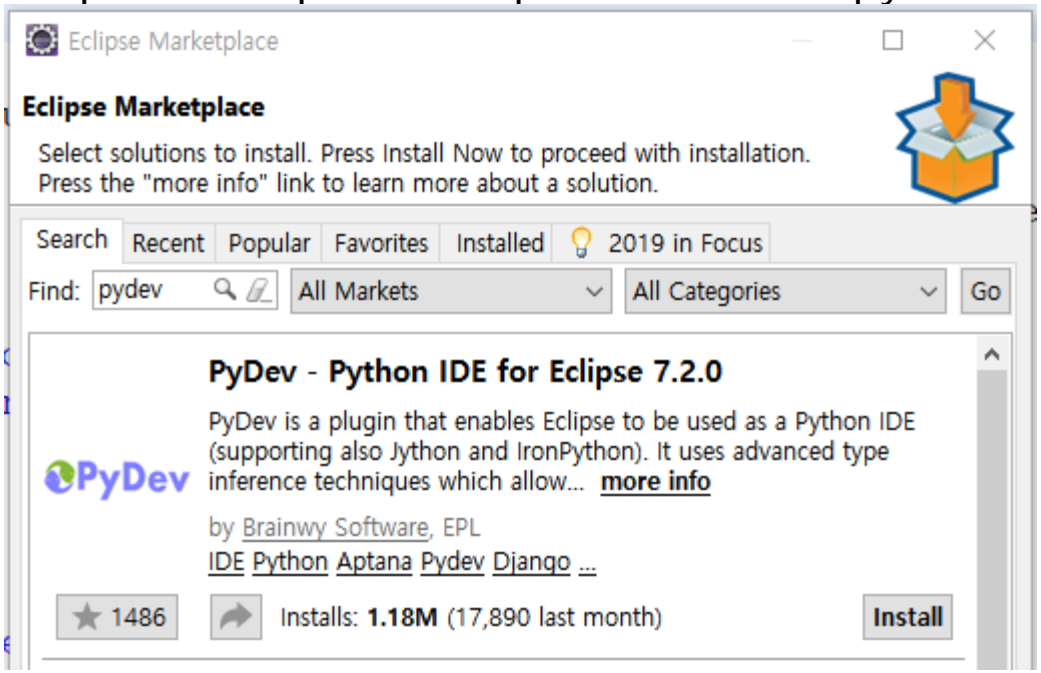
Django

1. pyDev 설치
2. 프로젝트 작성
3. 북마크앱
4. 설문조사
5. 게시판

1. pyDev 설치

- **Python 개발 툴**
  - ipython notebook-웹브라우저에서 실행
  - pyCham-python전용 개발툴(IntelliJ기반)
  - PyDev- 이클립스 플러그인

