

1. 페이스북 예제 (이어서)

1) 댓글 db 구축

- 테이블 생성

```
create table img_board_rep(  
    num number primary key,  
    img_num number references img_board(num) on delete cascade,  
    writer varchar2(20) not null,  
    content varchar2(100)  
);
```

- 시퀀스 생성

```
create sequence seq_img_board_rep;
```

2) img_rep 패키지 생성

- ImgRep (Dto)
- Dao interface *하나의 글에 해당하는 댓글을 selectAll
- DaoImpl
- Service interface
- ServiceImpl
- ImgBoardController에 ServiceImpl 기능 구현
- ImgBoard(dto) 수정 *reps 멤버 변수와 세터게터 추가
- ImgBoardController 수정

3) list.jsp

- 이중루프문 형식으로 각 이미지 마다 댓글들 출력.
- form에 댓글작성 버튼 누르면 실행될 action지정 (?cmd=write_rep)
- rep로만 두면 name같은 값들이 배열로 구성되기 때문에 num_rep로 변경

```
<tr><th>댓글</th><td>내용:<input type="text" id="${ib.num }_rep"><br/>  
작성자:<input type="text" id="${ib.num }_rep_writer">  
<form action="${pageContext.request.contextPath }/ImgBoardController?cmd=write_rep&num=${ib.num } method="post">  
<input type="button" value="댓글작성" onclick="write_rep(${ib.num })">
```

- dom(document object model)
모든 태그를 각각의 객체로 만듦
각 객체를 접근할 수 있다.
document.getElementById("id_name") => id로 접근(1개만 반환)
document.getElementsByTagName("tag_name ex.tr") => 태그로 접근(배열로 반환)

2. 쇼핑몰 연습

1) 구조

- 회원가입

id/pwd/name/email/type(구매자:consumer or 1, 판매자:seller or 2)*radio button으로

- 로그인

- 판매자 : 등록 상품 목록 출력
- 구매자 : 전체 상품 목록 출력

- 판매자 로그인 첫 페이지
- - 상품 등록
 - 내가 등록한 상품 목록. 상품명에 링크 => 상세페이지로 이동
 - 상세페이지
 - 수정
 - 삭제
- 구매자 로그인 첫 페이지
 - 전체 상품 목록. 상품명에 링크=> 구매페이지(상세 정보, 즉시구매, 장바구니, 구매 수량...)로 이동.
 - 구매페이지
 - 내 구매 목록
 - 장바구니 목록

2) db 구축

- 구매자 테이블

```
create table shop_mem(
  id varchar2(20) primary key,
  pwd varchar2(20) not null,
  name varchar2(20) not null,
  email varchar2(50),
  type varchar2(10) default 'consumer'
);
```

- 판매물품 테이블

```
create table shop_product(
  num number primary key,
  s_id varchar2(20) references shop_mem(id) on delete cascade,
  name varchar2(20) not null,
  price number not null,
  quantity number not null,
  img_path varchar2(100),
  info varchar2(100)
);
```

- 상품 이미지 폴더 경로: /shop_img/
- 판매물품 시퀀스

```
create sequence seq_shop_product;
```

- 주문 테이블

```

create table shop_order(
num number primary key,
prod_num number not null,
order_num number not null,
total_price number not null,
c_id varchar2(20) references shop_mem(id) on delete cascade,
order_date date,
order_type number
);

```

- 주문 시퀀스

```

create sequence seq_shop_order;

```

3) 흐름

a. 기본세팅

- src/Controllers 패키지 생성
- src/Controllers에 MemController, SellController, ConController 서블릿 생성
- cmd 패키지 생성
- cmd 패키지에 cmd.member, cmd.seller, cmd.consumer 패키지 생성
- cmd 패키지 아래 공통으로 쓸 Command.java 파일 넣기
- controllers 패키지 생성
- controllers 패키지에 memController, SellController, ConController 서블릿 생성

b. /WEB-INF/cmdprop에 properties 파일 생성

- memcmd.properties
 - joinForm=cmd.member.JoinFormCommand
 - join=cmd.member.JoinCommand
 - loginForm=cmd.member.LoginFormCommand
 - login=cmd.member.LoginCommand
 - editForm=cmd.member.EditFormCommand
 - edit=cmd.member.EditCommand
 - logout=cmd.member.LogoutCommand
 - out=cmd.member.OutCommand
- sellcmd.properties
 - menu = cmd.seller.MenuCommand
 - addForm=cmd.seller.AddFormCommand
 - add=cmd.seller.AddCommand
 - sellList=cmd.seller.SellListCommand
 - detail=cmd.seller.DetailCommand
 - editForm=cmd.seller.EditFormCommand
 - edit=cmd.seller.EditCommand
 - del=cmd.seller.DelCommand
- concmd.properties
 - menu = cmd.consumer.MenuCommand
 - productList=cmd.consumenr.ProductListCommand
 - detail=cmd.consumenr.DetailCommand

```
list_bought=cmd.consumer.ListBoughtCommand  
list_cart=cmd.consumer.ListCartCommand
```

- c. Controller 세팅
MemController=>초기:loginForm
SellController=>초기:list
ConController=>초기:list
- d. member 패키지 (model 구현)
 - Member(dto) 수정
 - dao 인터페이스
 - DaoImpl
 - service 인터페이스
 - serviceImpl
- e. cmd.member 패키지
- f. views/member, seller, consumer 디렉토리