, 학번, 이름을 나타내어 홈페이지의 구성을 갖추었습니다.

그리고 이제 어떤 Chart를 출력할 지에 대한 고민이 필요했습니다.

그래서 저는 2가지 차트를 출력하기로 했습니다.

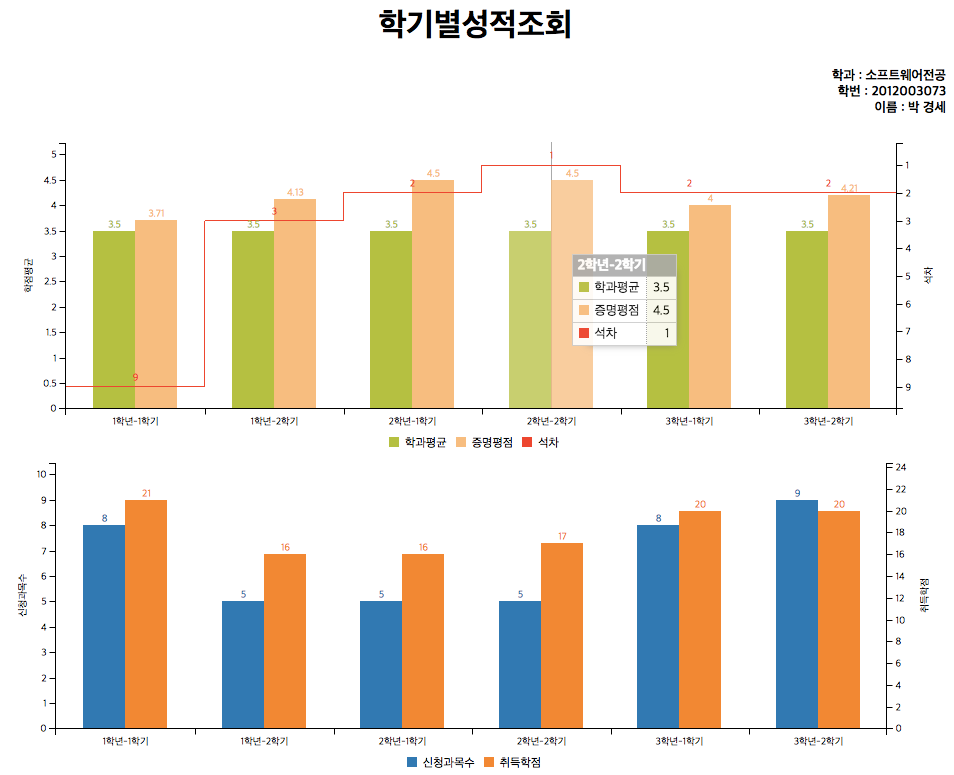
**첫번째, 학기 별 학과의 평균 학점과 학생 본인의 증명 평점에 대한 비교 막대 그래프**

**두번쨰, 학기 별 신청과목수 / 취득학점에 대한 그래프를 출력해주기로 했습니다.**

이제 어떤 그래프를 만들지에 대한 계획이 끝났으므로 실제 그래프 구현을 위해 예제를 학습하였습니다. Billboard.js에 대한 매우 다양한 예제들이 준비되어 있었고 학습을 하기에 정말 최적화 되어있었습니다.

막대그래프, 선형그래프, stack, 꺽은선 등의 다양한 그래프 형식과 시간에 따라 다른 그래프 형태로 변형도 할 수 있고, 새로운 데이터를 시간차로 추가한다 던지 정말 다양한 기능들을 학습할 수 있었습니다.

또한, y축을 2중으로 추가 한다 던지, x축 범위 설정, 범례 설정 등 차트에 필요한 각 기능들을 반복학습 하면서 제가 만들 차트에 필요한 기능들을 숙지하였습니다.



그 결과 위에 보이는것과 같이 깔끔한 그래프를 얻을 수 있었습니다. **첫번째 그래프**는 위에서 설명했듯 학과 평균과 학생의 증명 평점을 학기 별로 비교해주는 그래프입니다.

**두번째 그래프**는 처음에는 신청과목수에 대한 그래프만 나오다가, 3초 뒤에 신청 과목에 대한 총 취득 학점의 데이터를 추가하는 방식으로 시간차 그래프를 구현하였습니다.

또한, 새로운 기능을 추가해보았는데 **두번쨰 그래프**를 6초가 지난 후에는 도넛형식 그래프로 바꾸어 아래에 띄워보도록 하였습니다.

