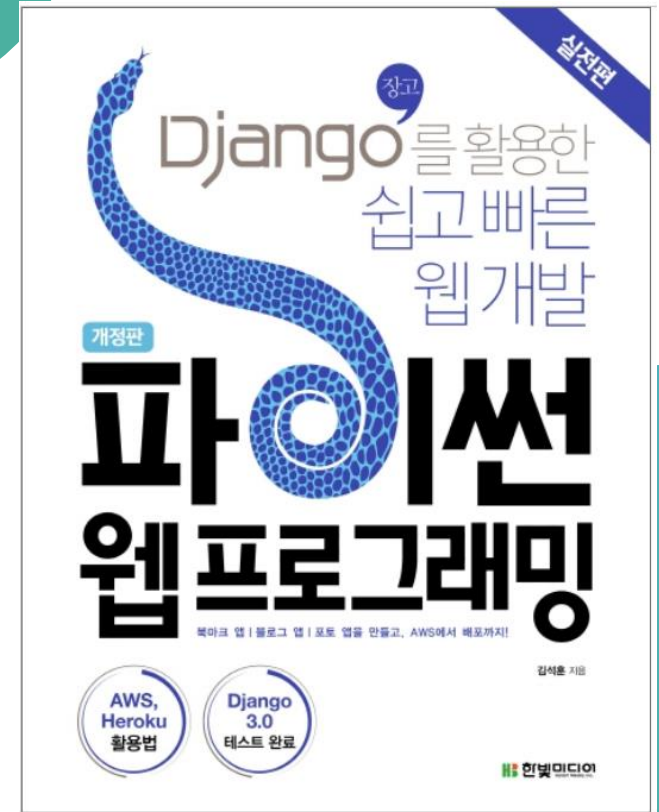


- ▶ CHAPTER 01 장고 개발의 기본 사항 및 가상 환경 사용하기
- ▶ CHAPTER 02 실전 프로그램 개발 – Bookmark 앱

파이썬 웹프로그래밍



가천대학교 컴퓨터공학과
왕보현

시작하기전에

- 개발 환경

- Anaconda
- Django 3.0.8
- Python 3.6.10 Anaconda
- Windows 10
- 장고 공식 홈페이지
<https://docs.djangoproject.com/>
(오픈소스)

- 예제 파일 다운로드

- <http://www.hanbit.co.kr/src/10104>



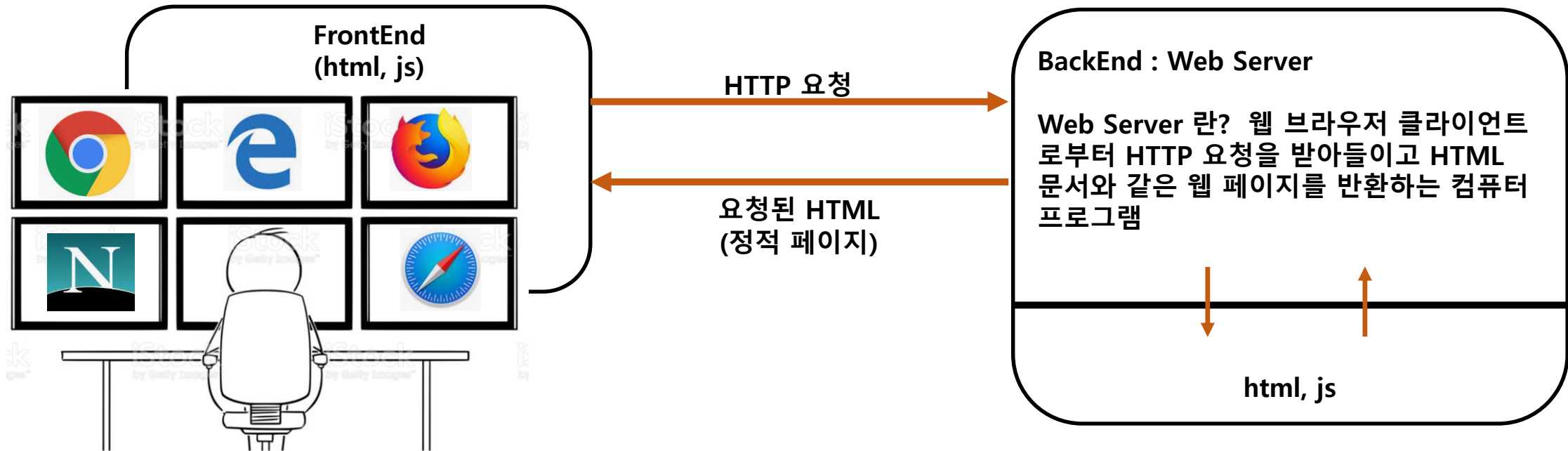
CHAPTER 01 장고 개발의 기본 사항 및 가상 환경 사용하기

CHAPTER 02 실전 프로그램 개발-Bookmark 앱

웹데이터베이스 란?



