

■ Paging

● 프로젝트 / 앱 생성

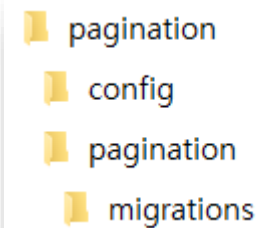
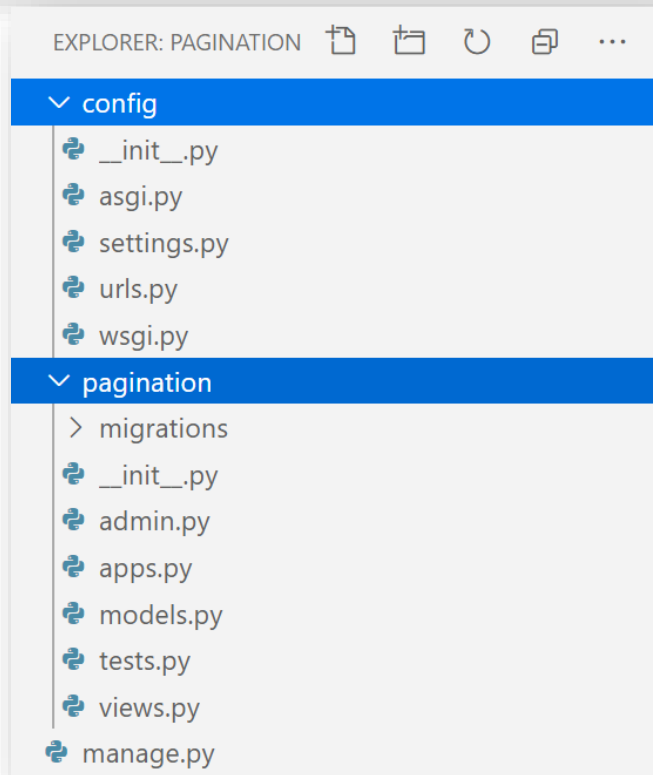
```
E:\dev\python_django>mkdir pagination
```

```
E:\dev\python_django>cd pagination
```

```
E:\dev\python_django\pagination>django-admin startproject config .
```

```
E:\dev\python_django\pagination>django-admin startapp pagination
```

```
E:\dev\python_django\pagination>_
```



■ Paging

● 테스트용 데이터 입력

1. settings.py app 등록
2. Model 작성
3. makemigrations & migrate 실행
4. urls.py 주소 생성
5. views.py 데이터 입력 기능 함수 작성

■ Paging

- 테스트용 데이터 입력 - 1. settings.py app 등록

```
INSTALLED_APPS = [  
    'pagination',  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.sessions',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
]
```

■ Paging

● 테스트용 데이터 입력 - 2. Model 작성

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

```
class Data(models.Model):  
    text = models.CharField(max_length=100)  
    cre_date = models.DateTimeField()
```

■ Paging

● 테스트용 데이터 입력 - 3. makemigrations & migrate 실행

- python manage.py makemigrations
- python manage.py migrate

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
PS E:\dev\python_django\pagination> python manage.py makemigrations
Migrations for 'pagination':
  pagination\migrations\0001_initial.py
    - Create model Data
PS E:\dev\python_django\pagination>
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: powershell

```
PS E:\dev\python_django\pagination> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, pagination, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying pagination.0001_initial... OK
  Applying sessions.0001_initial... OK
PS E:\dev\python_django\pagination>
```

■ Paging

- 테스트용 데이터 입력 - 4. urls.py 주소 생성

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

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

■ Paging

● 테스트용 데이터 입력 - 5. views.py 데이터 입력 기능 함수 작성

```
from django.shortcuts import render
from django.http import HttpResponse
from django.utils import timezone

from .models import *

def insert(request):
    for i in range(1, 123): # 게시물 122 개 입력
        data = Data(text='Data-%s' % i, cre_date=timezone.now())
        data.save()

    return HttpResponse('insert 122')
```