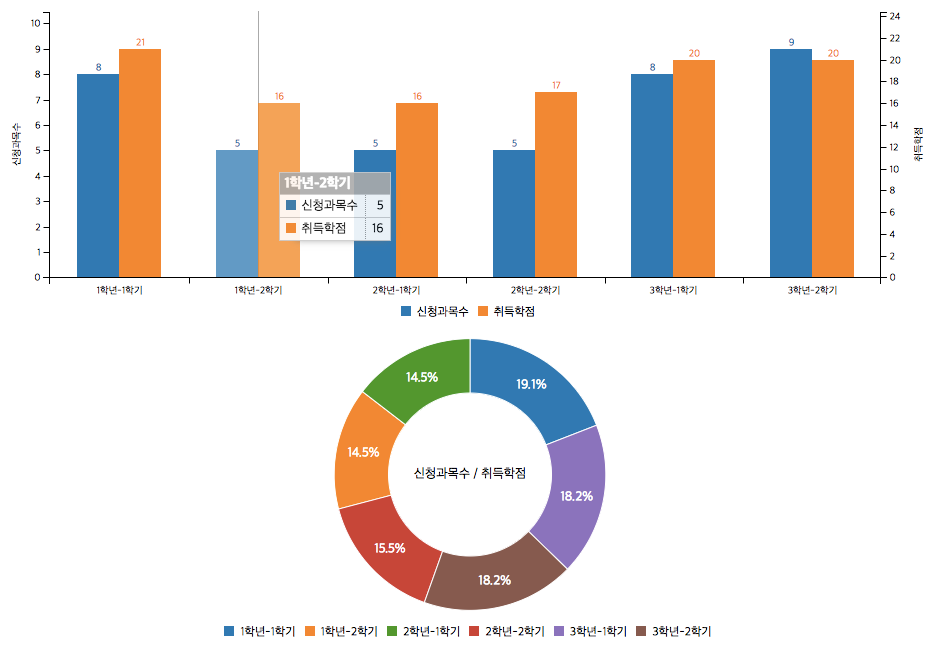


정말 기존의 숫자와 글자의 표로만 나타나 있던 정보들을 차트로 보게 되니 시각적으로 한눈에 들어오고 훨씬 직관적으로 알아 볼 수 있었습니다.

만약 이러한 기능이 추가된다면 내가 지금 상향곡선을 그리고 있는지를 판단할 수 있게 되고 좀 더 열심히 할 수 있는 동기부여도 가질 수 있지 않을까 생각했습니다.

또한, Billboard.js를 사용하면서 느낀 것은 내가 원하는 기능을 수행하기 위해서 작성해야하는 코드가 정말 직관적이다 라는 것입니다.

어떤 데이터를 넣고 싶은지, type, color등 정말 처음 보는 코드라도 이 코드는 어떤 것을 위한 코드인지를 저도 한눈에 알 수 있었습니다. 그래서 과제를 하면서 좀 더 예쁘게, 신기한 기능들을 하나라도 더 넣어보려고 노력하였습니다.



어떤 차트를 만들게 되었는지, 어떤 구성으로 이루어졌는지에 대한 설명은 이걸로 마치도록 하고, 데이터를 가져오는 방식에 대해 잠깐 설명 드리겠습니다.

처음에는 데이터를 대용량이나, 서버에서 가져오는 방식이라면 JSON Format으로 구성하여 Ajax를 통해 가져오는 방식을 공부해보려고 했습니다.

하지만 D3.js를 제외하고는 굳이 Ajax를 사용하지 않고 간단한 데이터의 경우에는 script자체에 데이터를 그냥 바로 넣어주는 것이 더 간편하고 간단한 테스트에는 적당하다고 생각하였고, D3.js의 경우에는 공부해본 결과 tsv file을 load하기 위해서는 httpserver 돌려 ajax방식으로 데이터를 가져오도록 하였습니다.

**저는 ‘데이터를 어떻게 가져오고, 어떤 방식으로 전달해주냐’와 같은 진짜 Web을 만드는데 집중하기 보다는 과제의 목적인 Billboard.js, C3.js, D3.js를 사용해보고 어떤 라이브러리인지 파악하는게 중요**하다고 생각하여 Billbaord.js, C3.js데이터는 java script파일에 직접 넣어주고 작은 규모의 데이터를 기반으로 테스트용으로 진행하였습니다.

반면에 D3.js의 경우 차트를 하나 그리는데도 엄청난 코드 양과 세세한 부분조차 일일이 다 구현해야 했고, 데이터도 tsv file을 HttpServer를 구동하여 가져오는 방식으로 진행하였습니다.

