

페이지네이션(Pagination)

동의과학대학교
컴퓨터정보과
김진숙

Paging

- 데이터베이스의 레코드를 원하는 만큼 나누어 한 화면에 출력하는 것
- 한 화면의 목록에는 다음과 같이 페이지를 네비게이션 할 수 있는 링크가 포함되어야 함

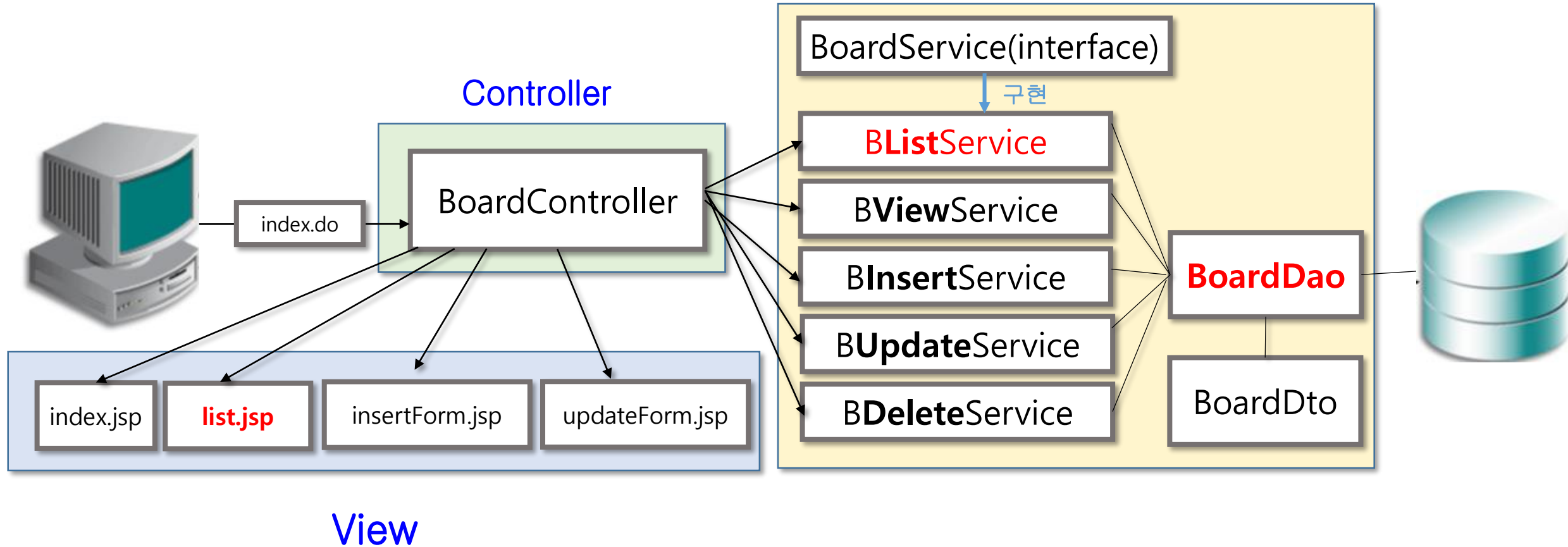


list.jsp 실행 화면

| 게시판 목록 | | | |
|--|----------------------|-----------|--------------|
| num | subject | writer | regDate |
| 59 | 안녕46 | gildong11 | 2022. 9. 18. |
| 58 | 안녕45 | gildong | 2022. 9. 18. |
| 57 | 안녕44 | gildong | 2022. 9. 18. |
| 56 | 안녕43 | gildong | 2022. 9. 18. |
| 54 | 안녕41 | gildong | 2022. 9. 18. |
| 53 | 안녕40 | gildong | 2022. 9. 18. |
| 52 | 안녕39 | gildong | 2022. 9. 18. |
| 51 | 안녕38 | gildong | 2022. 9. 18. |
| 47 | 안녕34 | gildong | 2022. 9. 18. |
| 44 | 안녕31 | gildong | 2022. 9. 18. |
| <div>홈으로 게시물 등록</div> | | | |
| <div>Prev 1 2 3 4 5 Next</div> | | | |