■ Paging

● 테스트용 데이터 입력

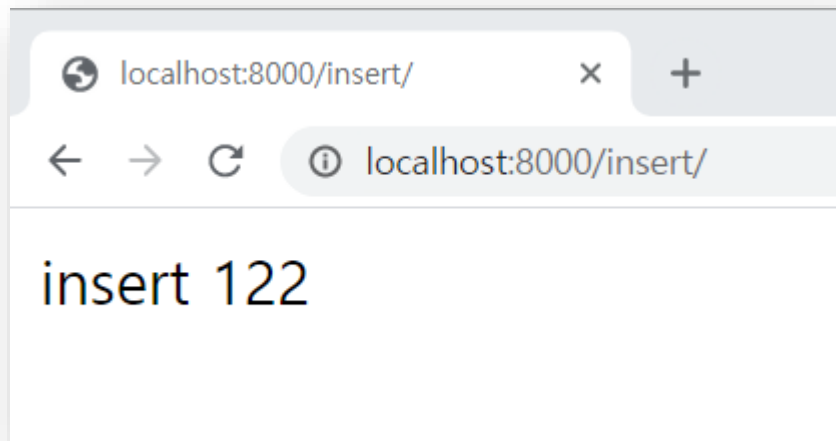
- 서버 구동

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: python ▾

PS E:\dev\python_django\pagination> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 30, 2020 - 14:52:08
Django version 3.0.3, using settings 'config.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
█
```

- 실행



테이블(T): pagination_data

	id ▲	text	cre_date
	필터	필터	필터
1	122	Data-122	2020-08-30 05:52:47.087884
2	121	Data-121	2020-08-30 05:52:47.083896
3	120	Data-120	2020-08-30 05:52:47.078907
4	119	Data-119	2020-08-30 05:52:47.074919
5	118	Data-118	2020-08-30 05:52:47.069932
6	117	Data-117	2020-08-30 05:52:47.065942
7	116	Data-116	2020-08-30 05:52:47.057965
8	115	Data-115	2020-08-30 05:52:47.052983
9	114	Data-114	2020-08-30 05:52:47.044999
10	113	Data-113	2020-08-30 05:52:47.035029
11	112	Data-112	2020-08-30 05:52:47.025058
12	111	Data-111	2020-08-30 05:52:47.009100
13	110	Data-110	2020-08-30 05:52:47.002106
14	109	Data-109	2020-08-30 05:52:46.997774

1 - 14 of 122

■ Paging

● Paginator

- import

```
from django.core.paginator import Paginator
```

- Paginator 적용

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

■ Paging

● Paginator

- 전체 데이터 수

```
p.count
```

- 전체 페이지 수

```
p.num_pages
```

- 페이지 지정

```
p.page(1)
```

- 현재 페이지의 데이터 목록

```
p.page(1).object_list
```

- 현재 페이지의 데이터 시작번호, 종료번호

```
p.page(1).start_index()
```

```
p.page(1).end_index()
```

■ Paging

● Paginator

- 이전 페이지 정보

```
p.page(1).has_previous() # 이전 페이지 존재 유무  
p.page(1).previous_page_number() # 이전 페이지 번호
```

- 다음 페이지 정보

```
p.page(1).has_next() # 다음 페이지 존재 유무  
p.page(1).next_page_number() # 다음 페이지 번호
```

■ Paging

● config/urls.py

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

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

■ Paging

● pagination/views.py

```
from django.core.paginator import Paginator

def pagination(request):
    result = ''
    datas = Data.objects.order_by('-id')
    p = Paginator(datas, 10)
    result += '전체 데이터 수 - %s<br>' % p.count
    result += '전체 페이지 수 - %s<hr>' % p.num_pages

    p = p.page(1)
    list = str(p.object_list).replace('<', '&lt;').replace('>', '&gt;')
    result += '1 페이지 데이터 - %s<hr>' % list

    result += '현재 페이지 데이터 시작번호 - %s<br>' % p.start_index()
    result += '현재 페이지 데이터 종료번호 - %s<hr>' % p.end_index()