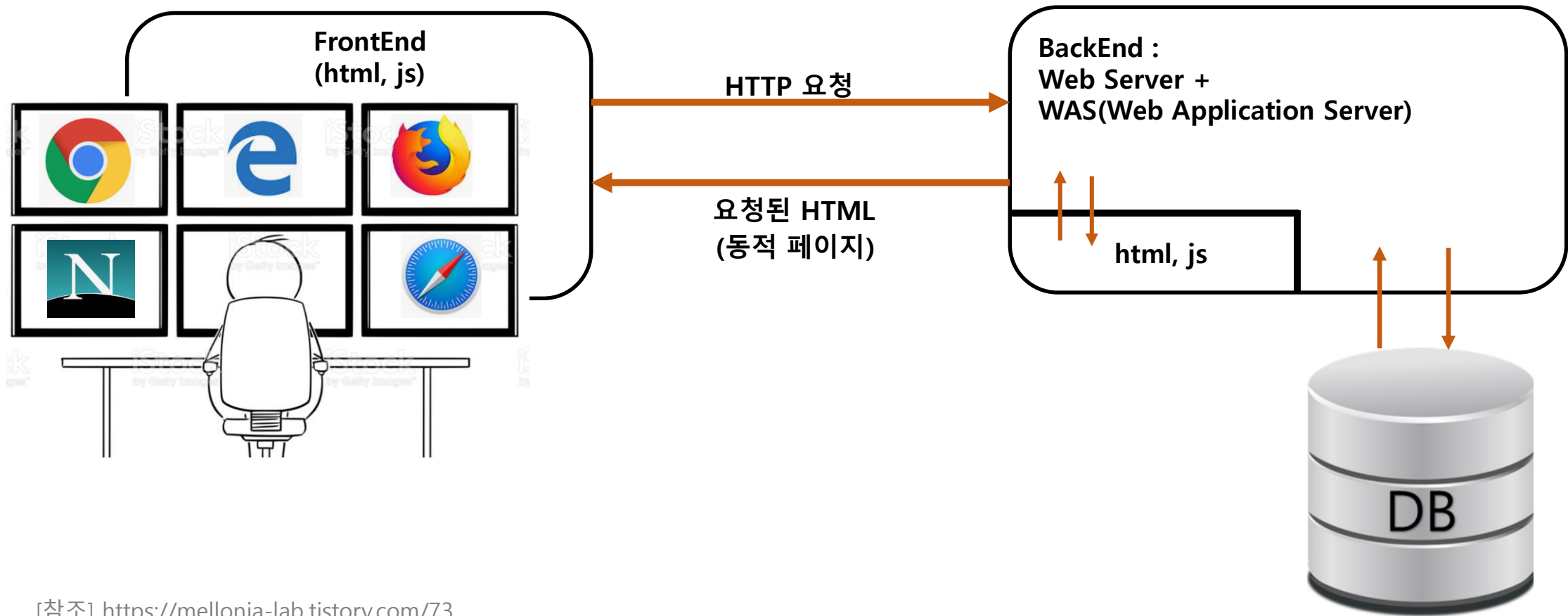
[웹데이터베이스 이전]



[참조] <https://mellonia-lab.tistory.com/73>

웹데이터베이스 란?

[웹데이터베이스 시대]



[참조] <https://mellonia-lab.tistory.com/73>



웹데이터베이스 란?