- **PyDev 설치**
  - eclipse-> Help->Maketplace 메뉴에서 pydev 검색하여 설치



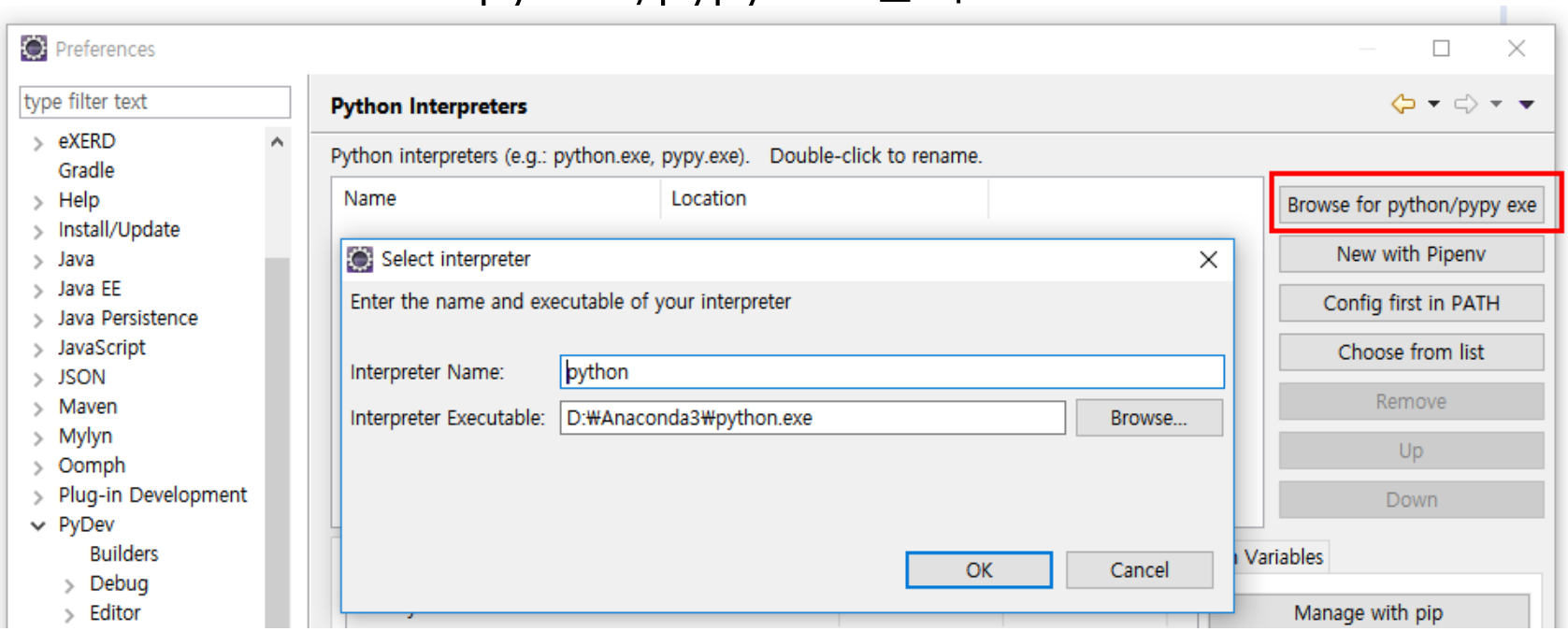
1. pyDev 설치

환경설정

- eclipse 재 실행 후 open perspactive 선택->pyDev선택

파이썬 인터프리터 설정

- window->Preference->PyDev-interpereters->python interpereter ->Browse for python/pypy.exe 선택



2. 프로젝트 작성

프로젝트 만들기

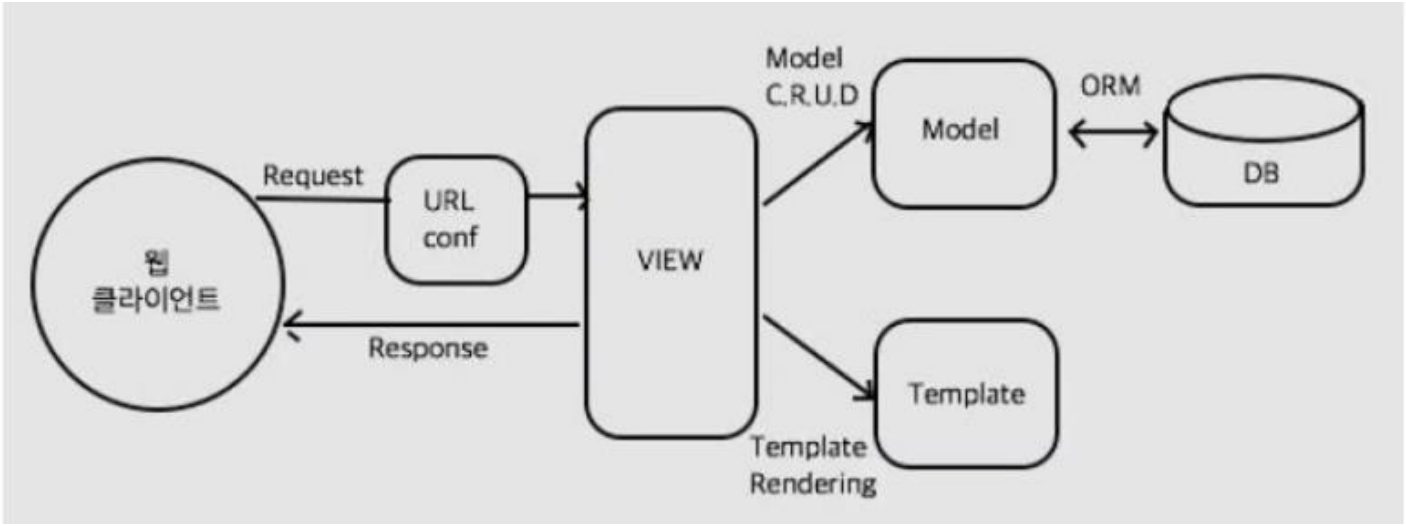
- django 설치 : pip install django
- New->Other-PyDev-PyDev Django Project
- 프로젝트 이름 : mysite
- pyweb01 디렉토리가 2개 만들어진다
- d:₩pgm₩django_work₩mysite
- d:₩python₩work₩mysite₩mysite
  - python 웹프로젝트의 설정 디렉토리