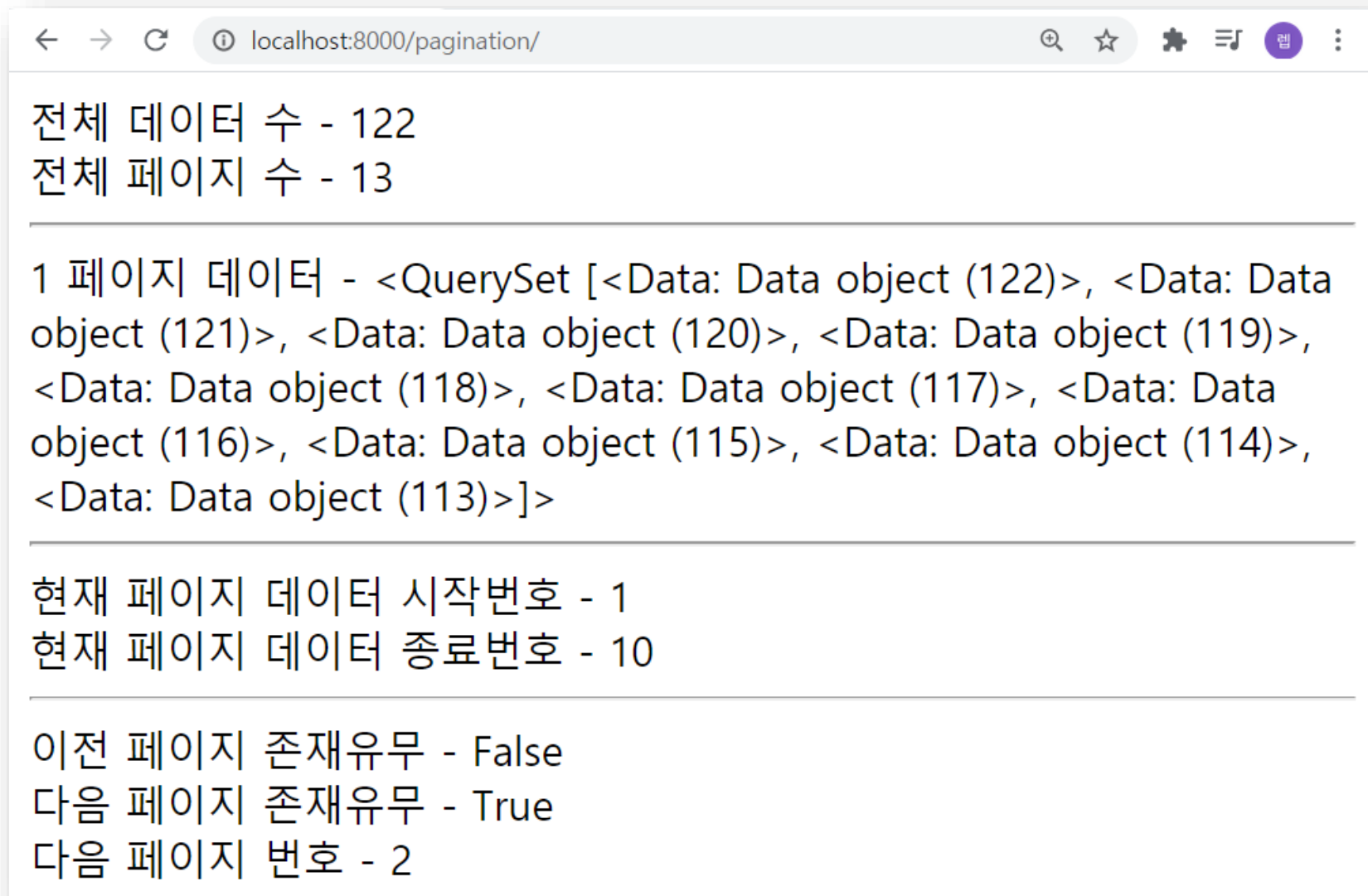
    result += '이전 페이지 존재유무 - %s<br>' % p.has_previous()
    if p.has_previous():
        result += '이전 페이지 번호 - %s<br>' % p.previous_page_number()

    result += '다음 페이지 존재유무 - %s<br>' % p.has_next()
    if p.has_next():
        result += '다음 페이지 번호 - %s<br>' % p.next_page_number()

    return HttpResponse(result)
```

■ Paging

● 실행



■ 템플릿을 사용하여 데이터 출력

● config/urls.py

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

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

■ 템플릿을 사용하여 데이터 출력

● pagination/views.py