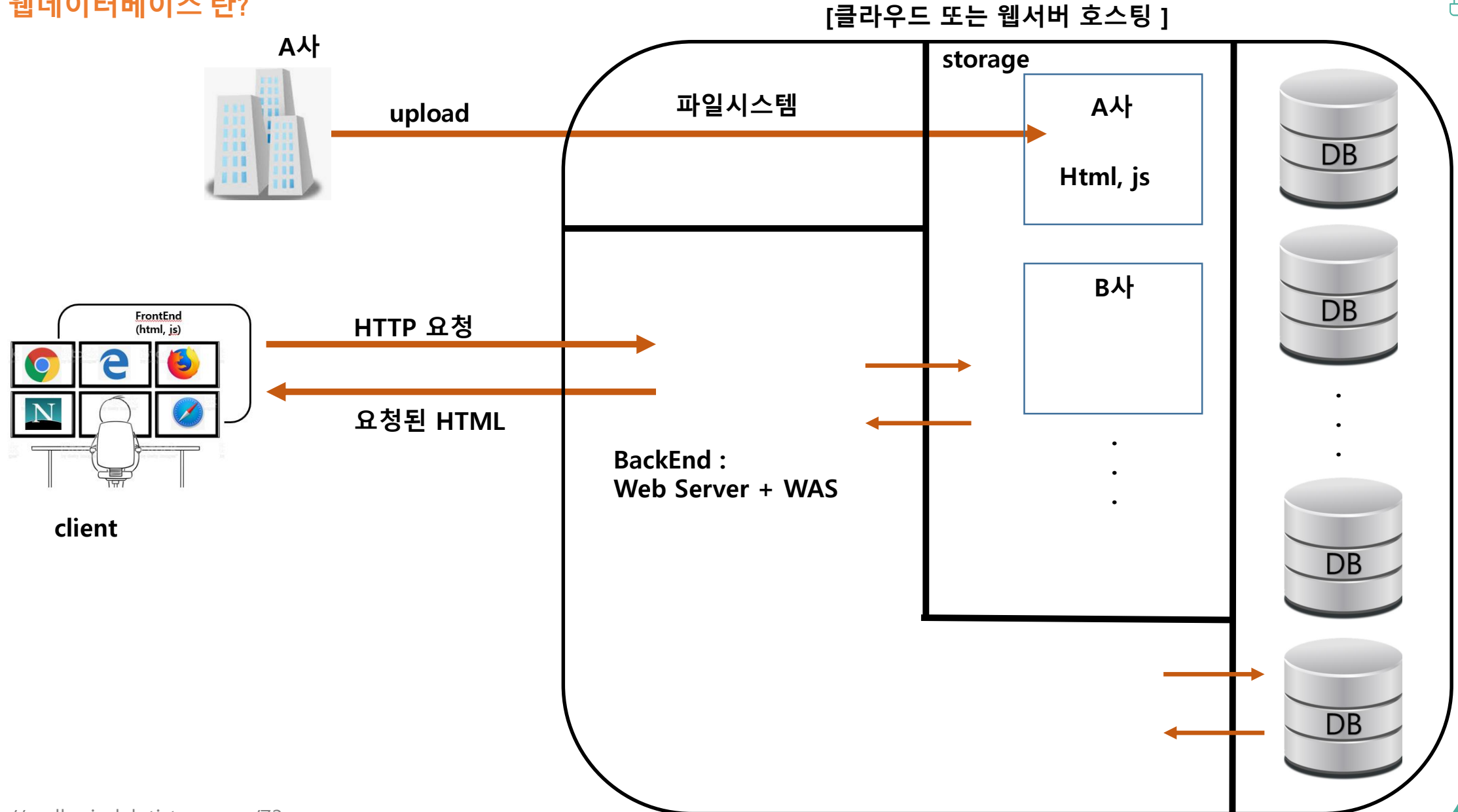
[웹데이터베이스 이전]

WAS 란? 인터넷 상에서 HTTP 프로토콜을 통해 사용자 컴퓨터나 장치에 애플리케이션을 수행해 주는 미들웨어. 주로 동적 서버 콘텐츠를 수행하는 것으로 일반적인 웹서버와 구별이 되며, 주로 데이터베이스 서버와 같이 수행된다.

- * WAS : 웹서버로는 처리할 수 없는 데이터베이스 조회나 다양한 로직 처리가 필요한 동적 콘텐츠를 제공함.
- WAS와 연계된 언어 : PHP, JSP, ASP, Python, Node.js, Django의 template 언어
- * 대표적인 WAS : Tomcat , IIS, Web sphere, Web Logic, Django
- * WAS에 웹서버가 포함된 경우도 있지만 대규모 웹 서비스를 제공한다면 WAS와 웹서버를 별도로 사용

[참조] <https://mellonia-lab.tistory.com/73>

웹데이터베이스 란?