페이징 추가하기

- 자바 패키지와 list.jsp파일 변경



Paging이 적용된 화면

| 게시판 목록 | | | |
|--------|----------------------|-----------|--------------|
| num | subject | writer | regDate |
| 59 | 안녕46 | gildong11 | 2022. 9. 18. |
| 58 | 안녕45 | gildong | 2022. 9. 18. |
| 57 | 안녕44 | gildong | 2022. 9. 18. |
| 56 | 안녕43 | gildong | 2022. 9. 18. |
| 54 | 안녕41 | gildong | 2022. 9. 18. |
| 53 | 안녕40 | gildong | 2022. 9. 18. |
| 52 | 안녕39 | gildong | 2022. 9. 18. |
| 51 | 안녕38 | gildong | 2022. 9. 18. |
| 47 | 안녕34 | gildong | 2022. 9. 18. |
| 44 | 안녕31 | gildong | 2022. 9. 18. |

[홈으로](#) [게시글 등록](#)

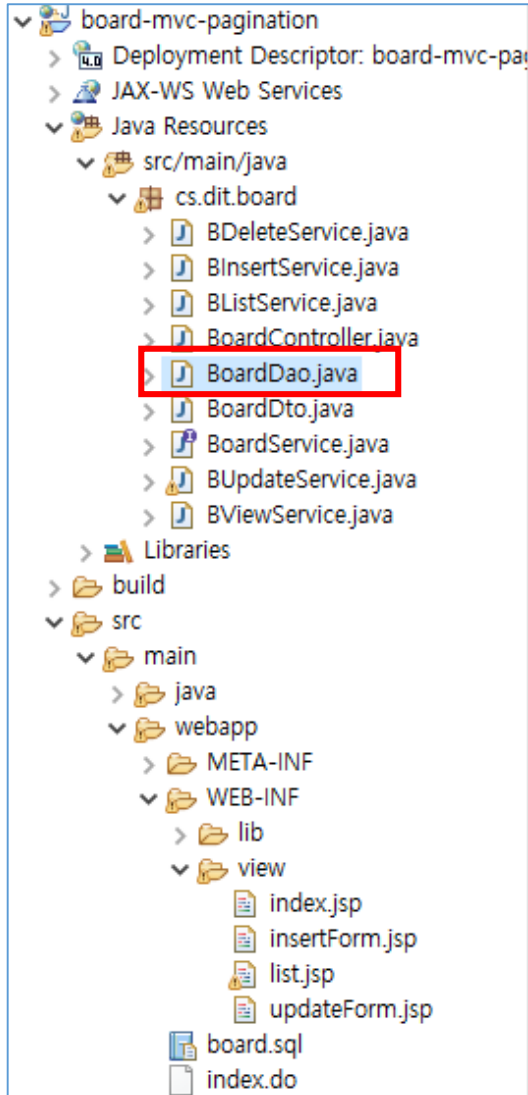
Prev 1 2 3 4 5 Next

- 전체 레코드의 개수 : DB연동 필요
 - count
- 한 화면에 출력되는 레코드 수 : 10개
 - numOfRecords
- 한 화면에 보여지는 페이지 번호 개수 : 5개
 - numOfPages
- 한 화면의 시작 페이지 번호 : 1, 6, 11, 16, ...
 - startNum
- 마지막 출력 페이지 번호
 - lastNum

Prev 6 Next

전체 레코드 수 파악하기 : getCount()

