

JSP

1. 페이징

1. 반드시 GET 방식으로만 처리한다
2. 이동할 때 페이지 번호, 보여줄 페이지 개수 를 가지고 다녀야 한다
 - > 목록 으로 나올 때 현재 페이지를 유지하기 위해
3. 페이징 처리하는 로직을 클래스로 분류한다
 - > PageVO클래스는 페이징 관련 모든 페이지를 계산한다

오라클의 쿼리문

인라인뷰

```
select *  
from (select rownum rn, num, title, content, writer, regdate, hit  
      from t_board  
      order by num desc  
      )  
where rn > 10 and rn <= 20;
```

← 11-20번 데이터가 조회된다

pageNum

(조회하고 있는 페이지 - 1) * 10

(조회하고 있는 페이지) * 10

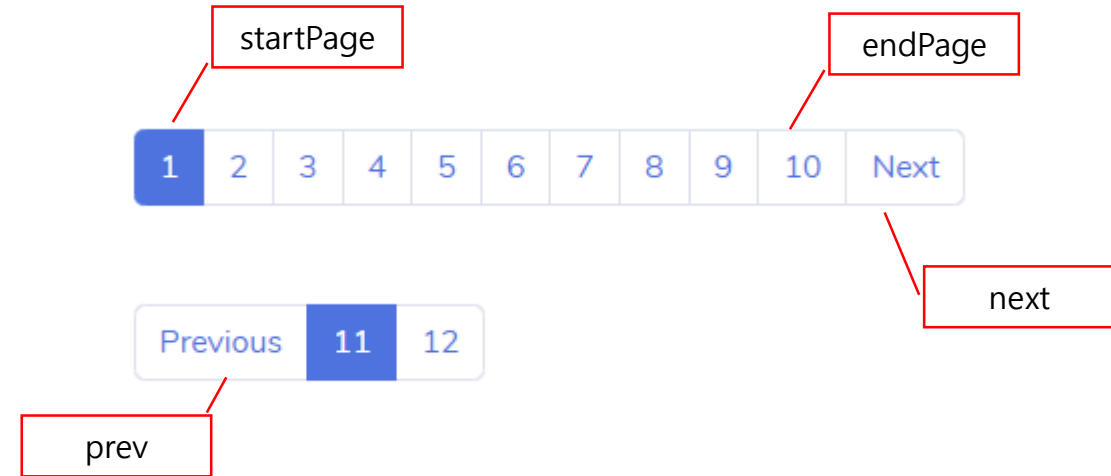
오라클의 sql문

```
select *  
from (select rownum as rn, num, writer, title, content, regdate, hit  
      from (select *  
            from t_board  
            order by num desc)  
      )  
where rn > ? and rn <= ?;
```

?는 뭐가 들어가는가?

2. 페이징 (PageVO 클래스) -페이징 계산 처리 클래스

- 1 **total** 게시판 글 전체 개수.
- 2 **endPage**: 게시판을 화면에 보여질 마지막 페이지 번호.
- 3 **startPage**: 게시판을 화면에 보여질 첫번째 페이지 번호.
- 4 **realEnd**: 게시판의 실제 마지막 페이지 번호.
- 5 **prev**: 이전 페이지 버튼 활성화 여부.
- 6 **next**: 다음 페이지 버튼 활성화 여부.



공식

2 **endPage** =

1~10 페이지 클릭시 10
11~20 페이지 클릭시 20
21~30 페이지 클릭시 30

3 **startPage** =

endPage가 10 일때 1
endPage가 20 일때 11
endPage가 30 일때 21

4 **realEnd** =

게시물이 50개 라면 5
게시물이 51개 라면 6
게시물이 101개 라면 11

공식

4 **prev** =

startPage가 1이면 비활성화
startPage가 11이면 활성화

5 **next** =

realEnd가 endPage보다 크면 활성화

Math.ceil() 함수를 이용한다

```
public class PageVO {
    private int startPage; //게시판 화면에 보여질 첫페이지 번호
    private int endPage; //게시판 화면에 보여질 끝페이지 번호
    private boolean prev, next; //다음 이전 활성화 여부

    private int pageNum; //현재 조회하는 페이지 번호
    private int amount = 10; //한 페이지에서 몇개의 데이터를 보여줄건가
    private int total; //총 게시물 수

    public PageVO(int pageNum ,int total, int amount) {

        this.total = total; //전체게시글 수
        this.pageNum = pageNum; //현재 조회하는 페이지 번호
        this.amount = amount; //한 페이지에서 몇개의 데이터를 보여줄건가
        this.endPage = (int)(Math.ceil(this.pageNum / 10.0) ) * 10;

        this.startPage = endPage - 10 + 1;

        int realEnd = (int)Math.ceil(total / (double)this.amount );

        if(this.endPage > realEnd ) {
            this.endPage = realEnd;
        }

        this.prev = this.startPage > 1;
        this.next = realEnd > this.endPage;

    }
}
```