```
def list(request):  
    datas = Data.objects.order_by('-id')  
    p = Paginator(datas, 10)  
  
    info = p.page(1)  
  
    context = {  
        'info' : info  
    }  
  
    return render(request, 'list.html', context)
```


■ 템플릿을 사용하여 데이터 출력

● config/settings.py

```
TEMPLATES = [  
    {  
        ...  
        'DIRS': [os.path.join(BASE_DIR, 'templates')],  
        'APP_DIRS': True,  
        ...  
    },  
]
```

■ 템플릿을 사용하여 데이터 출력

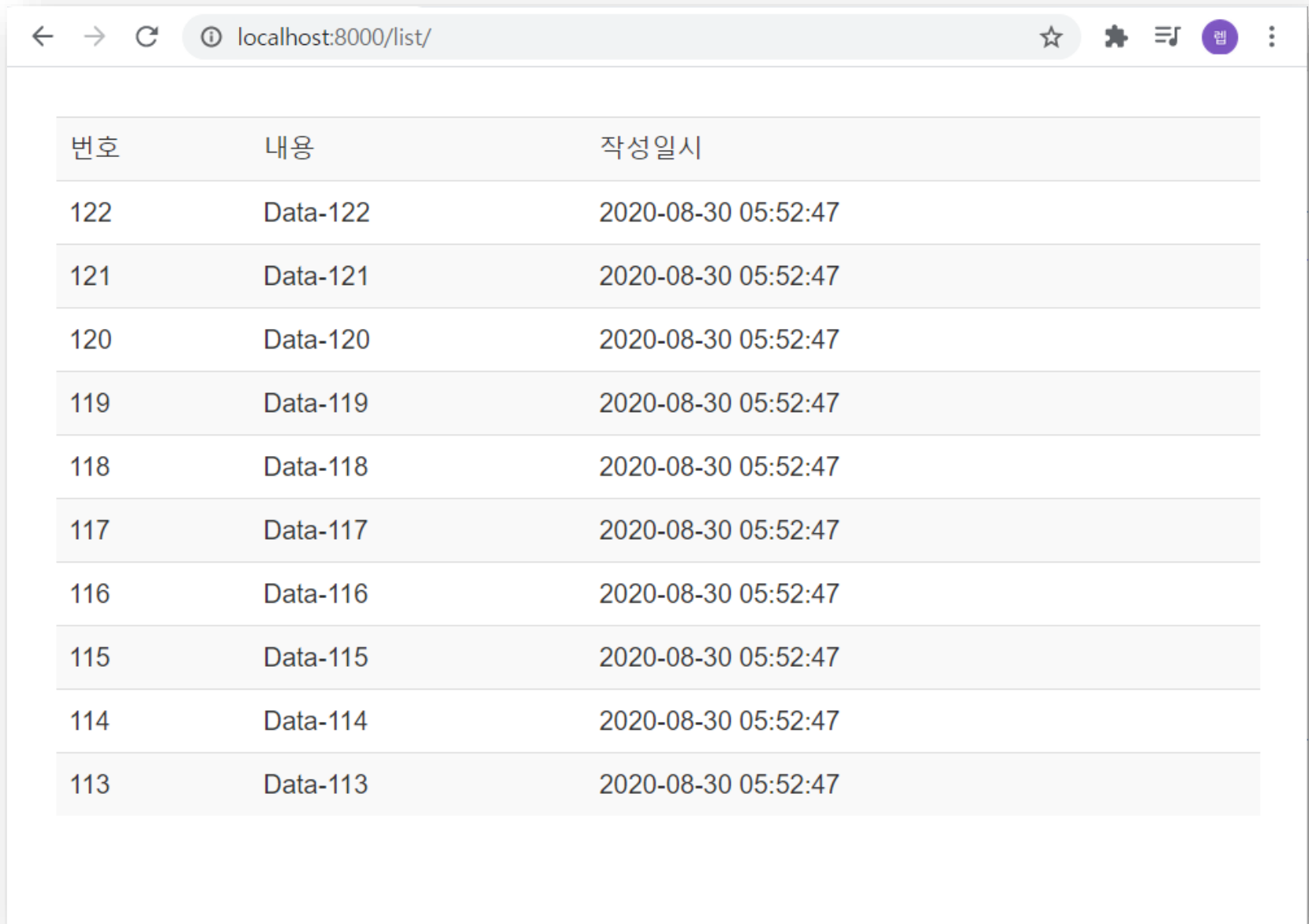
● templates/list.html

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

<body style='margin:30px;'>
  <table class='table table-striped'>
    <tr>
      <td>번호</td><td>내용</td><td>작성일시</td>
    </tr>
    {% for data in info %}
      <tr>
        <td>{{ data.id }}</td>
        <td>{{ data.text }}</td>
        <td>{{ data.cre_date | date:'Y-m-d H:i:s'}}</td>
      </tr>
    {% endfor %}
  </table>
</body>
```