2. 프로젝트 작성

- **Django**
  - 파이썬 기반의 무료 오픈소스 웹 애플리케이션 프레임워크(Open Source Web Application framework)
  - 장점
    - Python 기반 프레임워크로 배우기 쉬움
    - 빠른 개발속도, 개발 비용 절감
    - 코드 완성도를 높게 유지할수 있으며 확장성이 좋음
    - 사용자 인증, 사용자관리 등 기능이 기본적으로 구현되어 있음
  - 성공적인 도입 사례 – Instagram
  - Djang 패키지 설치
    - pip install django

2. 프로젝트 작성

- MTV 패턴



- MVC pattern과 MTV pattern의 비교

MVC	MTV	설명
Model	Model	데이터베이스와 관련된 처리를 담당하는 코드
View	Template	사용자가 보게되는 화면을 정의하는 코드
Controller	View	데이터를 처리한 후 결과를 템플릿에게 전달하는 코드

3. 북마크 앱

애플리케이션 설계하기

화면 UI 설계

- 화면설계는 주로 템플릿 코딩에 반영되고, templates/ 디렉토리 하위의 *.html 파일에 코딩
- 리스트 - bookmark_list.html
- 상세페이지 - bookmark_detail.html

테이블 설계

- 테이블 설계 내용은 모델 코딩에 반영되고, models.py 파일에 코딩

필드명	타입	제약 조건	설명
id	integer	PK, Auto Increment	Primary key
title	CharField(100)	Blank,Null	북마크제목
url	URLField	Unique	북마크 URL

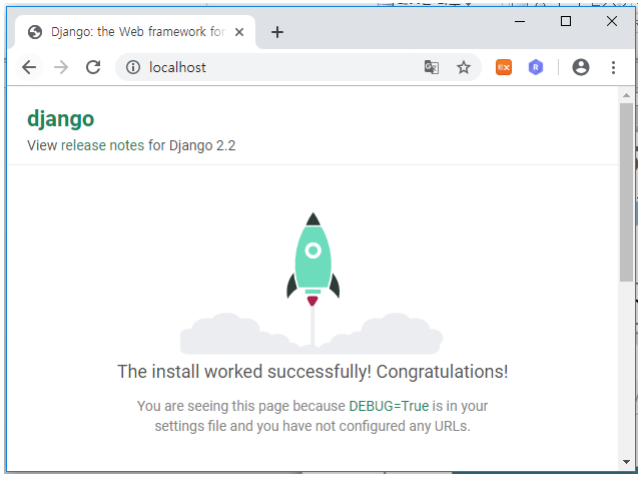
3. 북마크 앱

- **웹서버 구동**

```
python manage.py runserver localhost:80
```

- **웹브라우저에서 확인**

```
http://localhost
```



3. 북마크 앱

애플리케이션 설계

로직 설계

- 로직 설계는 처리 흐름을 설계하는 것
- 웹 프로그래밍에서는 URL을 받아서 최종 HTML템플릿 파일을 만드는 과정이 하나의 로직
- 리다이렉션이나 템플릿 파일에서 URL요청이 발생하는 일련의 과정을 모두 고려하여 표현하는 것

URL	View	Template
/bookmark/	BookmarkLV.as_view()	bookmark_list.html
/bookmark/99/	BookmarkDV.as_view()	bookmark_detail.html
Bookmark 앱 URL-뷰-템플릿 간의 처리흐름		

3. 북마크 앱

애플리케이션 설계

URL 설계

- URL설계 내용은 URLconf 코딩에 반영
- urls.py파일에 코딩
- URL패턴, 뷰 이름, 템플릿 파일 이름 정의
- 뷰에서 어떤 제네릭 뷰를 사용할 것인지 정의