- LoginDAO.java 에 메소드 추가하기



```
public class BoardDao {

    /**=====
    * 패키지명 : cs.dit.board
    * 파일명    : BoardDao.java
    * 작성자    : 김진숙
    * 변경이력 :
    * 2022-9-11/ 최초작성/ 김진숙
    * 프로그램 설명 : board 테이블의 내용과 연동하여 회원관리
    * - getConnection() : 커넥션풀에서 연결객체 얻기
    * - insert(BoardDto dto) : 사용자가 입력한 게시글 정보 DB 입력
    * - list() : DB에서 게시글 정보 조회
    * - update(BoardDto dto) : 변경된 게시글 정보 DB 변경
    * - delete(int bcode) : 해당 bcode 게시글 DB 삭제
    * - recordCount() : 전체 테이블의 레코드 수 조회
    *=====*/

    private Connection getConnection() throws Exception{
        //1. JNDI를 이용하기 위한 객체 생성
        Context initCtx = new InitialContext();
```

전체 레코드 수 파악하기 : recordCount()

- boardDao.java 에 메소드 추가하기

```
public int recordCount() {  
    String sql = "SELECT COUNT(bcode) FROM board";  
    int count = 0;  
  
    try (    Connection con = getConnection();  
          Statement stmt = con.createStatement();  
          ResultSet rs = stmt.executeQuery(sql);  
        )  
    {  
  
        if(rs.next())  
            count = rs.getInt(1);  
  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
  
    return count;  
}
```

레코드를 numOfRecords 개씩 반환하기 : list() 변경

- 매개변수

- int page : 선택된 페이지의 번호
- int numOfRecords : 한 화면에 표시되는 레코드의 갯수

```
public class BoardDao {  
  
    /**=====*/  
    * 패키지명 : cs.dit.board  
    * 파일명   : BoardDao.java  
    * 작성자   : 김진숙  
    * 변경이력 :  
    *   2022-9-11/ 최초작성/ 김진숙  
    * 프로그램 설명 : board 테이블의 내용과 연동하여 회원관리  
    *   - getConnection() : 커넥션풀에서 연결객체 얻기  
    *   - insert(BoardDto dto) : 사용자가 입력한 게시글 정보 DB 입력  
    *   - list(int page, int numOfRecords) : DB에서 게시글 정보 조회  
    *   - update(BoardDto dto) : 변경된 게시글 정보 DB 변경  
    *   - delete(int bcode) : 해당 bcode 게시글 DB 삭제  
    *   - recordCount() : 전체 테이블의 레코드 수 조회  
    *=====*/  
}
```


SQL문 작성하기

- Mariadb : LIMIT 절

```
SELECT * FROM table  
LIMIT 0, 10
```

```
SELECT * FROM table  
LIMIT 10 OFFSET 0
```

list(page, numOfRecords)

```
public ArrayList<BoardDto> list(int page, int numOfRecords){  
    ArrayList<BoardDto> dtos = new ArrayList<BoardDto>();  
    String sql = "SELECT * from board order by bcode desc limit ?, ?";  
    try (    Connection con = getConnection();  
           PreparedStatement pstmt = con.prepareStatement(sql);)  
    {  
        pstmt.setInt(1, (page-1)*numOfRecords);  
        pstmt.setInt(2, numOfRecords);  
        try(ResultSet rs = pstmt.executeQuery();)  
        {  
            while(rs.next()) {  
                BoardDto dto = new BoardDto();  
                dto.setBcode(rs.getInt("bcode"));  
                dto.setSubject(rs.getString("subject"));  
                dto.setContent(rs.getString("content"));  
                dto.setWriter(rs.getString("writer"));  
                dto.setRegDate(rs.getDate("regDate"));  
                dtos.add(dto);  
            }  
        } catch (Exception e) {  
            e.printStackTrace();  
        }  
    } catch (Exception e) {  
        e.printStackTrace();  
    }  
    return dtos;  
}
```

| page | numOfRecords | limit | 한 화면 출력 레코드 |
|------|--------------|-------|-------------|
| 1 | 10 | 0, 10 | 1 ~ 10 |
| 2 | 10 | 1, 10 | 11 ~ 20 |
| 3 | 10 | 2, 10 | 21 ~ 30 |
| 4 | 10 | 3, 10 | 31 ~ 40 |
| 5 | 10 | 4, 10 | 41 ~ 50 |
| | | ⋮ | |