* **D3.js C3.js, Billboard.js의 차이점에 대해**

**[ D3.js ]**

먼저 D3.js에 대해 공부한 내용과 제가 작성한 내용에 대해 설명 드리겠습니다.

D3.js는 이름부터 Data-Driven Documents로 데이터 기반으로 문서를 처리하는 Java Script 라이브러리입니다. 데이터를 시각화 해주는데 널리 사용되고 있습니다.

즉, Data-Driven Documents이지 Graph / Chart Library가 아니기 때문에 차트를 그리는 데에는 시간과 노력이 많이 필요했습니다.

D3의 경우에는 일일이 세부적인 기능조차 자신이 직접 코드로 작성하여야 하므로

**자유성**이 높습니다.

하지만, **자유성**이 높다는 것은 아주 간단한 이번 제가 만든 차트를 만드는데 있어서도 많은 공부가 필요하고 신경 써야할 부분들이 많았습니다.

왼쪽에 보이는 코드가 D3.js의 코드이고 오른쪽에 보이는 코드가 Billboard.js입니다.

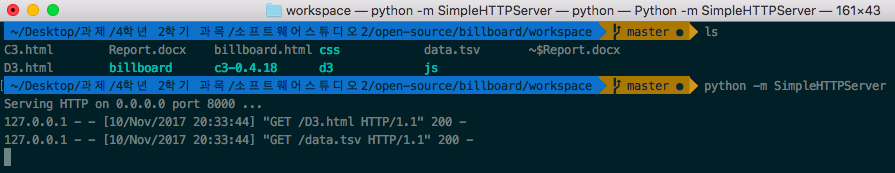
코드에서도 보이듯, 차트 라이브러리와 그렇지 않은 것의 차이는 확연히 보입니다.

일일일 신경 써줘야 할 부분들을 차트 라이브러리에서는 자동으로 지원해주고 있어서 사용자들로 하여금 신경 써줘야 할 부분을 줄여줍니다.

얼마만큼의 막대 사이의 간격을 줄 것인지에서부터, 데이터를 가져와서 막대로 어떤 크기만큼 표현할지를 함수로 구현하기도 하고, 데이터를 가져오기 위한 함수조차 따로 만들어줘야 하는 D3.js였습니다.

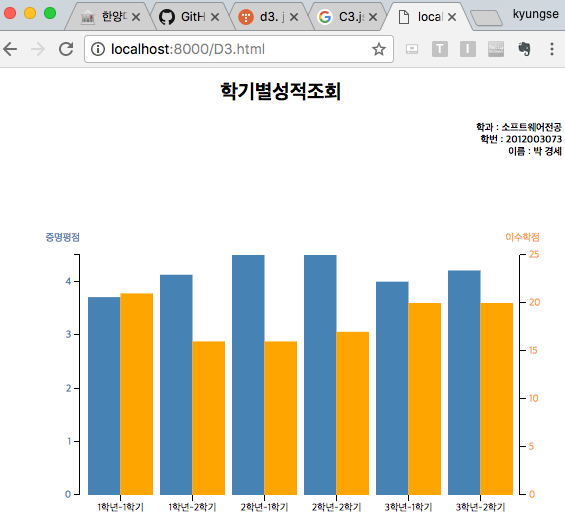
사실 이번 과제에 필요한 코드 중에서 가장 오래 걸리고 어려운 부분이었습니다.

# 코드에 포함된D3.html실행 방법



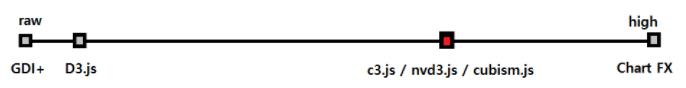
우선, D3.html과 data.tsv가 포함된 폴더로 이동해주고 간단한 서버 실행을 위해

**python –m SimpleHTTPServer**를 입력해주면 local server를 사용할 수 있게 됩니다.



그러고 나서 **localhost:8000/D3.html**로 이동해주면 원하는 차트를 확인할 수 있습니다.

**[ C3.js, Bilboard.js ]**



그림에서 볼 수 있듯, D3.js를 기반으로 나온 것이 C3.js인데 raw-high사이의 간격이 상대적으로 아주 큰 것을 확인할 수 있습니다.

이는 Chart / Graph를 그리기 위한 Library로 특성화 되었다는 것을 뜻하기도 합니다.

좀 더 사용자들에게 쉽고 간편한 기능을 제공하는 쪽으로 발전하였고, 그것을 더 간편하게 개선시킨 것이 Billboard.js라고 할 수 있습니다.