URL패턴	뷰 이름	템플릿 파일 이름
/bookmark/	BookmarkLV(ListView)	bookmark_list.html
/bookmark/??/*	BookmarkLV(DetailView)	bookmark_detail.html
/admin/	(Django 제공기능)	
**DetailView의 ??자리에는 PK인 id가 전달되어 들어간다		

3. 북마크 앱

작업/코딩 순서

작업순서	관련 명령/파일	필요한 작업 내용
뼈대만들기	startproject	mysite 프로젝트 생성
뼈대만들기	settings.py	프로젝트 설정 항목 변경
뼈대만들기	migrate	User/Group 테이블 생성
뼈대만들기	createsuperiser	프로젝트 관리자인 슈퍼유저를 만든다
뼈대만들기	startapp	북마크 앱 생성
뼈대만들기	settings.py	북마크 앱 등록
모델코딩하기	models.py	모델(테이블) 정의
모델코딩하기	admin.py	Admin 사이트에 모델 등록
모델코딩하기	makemigrations	모델을 데이터베이스에 반영
모델코딩하기	migrate	

3. 북마크 앱

작업/코딩 순서

URLconf 코딩하기	urls.py	URL 정의
뷰 코딩하기	views.py	뷰 로직 작성
템플릿 코딩하기	templates 디렉토리	템플릿 파일 작성
그 외 코딩하기	-	(없음)

3. 북마크 앱

프로젝트 생성

```
cd cd pgm/python_work/mysite

python manage.py startapp bookmark
```

bookmark 관련 디렉토리가 만들어짐
eclipse에서 F5를 눌러 bookmark 패키지(디렉토리) 확인

3. 북마크 앱

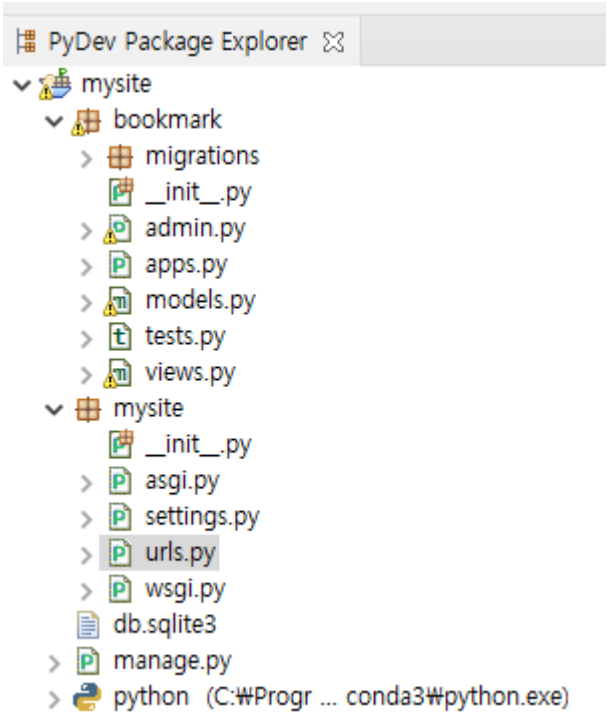
- 프로젝트 설정 파일 변경(setting.py 설정)

1. 데이터베이스 설정
2. 템플릿 설정
3. 정적파일(static url, staticfiles_dirs)
4. 타임존 지정 (UTC->Asia/Seoul)
5. 미디어 관련사항 설정(파일 업로드 기능 개발시 필요)
6. 애플리케이션 등록(setting.py)
7. 기타 설정 - 언어 등등

3. 북마크 앱

기본 테이블 생성

```
cd pgm/python_work/mysite  # 프로젝트 폴더로 위치이동
python manage.py migrate  # 기본 테이블 생성
```