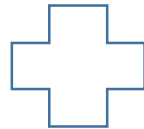
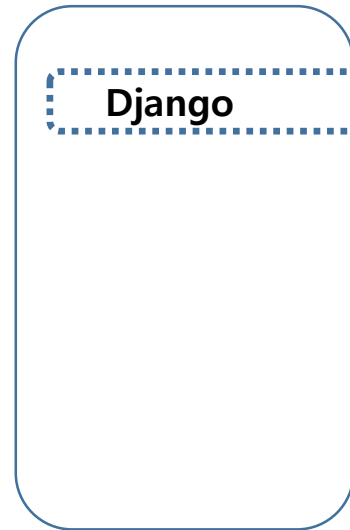


[참조] <https://mellonia-lab.tistory.com/73>

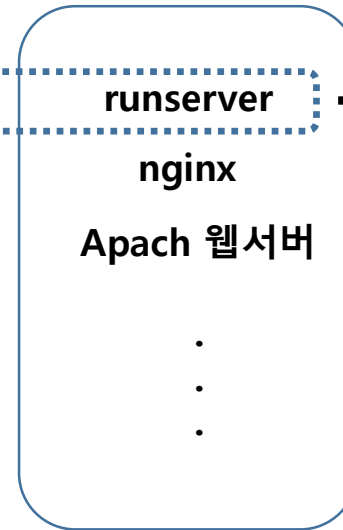
> > 파이썬 웹 프로그래밍



Web Application Server 또는
Web Application Framework



Web Server



→ 수업



왜 Django인가?

| Django란? | [출처] <https://developer.mozilla.org/ko/docs/Learn/Server-side/Django/Introduction>

보안이 우수하고 유지보수가 편리한 웹사이트를 신속하게 개발하는 하도록 도움을 주는 파이썬 웹 프레임워크
구글 앱 엔진에서 django를 사용하게 되면서 많은 사람들이 사용하게 되었음 [출처] <https://wendys.tistory.com/172>

| 탄생 배경 | [출처] <https://namu.wiki/w/Django>

장고는 2003~2004년에 로렌스 저널-월드([Lawrence Journal-World](#))라는 신문사의 인턴 웹 프로그래머였던 에이드리안 홀로바티(Adrian Holovaty)와 사이먼 윌리슨(Simon Willison)이 파이썬을 이용해 애플리케이션을 만들기 시작하면서 처음 개발되었다.

당시 로렌스 신문사의 웹사이트는 [PHP](#) 기반으로 쓰여졌었는데, 홀로바티와 윌리슨은 당시 PHP가 규모가 큰 웹사이트에 적합하지 않다고 생각했고 평소에 관심을 가지고 있던 파이썬으로 웹 개발을 하기로 결심했다. 하지만 규모가 큰 웹 개발에 적합한 파이썬 도구가 없다는 것을 깨닫고 새롭게 장고를 개발하게 된 것이다.

개발을 처음 할 때 사실 장고를 웹 프레임워크라고 생각한 적은 없었고, 그냥 CMS를 개발하는 것이라고 생각했다고 한다. 개발자가 개발 당시의 기억들을 풀어놓은 글을 읽고 싶다면 [여기\(영어\)](#)에 사이먼 윌리슨 본인이 직접 쓴 글이 있다. (공동 개발자인 에이드리안도 따봉을 눌렀다!) 장고라는 이름은 홀로바티가 좋아하던 기타리스트 장고 라인하르트에서 따온 것이라고 한다.

또한 장고는 사실 오직 로렌스 웹사이트를 구동하기 위해 개발된 것이었지만, 홀로바티와 윌리슨이 로렌스의 오너 기업인 더 월드 컴퍼니(The World Company)를 설립하여 오픈 소스로 2005년 7월에 [BSD 라이선스](#)로 공개하게 되었다. 윌리슨은 더 월드 컴



| Django의 인기 | 인터넷내용 [출처] <https://developer.mozilla.org/ko/docs/Learn/Server-side/Django/Introduction>

사실 서버 측 프레임워크의 인기에 대해 쉽고 확정적인 측정값은 없습니다(다만 [Hot Frameworks](#) 와 같은 사이트는 각 플랫폼에 대해 GitHub 프로젝트와 StackOverflow 질문의 숫자를 세는 방법으로 인기에 대해 접근하려고 합니다). 장고가 인기없는 플랫폼의 문제를 피할 수 있을 만큼 "충분히 인기있는지"가 더 좋은 질문입니다. 장고가 계속 발전하나요? 도움이 필요할 때 받을 수 있나요? 장고를 배우면 돈을 받고 일할 기회가 생기나요?