■ 템플릿을 사용하여 데이터 출력

● 실행



A screenshot of a web browser window displaying a list of data. The browser's address bar shows 'localhost:8000/list/'. The page content is a table with three columns: '번호' (Number), '내용' (Content), and '작성일시' (Creation Date/Time). The table contains 10 rows of data, with numbers ranging from 122 down to 113. Each row has a corresponding 'Data-' prefix and a timestamp of '2020-08-30 05:52:47'.

번호	내용	작성일시
122	Data-122	2020-08-30 05:52:47
121	Data-121	2020-08-30 05:52:47
120	Data-120	2020-08-30 05:52:47
119	Data-119	2020-08-30 05:52:47
118	Data-118	2020-08-30 05:52:47
117	Data-117	2020-08-30 05:52:47
116	Data-116	2020-08-30 05:52:47
115	Data-115	2020-08-30 05:52:47
114	Data-114	2020-08-30 05:52:47
113	Data-113	2020-08-30 05:52:47

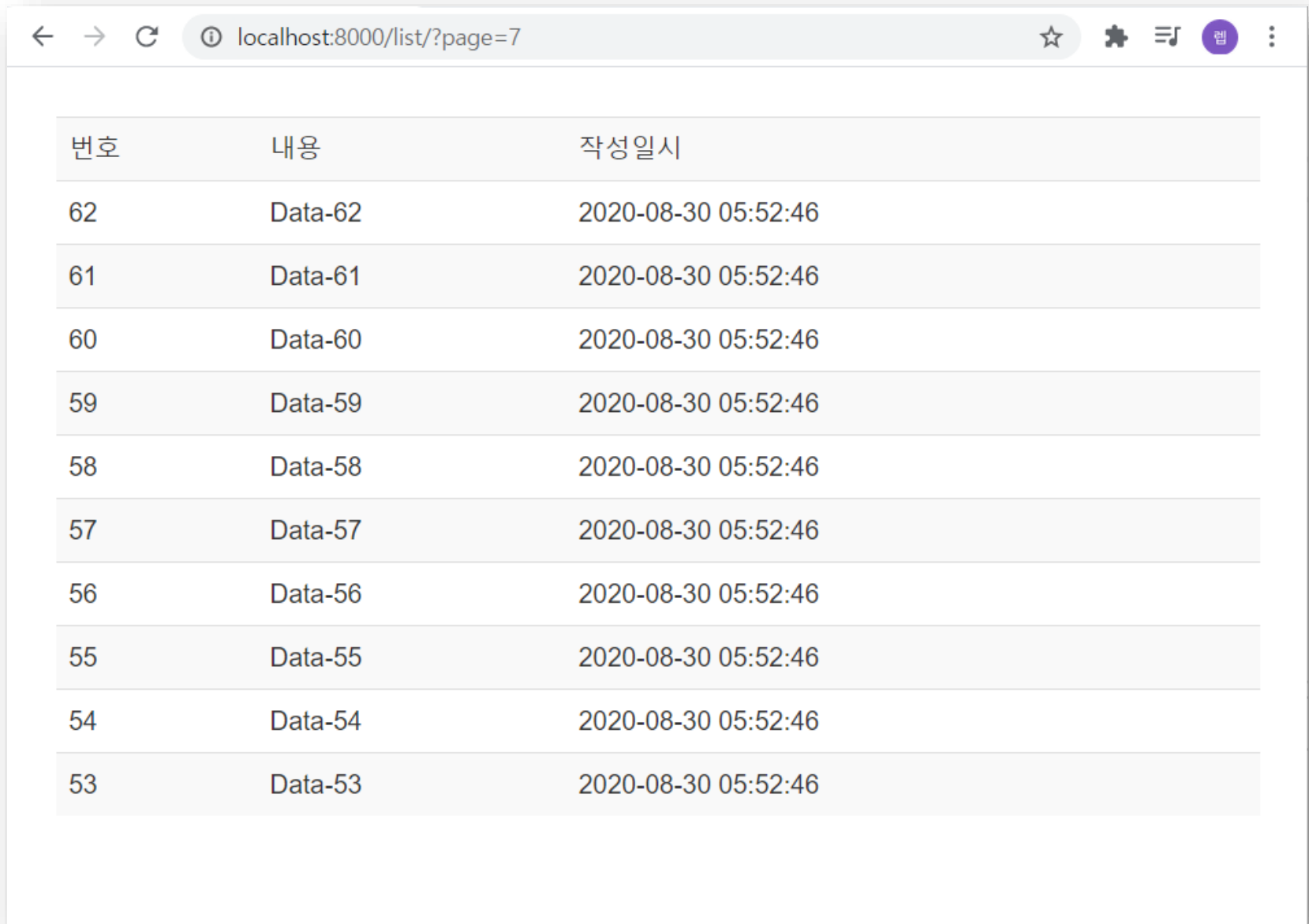
■ 페이지 지정 호출

● pagination/views.py

```
def list(request):  
    page = request.GET.get('page')  
  
    if not page:  
        page = 1  
  
    datas = Data.objects.order_by('-id')  
    p = Paginator(datas, 10)  
  
    info = p.page(page)  
  
    context = {  
        'info' : info  
    }  
  
    return render(request, 'list.html', context)
```

■ 페이지 지정 호출

● 실행



