

게시글		
번호	제목	작성자
O1	제목1	선녀
O2	제목2	나무꾼
O3	제목3	꽃사슴
O4	제목4	옥황상제

[ 1 ] [ 2 ] [ 3 ] [ 4 ] [ 5 ] [다음]

1그룹=> [ 1 ] [ 2 ] [ 3 ] [ 4 ] [ 5 ] [다음]  
 2그룹=> [이전] [ 6 ] [ 7 ] [ 8 ] [ 9 ] [ 10 ] [다음]  
 3그룹=> [이전] [ 11 ] [ 12 ] [ 13 ] [ 14 ]

현재페이지  
 2 -> 현재그룹 2 / 5 => 0 +1 => 1그룹  
 5 -> 현재그룹 5 / 5 => 1 그룹  
 11 -> 현재그룹 11/5 => 2 +1 => 3그룹

현재페이지 : **currentPage =1**

목록의 개수 : **pageSize :4**

총레코드수 : 33 테이블에서 구해야함 **totRecords**

총페이지수 : 9 총레코드수 / 페이지사이즈  
**totalPage**  
 33/4 => 8 +1 => 9페이지

그룹사이즈: 한 그룹에 5개 페이지를 보겠다.  
**grpSize =5**

현재페이지에 따른 현재그룹 구하기  
**currentGrp = currentPage / grpSize +1**

현재그룹의 시작번호, 끝번호 구하기  
**grpStartPage = (currentGrp-1) \*grpSize+ 1**  
**grpEndPage = currentGrp \* grpSize ;**

그룹의 마지막번호 확인  
 <총페이지가 9라면 10이 나오면 안됨>  
**grpEndPage = totalPage**

반복하기 , 반복제어변수 : **index**  
 (그룹시작번호~ 그룹마지막 번호까지)

```

<%int index = grpStartPage;

while( index <= grpEndPage) {
  %>
  <a href="paging.jsp?p=<%=index%>">[<%=index %>]</a>

  <%
    index++;
  } %>

```

<처리과정>

#### 페이징정보 정하기

1. 페이지 사이즈 정한다 => 한 페이지내 보여줄 레코드 수를 정한다 (예: 목록을 4건씩 보겠다)
2. 그룹의 사이즈를 정한다. ( 한 그룹내 5개씩 보겠다)

#### 페이징에 관련에 정보 구하기

- 3 총레코드를 구한다
- 4 총페이지를 구한다.
5. 현재페이지에 대한 현재그룹번호를 구한다
  - 현재그룹의 시작번호 구한다
  - 현재그룹의 마지막번호 구한다
6. 현재그룹의 마지막번호와 총페이지수를 판단한다
  - 총페이지수 보다 크다면 현재그룹의 마지막번호를 갱신한다
7. 현재그룹의 시작부터 마지막 번호까지 출력하기(반복문사용)
8. 이전, 다음 처리하기
  - 이전은 현재페이지가 1보다 작으면 이전 표시안됨
  - 다음은 전체페이지수가 현재페이지보다 크면 다음이 표시됨

#### 사용변수

현재페이지를 저장할 변수 : currentPage =1 , 기본은 1로 설정함

총레코드를 저장할 변수 : totRecords

총페이지를 저장할 변수 : totalPage

한페이지내 보여줄 레코드 수 : pageSize

#### 그룹관련변수

한 그룹의 사이즈를 저장할 변수 : grpSize

현재그룹의 시작번호를 저장할 변수 : grpStartPage

현재그룹의 끝번호를 저장할 변수 :grpEndPage

그룹의시작~끝까지 반복할 반복제어변수 : index

## 구현기능: 고객목록 제어장

### Model

#### MemberDAO

```
-dbCon() ;  
-ArrayList<Member> getListPage(int page)  // 페이지에 해당되는 목록 가져오기  
- int getTotal()                          // 전체레코드 수 가져오기
```

**Member** : 고객정보를 테이블로부터 저장할 객체 (VO 객체라고 함) value Object

### Controller (서블릿) 요청을 처리할 서블릿

**ListPage** url-pattern ("/list.page")

### View

**listPage.jsp**