장고를 사용하는 상위 사이트의 숫자, 장고 코드베이스에 기여하는 사람들의 숫자, 그리고 급여가 지불되거나 지불되지 않거나에 상관없이 지원을 제공하는 사람들의 숫자에 근거해서, 맞습니다. 장고는 인기있는 프레임워크 입니다!

장고를 사용하는 상위 사이트는 다음을 포함합니다 : Disqus, Instagram, Knight Foundation, MacArthur Foundation, Mozilla, National Geographic, Open Knowledge Foundation, Pinterest, and Open Stack (출처: [Django home page](#)).



왜 Django인가?

| 사용중인 곳 | 인터넷내용 [출처] <https://namu.wiki/w/Django>

- [인스타그램](#): 세계에서 제일 큰 규모의 장고 웹사이트이다. 그래서 그런지 장고 소프트웨어 재단에 매년 3만불 이상을 기부하고 있고 2개의 플래티넘 법인 회원 중 하나이다. 다른 하나는 [IntelliJ IDEA](#)를 만든 [JetBrains](#).
- [8퍼센트](#): P2P 대출 사이트. 프론트엔드를 [IUEditor](#)로 개발하였다.
- [스마트스터디](#): 핑크퐁 북스토어 등이 Django로 개발되었다.
- [피알앤디컴퍼니](#): 내차팔기앱 [헤이딜러](#)를 운영하고 있다.
- [위시켓](#): IT 아웃소싱 중개 서비스.
- [번개장터](#): C2C 플랫폼. Django 와 Flask를 사용하고 있다.
- [숨은고수](#): O2O 고수 매칭 플랫폼. Django와 Vue.js를 사용했다.
- [피플펀드](#): P2P 금융 플랫폼, Django를 사용하고 있다.
- [마이뮤직테이스트](#): 콘서트 메이킹 플랫폼, Django를 사용하여 global service를 하고 있다.
- [Lafel](#): 애니메이션 스트리밍 서비스, Django를 사용하고 있다.
- [딜리버리히어로코리아](#): 요기요, 배달통 서비스를 하고 있는 푸드테크 기업. Django를 사용하고 있다.
- [SendBird](#): 채팅 API를 제작하는 B2B 기업. Django를 사용하고 있다.



| MVC 패턴 기반 MVT |

장고는 MVC(Model-View-Controller)를 기반으로 한 프레임워크입니다. 하지만 장고에서는 View를 Template, Controller를 View라고 함

| 객체 관계 매핑 |

장고의 객체 관계 매핑(ORM, Object-Relational Mapping)은 데이터베이스 시스템과 데이터 모델 클래스를 연결시키는 다리 역할을 함

| 자동으로 구성되는 관리자 화면 |

장고는 웹 서버의 콘텐츠, 즉 데이터베이스에 대한 관리 기능을 위하여 프로젝트를 시작하는 시점에 기본 기능으로 관리자 화면을 제공합니다

| 우아한 URL 설계 |

웹 프로그래밍에서 URL 디자인은 필수인데, 장고에서는 유연하면서도 강력한 기능을 제공. 장고에서는 우아한(Elegant) URL 방식을 채택하여 URL을 직관적이고 쉽게 표현

| 자체 템플릿 시스템 |

장고는 내부적으로 확장이 가능하고 디자인이 쉬운 강력한 템플릿 시스템을 갖고 있음. 이를 통해 화면 디자인과 로직에 대한 코딩을 분리하여 독립적으로 개발 진행



| 캐시 시스템 |

컨텐츠를 동적으로 작성하는 것은 정적 컨텐츠를 제공하는 것 보다 많은 연산을 필요로 하기 때문에 느립니다. 장고는 유연한 캐싱을 제공하여 렌더링된 페이지 전체 또는 일부를 저장하여 필요할 때를 제외하고 다시 렌더링하지 않도록 할 수 있도록 하는 기능

