

- 다음주 화요일까지 해야 할 과제는 크게 두 가지입니다
- 과제 A. Django REST 를 이용한 채무관계 기록 백엔드 서버 개발
- 과제 B. HTML / Javascript 를 사용한 간단한 웹 페이지 작성

A. Django REST 과제 – 채무관계 기록 백엔드 개발

사람 간의 채무관계를 기록하는 백엔드 서버를 Django REST로 구현해 본다.

1) 설명

하나의 채무 관계는 <식별자, 생성날짜, 금액, 빌린 이, 빌려준 이> 로 나타낸다. JSON 양식은 아래와 같다.

```
{
  "id": 고유 아이디 (숫자)
  "created": 생성 날짜 (DateTime 타입, 채무 관계 생성시 자동으로 채워지도록 설정)
  "amount": 금액 (32비트 범위 내의 부호 없는 정수)
  "borrower": 빌린 이 (유저의 id를 Foreign key로 저장)
  "lender": 빌려준 이 (유저의 id를 Foreign key로 저장)
}
```

참고로, "id"는 django에서 model을 정의할 때 굳이 명시하지 않아도 자동으로 생성해준다.

빌린 이와 빌려준 이를 표현할 때에는 Django에서 기본적으로 제공하는 User 클래스를 활용한다. Django REST tutorial 4 (<http://www.django-rest-framework.org/tutorial/4-authentication-and-permissions/>)를 참조하자.

View를 구현해야 하는 url은 총 6가지이다. 각 url에서 어떤 기능을 구현해야 하는지는 아래 테이블과 같다. ('-' 표시는 구현하지 않아도 된다는 내용이다) 임의의 인증되지 않은 유저가 접근해서 새로운 채무관계를 추가할 수 있으며, 인증되지 않은 유저가 기존의 채무관계를 수정 및 삭제할 수 있다. `users/` 와 `usersum/` 이 반환해야 할 정보는 꼴이 다소 다른데, `users/`에서 나열할 유저 정보는 세부적으로 어떤 채무를 가지고 있는지를 명시해야 하고, `usersum/`에서 나열할 유저 정보는 자신이 갚아야 할 총 금액 및 받아야 할 총 금액을 출력해야 한다. 정확한 정보는 문서 뒤의 스크린샷을 참조하자.

	GET	POST	PUT	DELETE
debts/	전체 채무 관계 목록 반환	Amount, borrower, lender 를 받아서 새로운 채무 관계 생성	-	-
debts/id/	식별자가 id인 채무 관계 반환	-	amount, borrower, lender를 받아서 채무 관계의 내용을 수정	채무 관계 삭제
users/	전체 유저 목록 반환 (문서 뒤 스크린샷 참조)	-	-	-

users/id/	식별자가 id인 유저 반환 (문서 뒤 스크린샷 참조)	-	-	-
usersum/	전체 유저 목록 반환 (문서 뒤 스크린샷 참조)	-	-	-
usersum/id/	식별자가 id인 유저 반환 (문서 뒤 스크린샷 참조)	-	-	-

2) 제출 및 채점

채점은 수업 및 실습 시간에 공지된 바와 같이, 학생들이 제출한 bitbucket 리포지토리를 사용해 자동 채점을 할 것이다. **자동 채점은 3월 29일 수요일 00시 00분부터 시작한다.**

원활한 자동채점을 위해, 초기 환경을 다음과 같이 세팅하자.

```
swpp@ip-172-31-30-158:~$ pwd
/home/swpp (이 경로가 중요하다.)
swpp@ip-172-31-30-158:~$ virtualenv -p python3 env
... (설치중)
Installing setuptools, pkg_resources, pip, wheel...done.
swpp@ip-172-31-30-158:~$ source env/bin/activate
(env) swpp@ip-172-31-30-158:~$ pip3 install django djangoframework
... (설치중)
Successfully installed django-1.10.6 djangoframework-3.6.2
(env) swpp@ip-172-31-30-158:~$ django-admin.py startproject debtpages
(env) swpp@ip-172-31-30-158:~$ cd debtpages/
(env) swpp@ip-172-31-30-158:~/debtpages$ python manage.py startapp debtpages_rest
(env) swpp@ip-172-31-30-158:~/debtpages$ ls
debtpages  debtpages_rest  manage.py
(env) swpp@ip-172-31-30-158:~/debtpages$
```