3. 북마크 앱

슈퍼 유저 생성

```
cd pgm/python_work/mysite  # 프로젝트 폴더로 위치이동
python manage.py createsuperuser  #슈퍼유저 생성
```

username : admin
email :생략 가능
password : admin1234

3. 북마크 앱

애플리케이션 등록(setting.py) 및 언어 설정

- setting.py 수정
  - mysite/settings.py 열어서 다음과 같이 작성

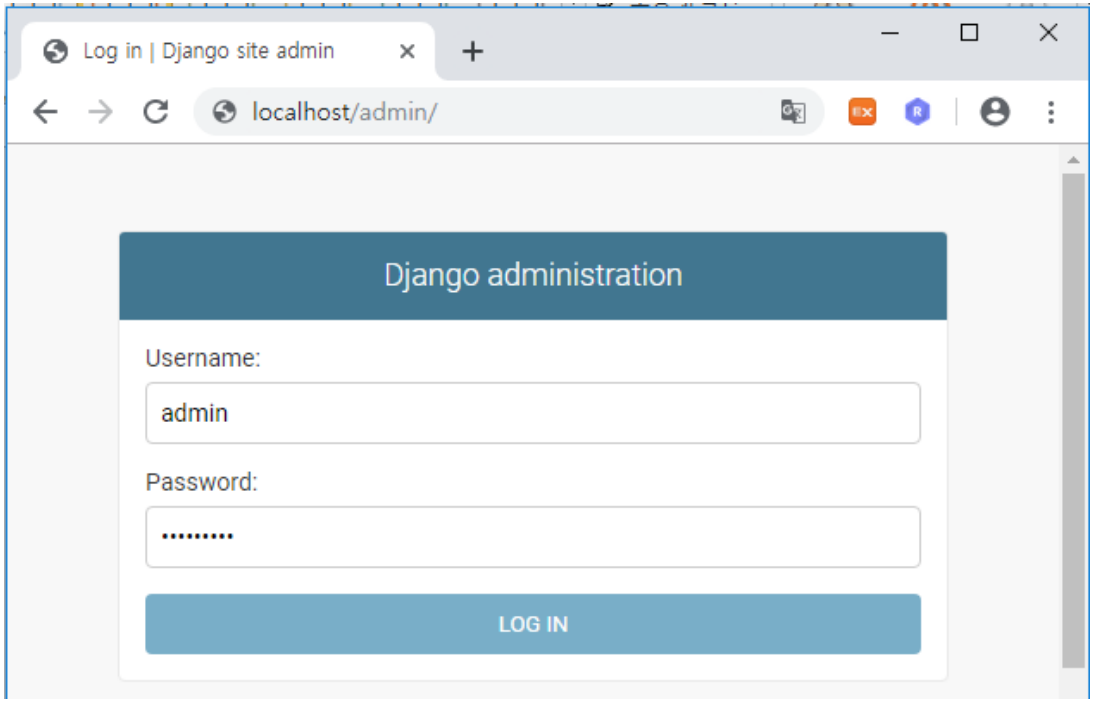
```
# Application definition

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

```
# Internationalization 이하 생략

LANGUAGE_CODE = 'ko'

TIME_ZONE = 'Asia/Seoul'

USE_I18N = True

USE_L10N = True

USE_TZ = True
```

3. 북마크 앱

- 웹브라우저에서 확인

http://localhost/admin



3. 북마크 앱

- **models.py 설정**
  - 테이블을 새로 만들면 models.py와 admin.py 2개의 파일을 수정해야 함.
  - models.py: 테이블에 대한 모델 클래스 정의
  - admin.py : models.py에 등록한 테이블이 admin 사이트에서도 보이도록 처리

3. 북마크 앱

- **bookmark/models.py 작성**
  - 테이블을 하나의 클래스로 정의하고 테이블의 컬럼은 클래스의 변수로 매핑
  - 테이블 클래스는 django.db.models.Model 클래스를 상속받아 정의
  - 변수 자료형도 장고에서 미리 정의된 자료형을 사용

```
from django.db import models

# Create your models here.
class Bookmark(models.Model):  #djang의 Model 클래스 상속받음
    #필드 선언,      #blank 빈값 허용 여부, null null 허용
    title=models.CharField(max_length=100, blank=True,null=True)
    #unique "primary key
    url=models.URLField("url",unique=True)

    #객체를 문자열로 표현하는 함수
    def __str__(self):
        return self.title
```

3. 북마크 앱

- **bookmark/admin.py 작성**

```
from django.contrib import admin
from bookmark.models import Bookmark

# Register your models here.
#관리자 사이트에서 Bookmark 클래스 출력 모양 정의하는 코드
@admin.register(Bookmark)
class BookmarkAdmin(admin.ModelAdmin):
    #관리자 화면에 출력할 필드 목록(튜플 형식)
    list_display=("id", "title", "url")