| 다국어 지원 |

장고는 동일한 소스코드를 다른 나라에서도 사용할 수 있도록 텍스트의 번역, 날짜/시간/숫자의 포맷, 타임존의 지정 등과 같은 다국어 환경을 제공

| 풍부한 개발 환경 |

장고는 개발에 도움이 될 수 있는 여러 가지 기능을 제공

| 소스 변경사항 자동 반영 |

장고에서는 *.py 파일의 변경 여부를 감시하고 있다가 변경이 되면 실행 파일에 변경 내역을 바로 반영

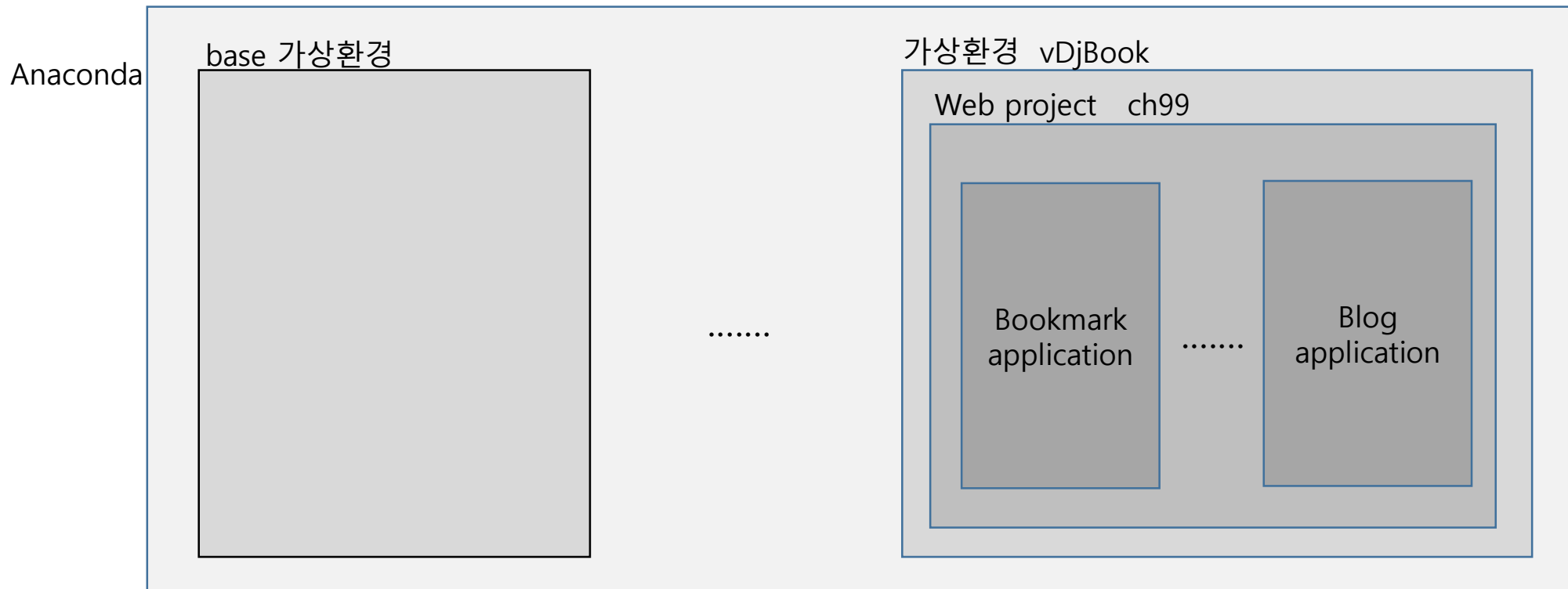
개발환경 및 웹 구축 순서



1. 아나콘다 설치 , pycharm 설치
2. 가상환경 만들기
3. Django 및 필요한 프로그램 설치
4. 프로젝트 만들기(웹서버) , 프로젝트 설정 파일 변경
5. admin 관련 기본 테이블 생성 및 슈퍼유저 생성

1~5는 한 번만 수행

6. 애플리케이션 만들기 : 예) 가천대 글로벌 캠퍼스 웹, 가천대 메디컬 캠퍼스 웹 등등
7. 모델(테이블 또는 SQL을 이용한 가상 테이블) → URLconf(URL 설계) → View(커서개념 또는 테이블과 HTML 중간) → Template 파일(HTML)



6~7은
새로운
응용을
만들때
마다
수행



1. Anaconda 설치

<https://www.anaconda.com/products/individual>



Individual Edition

Your data science toolkit

With over 20 million users worldwide, the open-source Individual Edition (Distribution) is the easiest way to perform Python/R data science and machine learning on a single machine. Developed for solo practitioners, it is the toolkit that equips you to work with thousands of open-source packages and libraries.

