

■ Pagination

● 프로젝트 / 앱 생성

- django-admin startproject **project4**
- python manage.py startapp **pagination**

● model 생성




- pagination/models.py

```
from django.db import models
```

```
class Board(models.Model):
```

```
    title = models.CharField(max_length=100)
```

```
    content = models.CharField(max_length=200)
```

pagination_board	
 id	integer
 title	varchar(100)
 content	varchar(200)

■ Pagination

● 뷰 생성

– pagination/views.py

```
from django.http import HttpResponse
from .models import *

def insert(request):
    for i in range(1, 1001): # 게시물 1000개 입력
        Board.objects.create(title='제목 ' + str(i), content='내용 ' + str(i))
    return HttpResponse('insert 1000')
```

● 주소 생성

– project4/urls.py

```
from django.contrib import admin
from django.urls import path
from pagination.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('insert', insert),
]
```

■ Pagination

● 게시물 데이터 입력

– <http://localhost:8000/insert>

[←](#) [→](#) [↻](#) [ℹ localhost:8000/insert](#)

insert 1000

	id	title	content
	필터	필터	필터
1	1000	제목 1000	내용 1000
2	999	제목 999	내용 999
3	998	제목 998	내용 998
4	997	제목 997	내용 997
5	996	제목 996	내용 996
6	995	제목 995	내용 995
7	994	제목 994	내용 994
8	993	제목 993	내용 993
9	992	제목 992	내용 992
10	991	제목 991	내용 991
11	990	제목 990	내용 990

[|<](#) [<](#) 1 - 11 of 1000 [>](#) [>|](#) [특정 레코드 행으로 가기](#)

■ Pagination

● Paginator 객체 활용

– import

```
from django.core.paginator import Paginator  
from pagination.models import Board
```

– Paginator 객체 생성

```
p = Paginator(Board.objects.order_by('-id'), 10)
```

Terminal

```
+ In [1]: from django.core.paginator import Paginator  
×  
In [2]: from pagination.models import Board  
  
In [3]: board = Board.objects.order_by('-id')  
  
In [4]: p = Paginator(board, 10)
```

■ Pagination

● Paginator 객체 활용

– 전체 데이터 수

```
Terminal
+ In [5]: p.count
× Out[5]: 1000
```

– 전체 페이지 수

```
Terminal
+ In [8]: p.num_pages
× Out[8]: 100

In [9]: p.page_range
Out[9]: range(1, 101)
```

■ Pagination

● Paginator 객체 활용

- 페이지별 데이터 정보 확인

```
Terminal
+ In [15]: page = p.page(1)
x
In [16]: page.object_list
Out[16]: <QuerySet [<Board: Board object (1000)>, <Board: Board object (999)>, <Board: Board object (998)>, <Board: Board object (997)>, <Board: Board object (996)>, <Board: Board object (995)>, <Board: Board object (994)>, <Board: Board object (993)>, <Board: Board object (992)>, <Board: Board object (991)>]>
```

■ Pagination

● Paninator 객체 활용

- 현재 페이지를 기준으로 앞 뒤 페이지 정보 확인

```
Terminal
+ In [17]: page.has_next()
× Out[17]: True

In [18]: page.has_previous()
Out[18]: False
```

[illegible]

■ Pagination

● Paginator 객체 활용

- 현재 페이지 데이터 시작 / 끝 번호 확인

```
Terminal
+ In [21]: page.start_index()
× Out[21]: 1

In [22]: page.end_index()
Out[22]: 10
```


■ Pagination

● Pagination 적용 데이터 출력

– pagination/views.py

```
def list(request):
    page = request.GET['page']
    board_list = Board.objects.order_by('-id')
    paginator = Paginator(board_list, 10)
    page_info = paginator.page(page)
    return render(
        request, 'pagination/list.html',
        {
            'page_info': page_info
        }
    )
```

– project4/urls.py

```
urlpatterns = [
    ...
    path('list', list),
]
```

■ Pagination

● Pagination 적용 데이터 출력

– pagination/templates/pagination/list.html

```
<table border="1">
  <tr>
    <td>번호</td><td>제목</td><td>내용</td>
  </tr>
  {% for board in page_info %}
  <tr>
    <td>{{ board.id }}</td>
    <td>{{ board.title }}</td>
    <td>{{ board.content }}</td>
  </tr>
  {% endfor %}
</table>
```

■ Pagination

● Pagination 적용 데이터 출력

– pagination/templates/pagination/list.html

```
<div class="pagination">
  <span class="step-links">
    이전 페이지 존재 여부
    {% if page_info.has_previous %}
      <a href="?page=1">&laquo; first 이전 페이지
      <a href="?page={{ page_info.previous_page_number }}">previous</a>
    {% endif %}

    <span class="current"> 현재 페이지 전체 페이지
    Page {{ page_info.number }} of {{ page_info.paginator.num_pages }}.
  </span>

  다음 페이지 존재 여부
  {% if page_info.has_next %}
    <a href="?page={{ page_info.next_page_number }}">next 다음 페이지
    <a href="?page={{ page_info.paginator.num_pages }}">last &raquo; 전체 페이지
  {% endif %}
</span>
</div>
```

■ Pagination

● Pagination 적용 데이터 출력

번호	제목	내용
1000	제목 1000	내용 1000
999	제목 999	내용 999
998	제목 998	내용 998
997	제목 997	내용 997
996	제목 996	내용 996
995	제목 995	내용 995
994	제목 994	내용 994
993	제목 993	내용 993
992	제목 992	내용 992
991	제목 991	내용 991

Page 1 of 100. [next](#) [last](#) »

번호	제목	내용
980	제목 980	내용 980
979	제목 979	내용 979
978	제목 978	내용 978
977	제목 977	내용 977
976	제목 976	내용 976
975	제목 975	내용 975
974	제목 974	내용 974
973	제목 973	내용 973
972	제목 972	내용 972
971	제목 971	내용 971

« [first](#) [previous](#) Page 3 of 100. [next](#) [last](#) »

번호	제목	내용
10	제목 10	내용 10
9	제목 9	내용 9
8	제목 8	내용 8
7	제목 7	내용 7
6	제목 6	내용 6
5	제목 5	내용 5
4	제목 4	내용 4
3	제목 3	내용 3
2	제목 2	내용 2
1	제목 1	내용 1

« [first](#) [previous](#) Page 100 of 100.