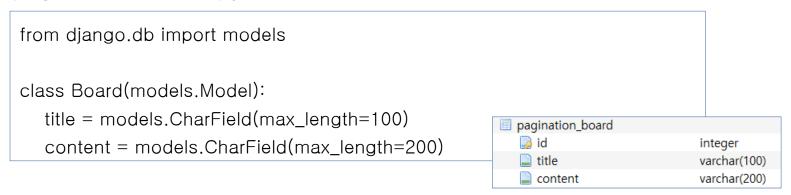
- Pagination
 - 프로젝트 / 앱 생성
 - django-admin startproject project4
 - python manage.py startapp pagination
 - model 생성
 - pagination/models.py



- 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),
]
```

- 게시물 데이터 입력
 - http://localhost:8000/insert



	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

- Paninator 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
x
In [2]: from pagination.models import Board
In [3]: board = Board.objects.order_by('-id')
In [4]: p = Paginator(board, 10)
```

- Paninator 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)
```

- Paninator 객체 활용
 - 페이지별 데이터 정보 확인

- Paninator 객체 활용
 - 현재 페이지를 기준으로 앞 뒤 페이지 정보 확인

```
+ In [17]: page.has_next()

Out[17]: True

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

```
the important in [19]: page.next_page_number()

volut[19]: 2

In [20]: page.previous_page_number()

EmptyPage Traceback (most recent call last)
```

- Paninator 객체 활용
 - 현재 페이지 데이터 시작 / 끝 번호 확인

- 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 적용 데이터 출력
 - pagination/templates/pagination/list.html

```
선성>번호
\td>\td>\td>\td>\\

(% for board in page_info %}

{tr>
{{ board.id }}

{td>{{ board.id }}

{td>{{ board.title }}

{td>{{ board.content }}

{td>}{

{% endfor %}
```

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

● Pagination 적용 데이터 출력