Download

클릭



1. Anaconda 설치

해당 OS와 해당 bit를 선택하여 설치

Anaconda Installers

Windows

Python 3.8

64-Bit Graphical Installer (466 MB)

32-Bit Graphical Installer (397 MB)

MacOS

Python 3.8

64-Bit Graphical Installer (462 MB)

64-Bit Command Line Installer (454 MB)

Linux

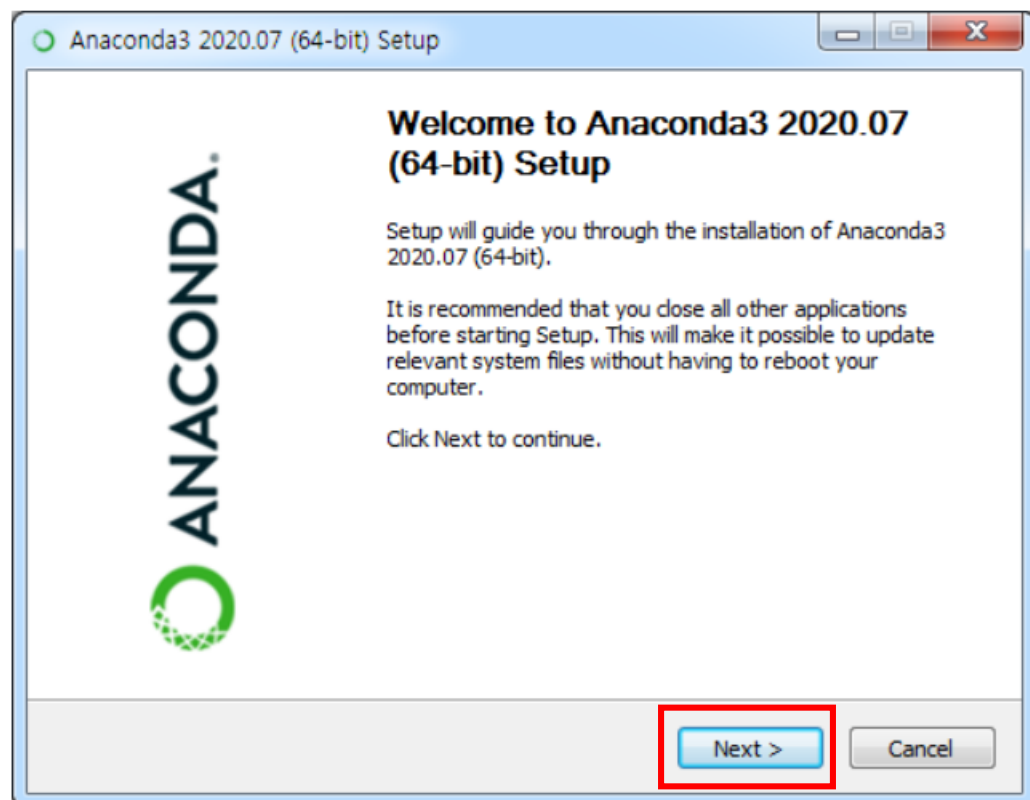
Python 3.8

64-Bit (x86) Installer (550 MB)