 #Bookmark 클래스와 BookmaarAdmin 클래스를 등록
#admin.site.register(Bookmark,BookmarkAdmin) @admin.register(Bookmark)와 동일
```

3. 북마크 앱

- **데이터베이스 변경 사항 반영**

```
cd python₩work/mysite

python manage.py makemigrations
python manage.py migrate
```

- **웹서버 재 구동**

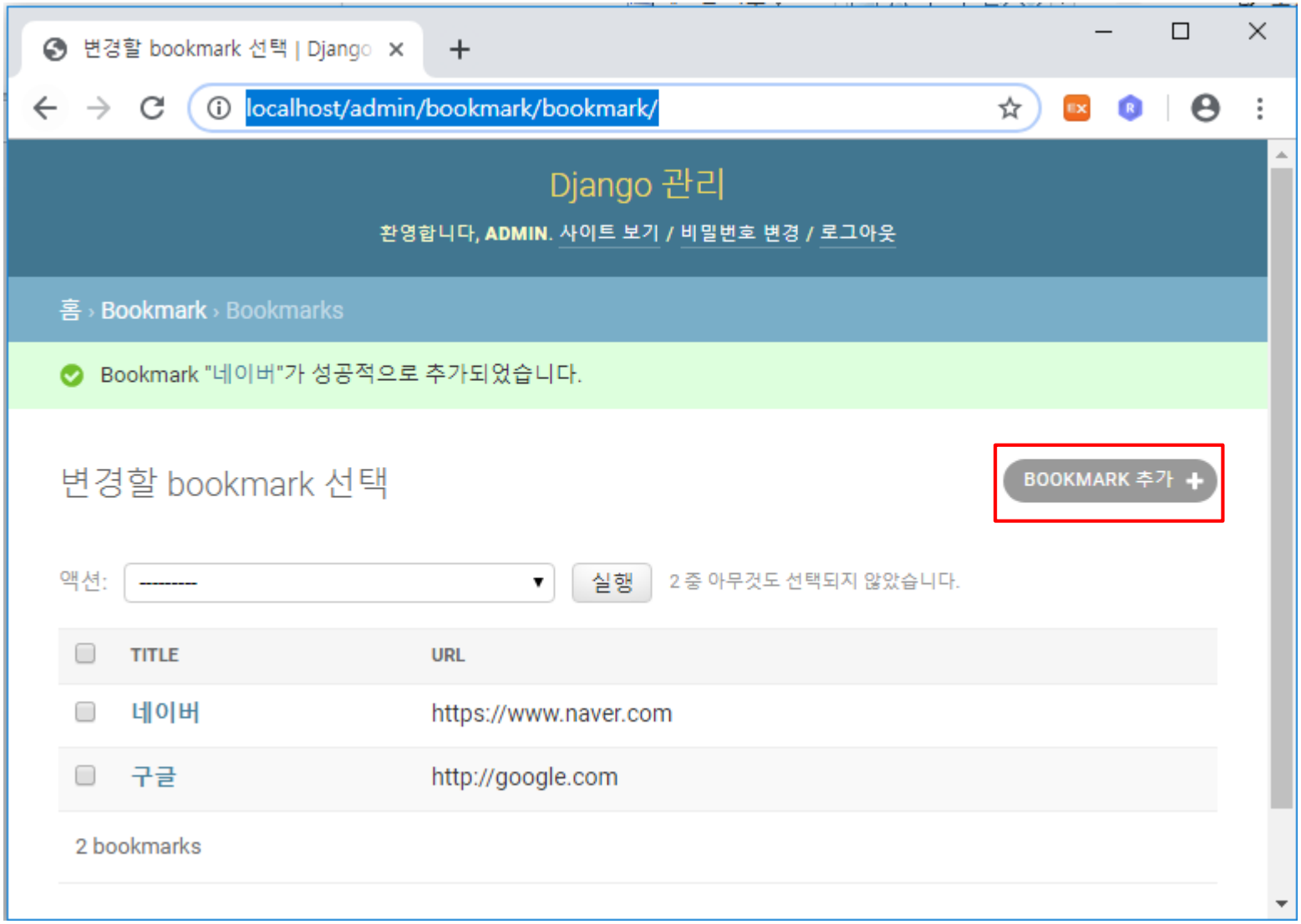
```
python manage.py runserver localhost:80
```

- **브라우즈 확인**