(Page - 1)

목록 화면 출력 준비 : BListService.java

이전 코드

```
1 package cs.dit.board;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 public class BListService implements BoardService {
11
12     @Override
13     public void execute(HttpServletRequest request, HttpServletResponse response) {
14         BoardDao dao = new BoardDao();
15         ArrayList<BoardDto> dtos = dao.list();
16         request.setAttribute("dtos", dtos);
17     }
18 }
19 }
```

목록 화면 출력 준비 : BListService.java

```
1 package cs.dit.board;
2
3 import java.io.IOException;
4 import java.util.ArrayList;
5
6 import javax.servlet.ServletException;
7 import javax.servlet.http.HttpServletRequest;
8 import javax.servlet.http.HttpServletResponse;
9
10 public class BListService implements BoardService {
11
12     @Override
13     public void execute(HttpServletRequest request, HttpServletResponse response) {
14         BoardDao dao = new BoardDao();
15         ArrayList<BoardDto> dtos = dao.list();
16         request.setAttribute("dtos", dtos);
17     }
18 }
19 }
```

- 시작될 페이지 숫자 : startNum
- 전체 페이지 숫자 : lastNum

lastNum -> 전체레코드의 수 / numOfRecords

startNum 찾기

- 한 화면의 시작 페이지 번호 : **startNum**

| | | | | | |
|---|-------|----|----|----|----|
| 1 | 1 | 2 | 3 | 4 | 5 |
| 2 | 6 | 7 | 8 | 9 | 10 |
| 3 | 11 | 12 | 13 | 14 | 15 |
| 4 | 16 | 17 | 18 | 19 | 20 |
| 5 | 21 | 22 | 23 | 24 | 25 |
| | | | | | |

현재 페이지 번호(p)가 4일 때 startNum? → 1

현재 페이지 번호(p)가 8일 때 startNum? → 6

현재 페이지 번호(p)가 22일 때 startNum? → 21

| | | |
|-----------|------------|-------------|
| 5 - 4 = 1 | 10 - 4 = 6 | 15 - 4 = 11 |
| 4 - 3 = 1 | 9 - 3 = 6 | 14 - 3 = 11 |
| 3 - 2 = 1 | 8 - 2 = 6 | 13 - 2 = 11 |
| 2 - 1 = 1 | 7 - 1 = 6 | 12 - 1 = 11 |
| 1 - 0 = 1 | 6 - 0 = 6 | 11 - 0 = 11 |

- 나머지 계산 필요 : $(p - 1) \% 5$
- $\text{startNum} = p - (p - 1) \% 5$

목록 화면 출력 준비 : BListService.java

```
public class BListService implements BoardService {
    @Override
    public void execute(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {

        BoardDao dao = new BoardDao();
        int numberOfRecords = 10;
        int count = dao.recordCount(); //전체 레코드 수를 조회

        //전달되는 page 번호를 얻고 확인 : null값이거나 ""(빈문자열)이 아니라는 것을 확인
        String page_ = request.getParameter("p");
        int p = 1; //페이지 초기 값

        //전달된 page 번호가 null이거나 ""빈문자열이 아닐경우 page_값을 int 변환하여 list에 전달
        if(page_ != null && !page_.equals(""))
            p = Integer.parseInt(page_);

        //list(page, numOfRecords)호출 : 현재 페이지번호를 매개변수로 전달
        ArrayList<BoardDto> dtos = dao.list(p, numberOfRecords);

        //화면에 출력될 첫 페이지 번호 값 계산
        int startNum = p - ((p - 1) % 5);

        //마지막 출력 페이지 번호를 얻기 위해 10으로 나누고 올림하기
        int lastNum = (int) Math.ceil(count / 10.0);

        //requestScope에 저장 -> el에 사용
        request.setAttribute("dtos", dtos);
        request.setAttribute("p", p); //현재페이지 출력에 사용
        request.setAttribute("startNum", startNum);
        request.setAttribute("lastNum", lastNum);
    }
}
```