제출은 자신의 bitbucket 개인 리포지토리를 이용한다. 위의 ~/debtpages 를 아래와 같이 통째로 복사한 뒤, **assn02** 라는 branch 로 push 한다.

```
(env) swpp@ip-172-31-30-158:~/swpp-assn201701$ cp -r ../debtpages/ .
(env) swpp@ip-172-31-30-158:~/swpp-assn201701$ ls
debtpages  ..... (다른파일,ex. mysite)
(env) swpp@ip-172-31-30-158:~/swpp-assn201701$ ls debtpages/
debtpages  debtpages_rest  manage.py  .... (다른 파일)
```

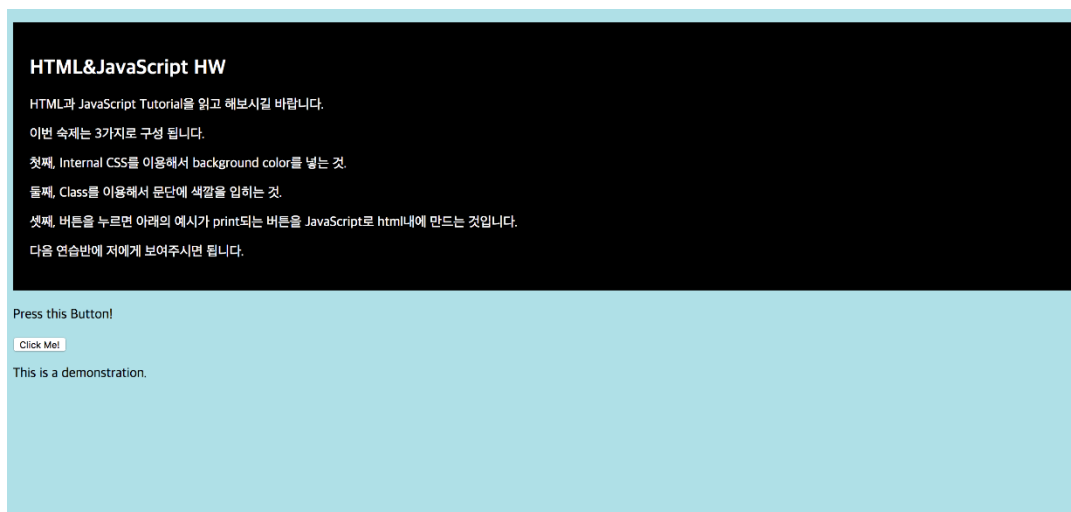
채점에 사용할 스크립트는 이번주중에 공지한다.

B. HTML 및 Javascript 과제

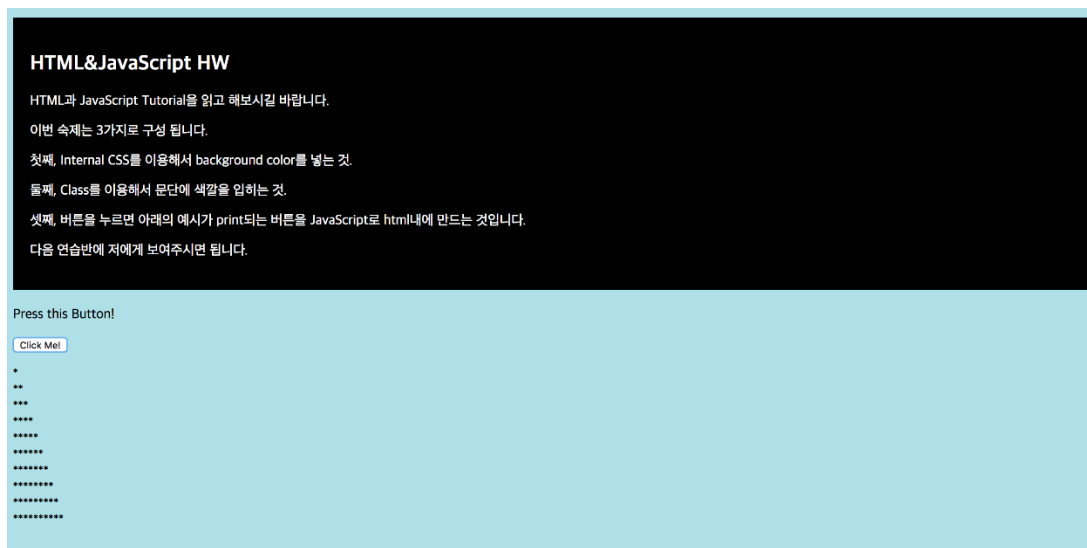
1) 설명

CSS 및 Javascript 를 이용한 간단한 예제를 작성한다. 그림 1 와 유사한 HTML 페이지를 만들되, 아래 세 가지 요구사항을 구현해야 한다.