```
Localhost/admin
```

3. 북마크 앱

- localhost/admin 사이트 데이터 추가



3. 북마크 앱

클래스 방식 view

```
from bookmark.models import Bookmark
from django.views.generic import ListView, DetailView
# Create your views here.
#--- ListView #템플릿파일 : 모델명소문자_list.html
class BookmarkLV(ListView):
    model=Bookmark

#--- DetailView
#템플릿파일 : 모델명소문자_detail.html
class BookmarkDV(DetailView):
    model=Bookmark
```

3. 북마크 앱

- **URLconf(urls.py) : path() 사용**

```
from django.contrib import admin
from django.urls import path
from django.views.generic import ListView,DeleteView
from bookmark.models import Bookmark
from django.views.generic.detail import DetailView

urlpatterns = [
    path('admin/', admin.site.urls),
    path('bookmark/', ListView.as_view(model=Bookmark),name='index'),
    path('bookmark/<int:pk>/', DetailView.as_view(model=Bookmark), name='detail'),
 ]
```

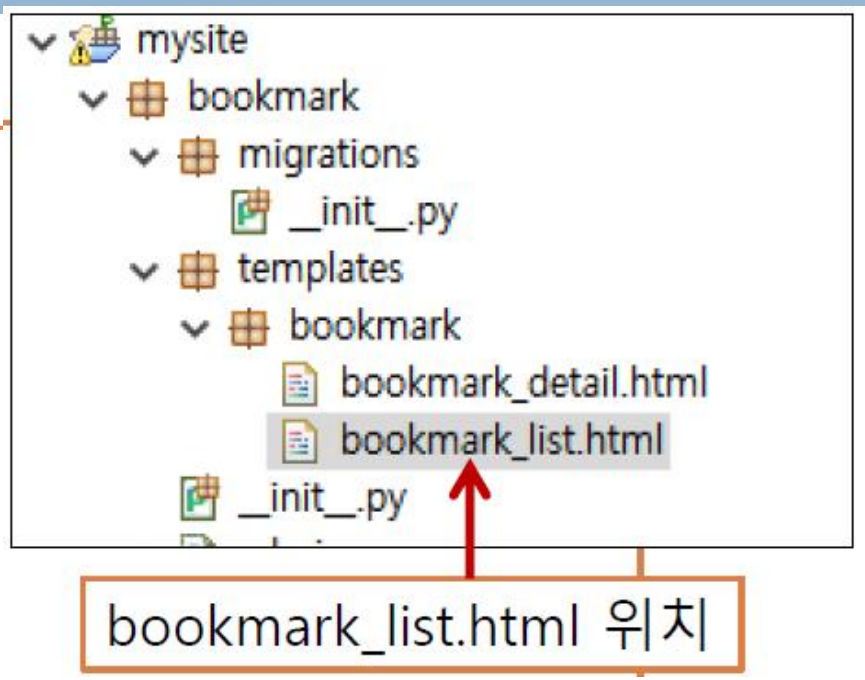
3. 북마크 앱

- **URLconf(urls.py) : url() 사용**

```
from django.contrib import admin
from bookmark.models import BookmarkLV, BookmarkDV
from django.conf.urls import url
from django.contrib import admin

urlpatterns = [
    url(r'^admin/', admin.site.urls),

    #Class-based views for Bookmark app
    url(r'^bookmark/$', BookmarkLV.as_view(), name='index'),
    url(r'^bookmark/(?P<pk>₩d+)/$', BookmarkDV.as_view(), name='detail'),
]
```

3. 북마크 앱

- **bookmark_list.html**




```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
</head>
<body>
  <div id="containter">
    <h1>Bookmark List</h1>
    <ul>
      {% for bookmark in object_list %}
        <li><a href="{%url 'detail' bookmark.id%}">{{bookmark}}</a></li>
      {% endfor %}
    </ul>
  </div>
</body>
</html>
```

3. 북마크 앱

- **bookmark_detail.html**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Insert title here</title>
</head>
<body>
  <div id="content">
    <h1>{{object.title}}</h1>
    <ul>
      <li>URL : <a href="{{object.url}}">{{object.url}}</a></li>
    </ul>
  </div>
</body>
</html>
```