목록 화면 출력 준비

The image shows a web browser window displaying a Bootstrap 4 tutorial page. The browser's address bar shows the URL `https://www.w3schools.com/bootstrap4/bootstrap_pagination.asp`. The page title is "Bootstrap 4 Tutorial". A sidebar menu on the left lists various Bootstrap components, with "BS4 Pagination" highlighted by a red rectangle. A red arrow points from this menu item to a code block. The code block contains the following HTML code for a pagination component:

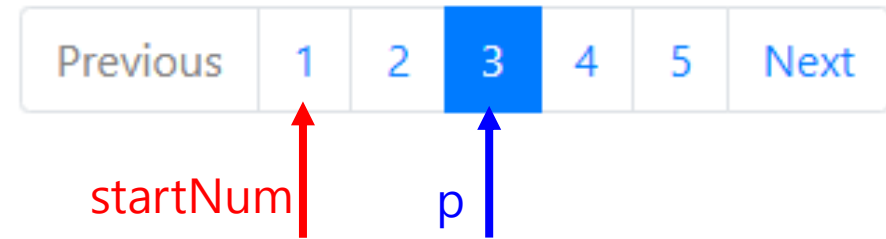
```
<ul class="pagination">
  <li class="page-item"><a class="page-link" href="#">Previous</a></li>
  <li class="page-item"><a class="page-link" href="#">1</a></li>
  <li class="page-item"><a class="page-link" href="#">2</a></li>
  <li class="page-item"><a class="page-link" href="#">3</a></li>
  <li class="page-item"><a class="page-link" href="#">Next</a></li>
</ul>
```

Below the code block, a rendered pagination component is shown. It consists of a "Previous" button, three numbered buttons (1, 2, 3), and a "Next" button. The "Next" button is highlighted with a green background and a white arrow pointing to the right. The text "Tutorial" is visible in the background of the rendered component.

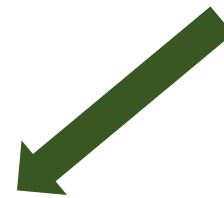
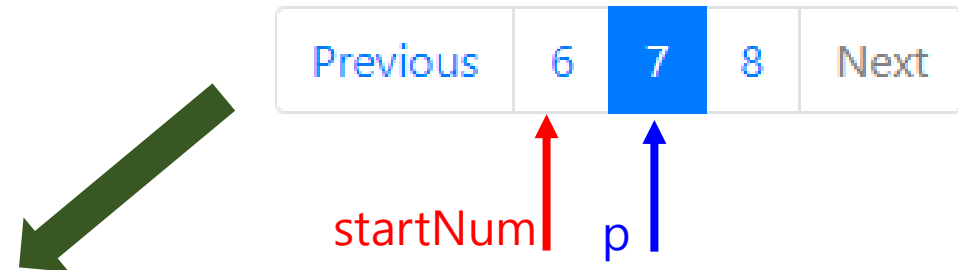
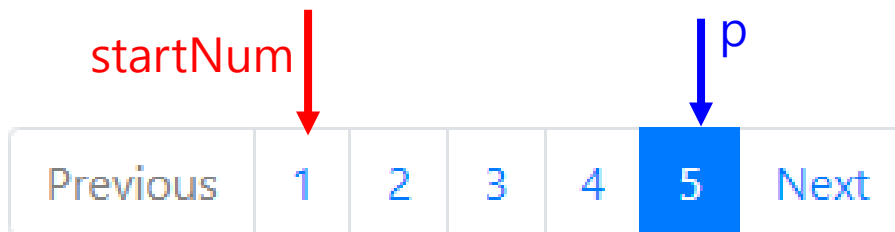
pagination