A screenshot of a web browser window. The address bar shows 'localhost:8000/list/?page=7'. The browser interface includes back, forward, and refresh buttons, as well as icons for bookmarks, extensions, and a menu. The main content area displays a table with three columns: '번호' (Number), '내용' (Content), and '작성일시' (Creation Date/Time). The table contains 10 rows of data, with numbers ranging from 62 down to 53. Each row has a corresponding 'Data-' prefix and a timestamp of '2020-08-30 05:52:46'.

번호	내용	작성일시
62	Data-62	2020-08-30 05:52:46
61	Data-61	2020-08-30 05:52:46
60	Data-60	2020-08-30 05:52:46
59	Data-59	2020-08-30 05:52:46
58	Data-58	2020-08-30 05:52:46
57	Data-57	2020-08-30 05:52:46
56	Data-56	2020-08-30 05:52:46
55	Data-55	2020-08-30 05:52:46
54	Data-54	2020-08-30 05:52:46
53	Data-53	2020-08-30 05:52:46

■ 페이지 네비게이션 구현

● pagination/views.py

```
def list(request):  
    ...  
  
    info = p.page(page)  
  
    start_page = (int(page) - 1) // 10 * 10 + 1  
    end_page = start_page + 9  
  
    if end_page > p.num_pages:  
        end_page = p.num_pages  
  
    context = {  
        'info' : info,  
        'pagination' : range(start_page, end_page + 1)  
    }  
  
    return render(request, 'list.html', context)
```

