65일차 강의

* 보통 **<div>** 안에 <form> 을 넣는게 일반적
* 그래야 form 안의 액션이나 안의 인풋태그들이 하나의 트랜잭션으로 관리하기 수월하다
* 그래야 div는 하나의 물건들을 관리하는 **컨테이너** 같은 느낌으로 사용된다
* 테이블없이도 div나 span 만 있으면 훨씬 레고처럼 규격화하여 다룰 수 있다
* 기존 프로젝트
* 차세대 프로젝트 참가
* Crud
  + Select
* 조회, 있니 없니의 판단
  + Read
  + Update
  + Delete
* 로그인도 무수한 과정이 있으므로 로그인 컨트롤러 따로 만들어야한다
* 사용자가 접근하는 것은 키보드로 소통하고, 시점이나 이벤트는 마우스를 이용한다.
* Model

: 데이터를 다른 페이지나 다른곳으로 전달할 때 담아 보내면된다

* 마치 req, request 객체처럼 담아서 전송하면 되듯이 Model이 그역할을 대신한다
* 모델은 자동형변환을 제공한다
* 다른 곳으로 전송하는 보관소 역할을 한다
* (실습)

Index.jps에서 버튼을 하나 만든다

버튼을 클릭하면 auth/loginForm.jsp화면을 전환하시오

* Service
* 여러 DAO를 호출하여 여러 번의 데이터 접근/갱신을 하며 그렇게 읽은 데이터에 대한 비즈니스 로직을 수행하고, 그것을 하나의 혹은 여러 개의 트랜잭션으로 묶는다.

즉, Service가 트랜잭션 단위이다.

* DAO
* 단일 데이터 접근/갱신만 처리한다
* 무조건 select문만 쓴다