- Previous 관련 내용

- $\text{startNum} \leq 1$ 경우
 - 앞으로 더 움직일 페이지 없음



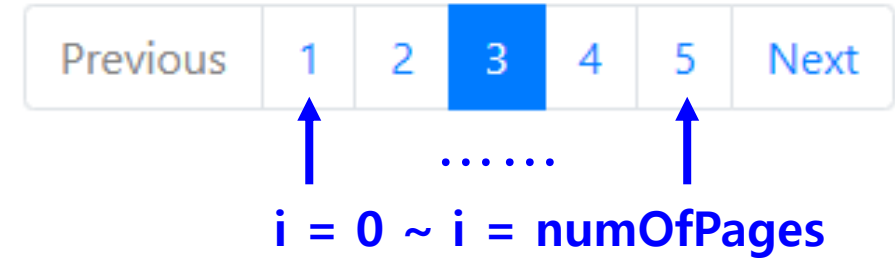
- $\text{startNum} > 1$ 경우
 - Previous를 선택하면 이전 화면의 마지막 페이지가 p가 되도록 함



pagination

• Page 숫자 링크 관련 내용

- 한 화면당 numOfPages(5개)의 페이지 생성
 - startNum+i (0~4)



- 페이지의 개수가 마지막 페이지(lastNum)을 넘어서 생성되면 안됨
 - $\text{startNum} + i \leq \text{lastNum}$ 보다 작거나 같을 때만 화면에 페이지 링크 생성
 - 예) $\text{startNum} = 6$ / $\text{lastNum} = 8$

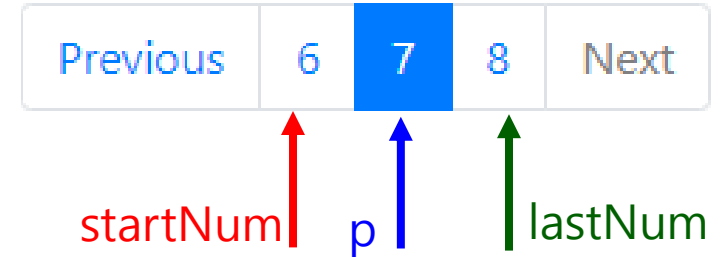


| | |
|---------|-----------------|
| $i = 0$ | $6 + 0 = 6 < 8$ |
| $i = 1$ | $6 + 1 = 7 < 8$ |
| $i = 2$ | $6 + 2 = 8 = 8$ |
| $i = 3$ | $6 + 3 = 9 > 8$ |

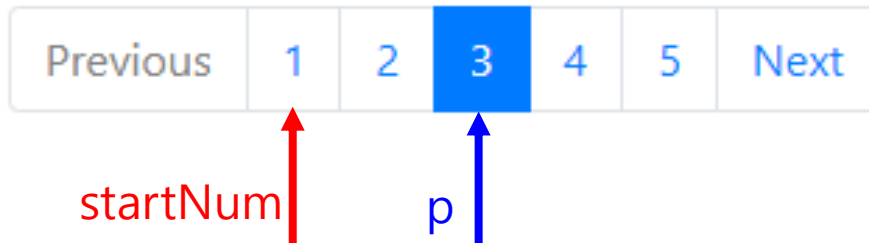
pagination

- Next 관련 내용

- $\text{startNum} + \text{numOfPages}(5) > \text{lastNum}$ 경우
 - 뒤로 더 움직일 페이지가 없음
 - 예) $6 + 5 > 8$