■ 페이지 네비게이션 구현

● templates/list.html

```
<link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">

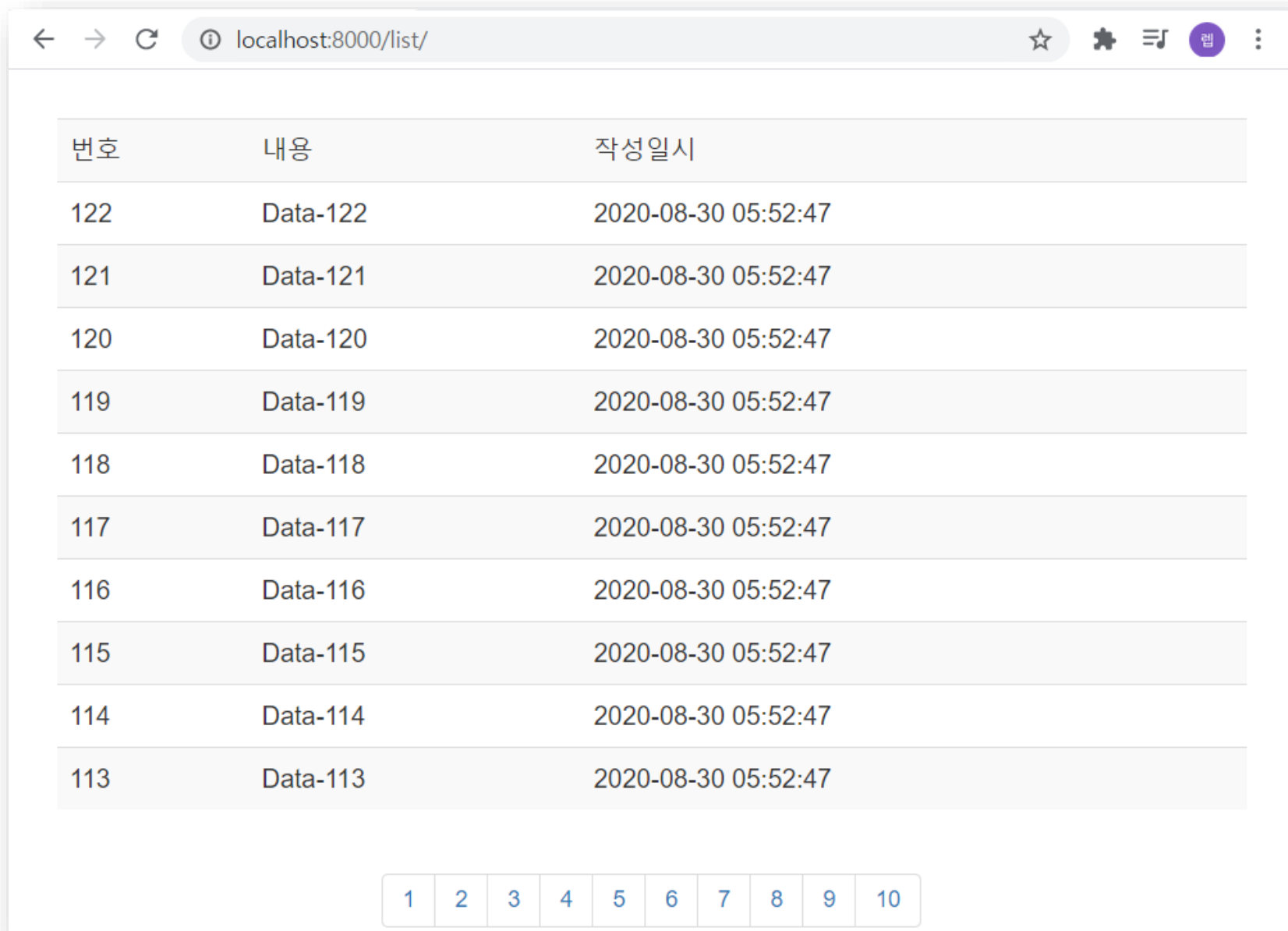
<body style='margin:30px;'>
  <table class='table table-striped'>
    ...
  </table>

  <div style='text-align:center;'>
    <ul class='pagination'>
      {% for i in pagination %}
      <li>
        <a href='/list/?page={{ i }}'>{{ i }}</a>
      </li>
      {% endfor %}
    </ul>
  </div>

</body>
```

■ 페이지 네비게이션 구현

● 실행



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/list/'. The page content consists of a table with three columns: '번호' (Number), '내용' (Content), and '작성일시' (Creation Date/Time). The table lists 10 items, numbered 122 down to 113, each with the content 'Data-122' through 'Data-113' and the timestamp '2020-08-30 05:52:47'. At the bottom of the page, there is a pagination bar with buttons for pages 1 through 10.

번호	내용	작성일시
122	Data-122	2020-08-30 05:52:47
121	Data-121	2020-08-30 05:52:47
120	Data-120	2020-08-30 05:52:47
119	Data-119	2020-08-30 05:52:47
118	Data-118	2020-08-30 05:52:47
117	Data-117	2020-08-30 05:52:47
116	Data-116	2020-08-30 05:52:47
115	Data-115	2020-08-30 05:52:47
114	Data-114	2020-08-30 05:52:47
113	Data-113	2020-08-30 05:52:47

1 2 3 4 5 6 7 8 9 10