3. 북마크 앱

Views와 urls

- method 방식(views.py)

```
from django.shortcuts import render
from bookmark.models import Bookmark

# Create your views here.
def home(request):
    #select * from bookmark_bookmark order by title
    urlList=Bookmark.objects.order_by("title") #-title 내림차순 정렬
    #select count(*) from bookmark_bookmark
    urlCount =Bookmark.objects.all().count()
    #list.html 페이지로 넘어가서 출력됨
    #rander("url",{"변수명":값,"변수명":값})
    return render(request,"bookmark/list.html",
                  {"urlList":urlList, "urlCount":urlCount})
```

3. 북마크 앱

- **urls.py**

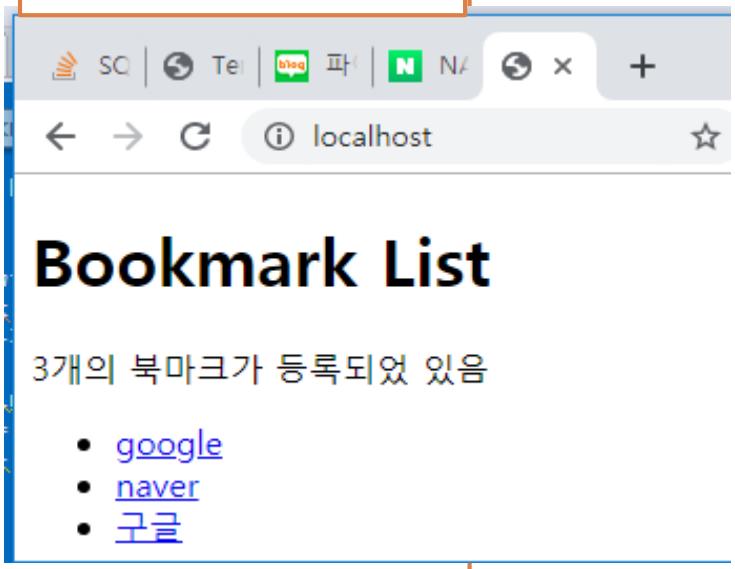
```
from django.contrib import admin
from django.urls import path
from django.conf.urls import url
from bookmark import views

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

    # r정규표현식, ^ 시작, $끝
    # http://localhost로 요청하면 views 모듈의 home 함수 실행
    url(r"^$",views.home()),
    #path( '',views.home),
]
```

3. 북마크 앱: templates 작성

- **list.html 작성(bookmark/templates/bookmark에 작성)**

```
<!DOCTYPE html>
<html>
<head>
<title>Insert title here</title>
</head>
<body>
<div id="containter">
      <h1>Bookmark List</h1>
      {{urlCount}}개의 북마크가 등록되었 있음
      <ul>
            {% for row in urlList %}
               <li><a href="detail?url={{row.url}}">{{row.title}}</a></li>
            {% endfor %}
      </ul>
</div>
</body>
</html>
```

localhost 실행결과



3. 북마크 앱: templates 작성

- **detail.html 페이지 작성**
  - views.py에 detail 메소드 작성

```
from django.shortcuts import render, render_to_response
from bookmark.models import Bookmark

# Create your views here.
def home(request):
    #....생략

def detail(request):
    #get 방식 변수 받아오기 request.GET["변수명"]
    #post 방식 변수 받아오기 request.POST["변수명"]
    addr=request.GET["url"]
    #select * from bookmark_bookmark where url="..."
    dto=Bookmark.objects.get(url=addr)
    #detail.html로 포워딩
    return render(request, "bookmark/detail.html", {"dto":dto})
```

3. 북마크 앱: templates 작성

- **detail.html 페이지 작성**
  - urls.py에 detail url 추가

```
from django.contrib import admin
from django.urls import path
from django.conf.urls import url
from bookmark import views

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

    # r정규표현식, ^ 시작, $끝
    # http://localhost로 요청하면 views 모듈의 home 함수 실행
    url(r"^$",views.home),

    # http://localhost/detail로 요청하면 views 모듈의 home 함수 실행
    url(r"^detail$",views.detail),
    #path("detail/",views.detail),
]
```

3. 북마크 앱: templates 작성

- **detail.html 작성**

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>detail page</title>
</head>
<body>
  <div id="container">
    <h1>{{dto.title}}</h1>
    <ul>
      <li>URL:<a href="{{dto.url}}">{{dto.url}}</a></li>
    </ul>
  </div>
</body>
</html>
```