- $\text{startNum} + \text{numOfPages}(5) \leq \text{lastNum}$ 경우
 - Next를 선택하면 다음 화면의
 - 첫 페이지가 p가 되도록 함
 - 예) $1 + 5 < 8$



pagination

```
<body>
  <div class="container"><br>
  <h2 class="text-center font-weight-bold">게시판 목록</h2>
  <br>
  <table class="table table-hover">
    <tr>
      <th>num</th>
      <th>subject</th>
      <th>writer</th>
      <th>regDate</th>
    </tr>

    <c:forEach var='dto' items='${dtos}'>
      <tr>
        <td>${dto.bcode }</a></td>
        <td><a href="updateForm.do?bcode=${dto.bcode}">${dto.subject}</a></td>
        <td>${dto.writer}</td>
        <td><fmt:formatDate value="${dto.regDate}"/></td>
      </tr>
    </c:forEach>
  </table>

  <input type="button" value = "홈으로" onclick = "location.href='index.do'">
  <input type="button" value = "게시글 등록" onclick = "location.href='insertForm.do'">
</div>

<!-- 현재페이지/전체페이지 출력 부분 -->
<div class="d-flex justify-content-end">
  <span>${p}</span>&nbsp;&nbsp;&nbsp;/ <span>${lastNum}</span>pages
</div>
```

pagination

```
<!-- 페이지 나누기 -->
```

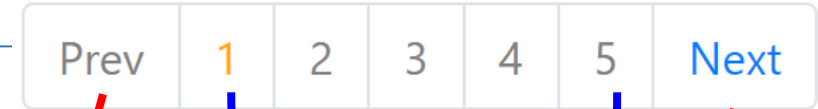
```
<div class="d-flex justify-content-center">
  <ul class="pagination">
```

```
    <c:if test = "${startNum>1}">
      <li class="page-item"><a class="page-link" href="list.do?p=${startNum-1}">Prev</a></li>
    </c:if>
    <c:if test = "${startNum <=1}">
      <li class="page-item"><a class="page-link" style="color:gray" onclick="alert('이전 페이지가 없습니다!')" href="#">Prev</a></li>
    </c:if>
```

```
    <c:forEach var="i" begin="0" end="4" step="1">
      <c:if test="${startNum + i <= lastNum}">
        <c:if test = "${startNum+i == p}">
          <li class="page-item active"><a class="page-link" href="list.do?p=${startNum+i}">${startNum+i}</a></li>
        </c:if>
        <c:if test = "${startNum + i != p}">
          <li class="page-item"><a class="page-link" href="list.do?p=${startNum+i}">${startNum+i}</a></li>
        </c:if>
      </c:if>
    </c:forEach>
```

```
    <c:if test = "${startNum + 5 <= lastNum}">
      <li class="page-item"><a class="page-link" href="list.do?p=${startNum+5}">Next</a></li>
    </c:if>
    <c:if test = "${startNum+5 > lastNum}">
      <li class="page-item"><a class="page-link" style="color:gray" onclick="alert('다음 페이지가 없습니다!')" href="#">Next</a></li>
    </c:if>
```

```
</ul>
</div>
```



결과 화면 출력

| 게시판 목록 | | | |
|---|----------------------|---------|--------------|
| num | subject | writer | regDate |
| 39 | 안녕26 | gildong | 2022. 9. 18. |
| 38 | 안녕25 | gildong | 2022. 9. 18. |
| 37 | 안녕24 | gildong | 2022. 9. 18. |
| 36 | 안녕23 | gildong | 2022. 9. 18. |
| 34 | 안녕21 | gildong | 2022. 9. 18. |
| 33 | 안녕20 | gildong | 2022. 9. 18. |
| 32 | 안녕19 | gildong | 2022. 9. 18. |
| 31 | 안녕18 | gildong | 2022. 9. 18. |
| 30 | 안녕17 | gildong | 2022. 9. 18. |
| 29 | 안녕16 | gildong | 2022. 9. 18. |
| <div>홈으로 게시글 등록</div> | | | |
| <div>Prev 1 2 3 4 5 Next</div> | | | |

과제

- member.war 파일을 아레테에서 다운로드 하여 다음과 같이 작성
 - 코드 내 누락된 코드 완성
 - member-mvc로 디자인 패턴 변경
 - pagination을 추가할 것
 - 프로젝트를 자신의 github에 업로드 할 것(10월 7일에 완료할 것)
 - 개인 과제 점수에 포함