- (1) HTML 소스에 CSS 를 이용해 body 태그의 background color 를 넣는 부분이 있어야 한다. (단, 검은색/흰색은 제외)
- (2) HTML 소스에 Class 를 이용해 element 에 색을 입히는 부분이 있어야 한다.
- (3) 버튼을 클릭할 경우 그림 2 와 같이 *이 print 되는 Javascript 를 작성해야 한다.



<그림 1. Click Me! 클릭 전>



<그림 2. Click Me! 클릭 후>

2) 채점

다음주 화요일 3 월 28 일 7 시 실습시간에 웹 페이지 (및 필요시 소스)를 확인함으로써 채점을 한다. 개인 노트북을 지참할 필요는 없다.

Appendix. Django REST 과제에서 구현해야 할 페이지의 예시

127.0.0.1:8000/debts/	127.0.0.1:8000/debts/id (id는 숫자)
채무 관계의 리스트	채무 관계 id의 정보
<div>Django REST framework</div> <div>Debt List</div> <h1>Debt List</h1> <div>GET /debts/</div> <pre> HTTP 200 OK Allow: GET, POST, HEAD, OPTIONS Content-Type: application/json Vary: Accept [{ "id": 11, "created": "2017-03-21T19:51:49.152749Z", "amount": 352, "borrower": 76, "lender": 79 }, { "id": 12, "created": "2017-03-21T19:51:49.163834Z", "amount": 263, "borrower": 78, "lender": 77 }, { "id": 13, "created": "2017-03-21T19:51:49.175327Z", "amount": 808, "borrower": 78, "lender": 75 }, { "id": 14, "created": "2017-03-21T19:51:49.186390Z", "amount": 518, "borrower": 81, "lender": 78 }, { "id": 15, "created": "2017-03-21T19:51:49.197827Z", "amount": 393, </pre>	<div>Django REST framework</div> <div>Debt List / Debt Detail</div> <h1>Debt Detail</h1> <div>GET /debts/12/</div> <pre> HTTP 200 OK Allow: PUT, GET, DELETE, OPTIONS Content-Type: application/json Vary: Accept { "id": 12, "created": "2017-03-21T19:51:49.163834Z", "amount": 263, "borrower": 78, "lender": 77 } </pre>

users/ 에 접속할 시 나오는 데이터	users/id 에 접속할 시 나오는 데이터
전체 유저의 목록	아이디가 id인 유저의 정보
<div>Django REST framework</div> <div>User List</div> <div>User List</div> <div>GET /users/</div> <div><pre>HTTP 200 OK Allow: GET, HEAD, OPTIONS Content-Type: application/json Vary: Accept [{ "id": 1, "username": "admin", "debts_as_borrower": [], "debts_as_lender": [] }, { "id": 74, "username": "test1", "debts_as_borrower": [], "debts_as_lender": [15] }, { "id": 75, "username": "test2", "debts_as_borrower": [18], "debts_as_lender": [13] }, { </pre></div>	<div>Django REST framework</div> <div>User List / User Detail</div> <div>User Detail</div> <div>GET /users/74/</div> <div><pre>HTTP 200 OK Allow: GET, HEAD, OPTIONS Content-Type: application/json Vary: Accept { "id": 74, "username": "test1", "debts_as_borrower": [], "debts_as_lender": [15] }</pre></div>

127.0.0.1:8000/usersum/	127.0.0.1:8000/usersum/id
전체 유저의 목록, 단 빌린 돈과 빌려준 돈을 보여줘야 함	아이디가 id인 유저의 정보, 단 빌린 돈과 빌려준 돈을 보여줘야 함
<div>Django REST framework</div> <div>User Sum List</div> <div>User Sum List</div> <div>GET /usersum/</div> <div>HTTP 200 OK Allow: GET, HEAD, OPTIONS Content-Type: application/json Vary: Accept</div> <div>[{ "username": "admin", "id": 1, "lended_money": 0, "borrowed_money": 0 }, { "username": "test1", "id": 74, "lended_money": 393, "borrowed_money": 0 }, { "username": "test2", "id": 75, "lended_money": 808, "borrowed_money": 846 }, { "username": "test3", "id": 76, "lended_money": 0, "borrowed_money": 352 }, { "username": "test4", "id": 77, "lended_money": 2151, "borrowed_money": 605 }]</div>	<div>Django REST framework</div> <div>User Sum List / User Sum Detail</div> <div>User Sum Detail</div> <div>GET /usersum/75/</div> <div>HTTP 200 OK Allow: GET, HEAD, OPTIONS Content-Type: application/json Vary: Accept</div> <div>{ "id": 75, "borrowed_money": 846, "username": "test2", "lended_money": 808 }</div>