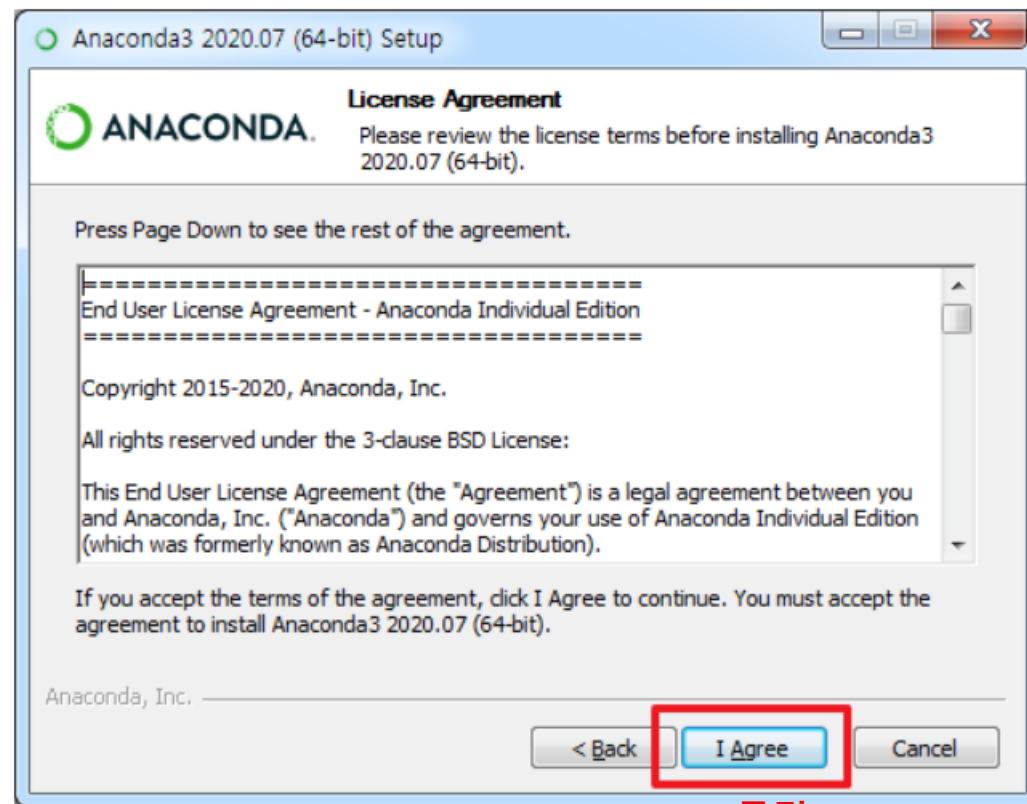
64-Bit (Power8 and Power9) Installer (290 MB)



1. Anaconda 설치



클릭

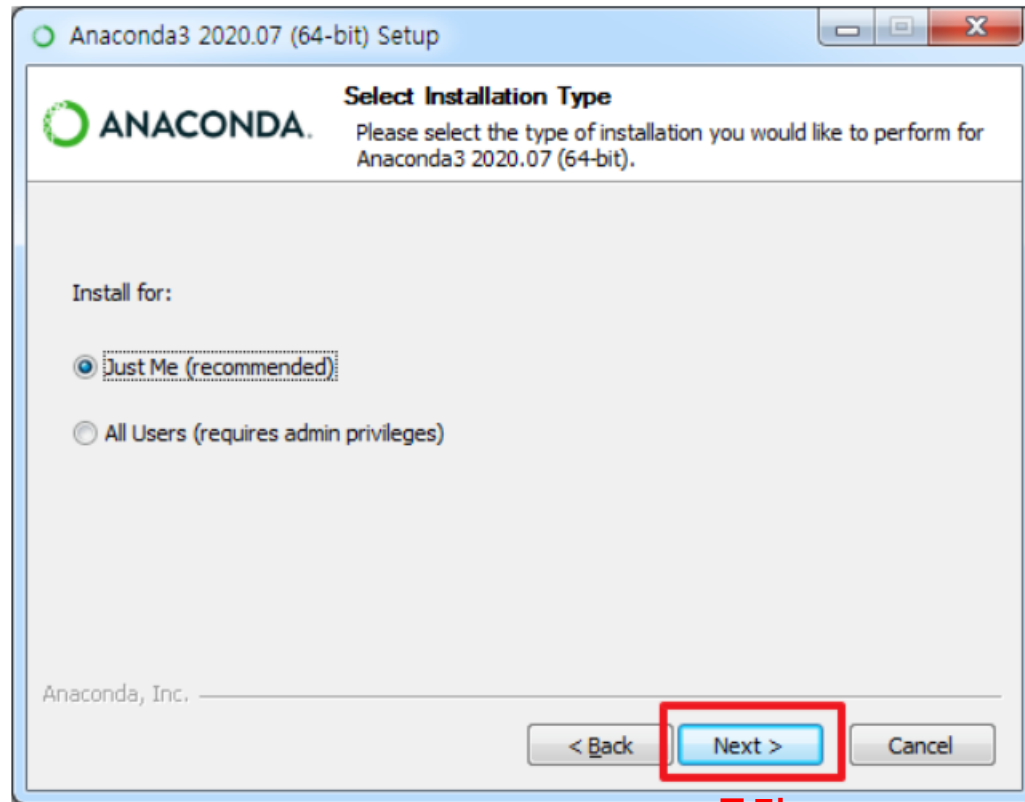


클릭

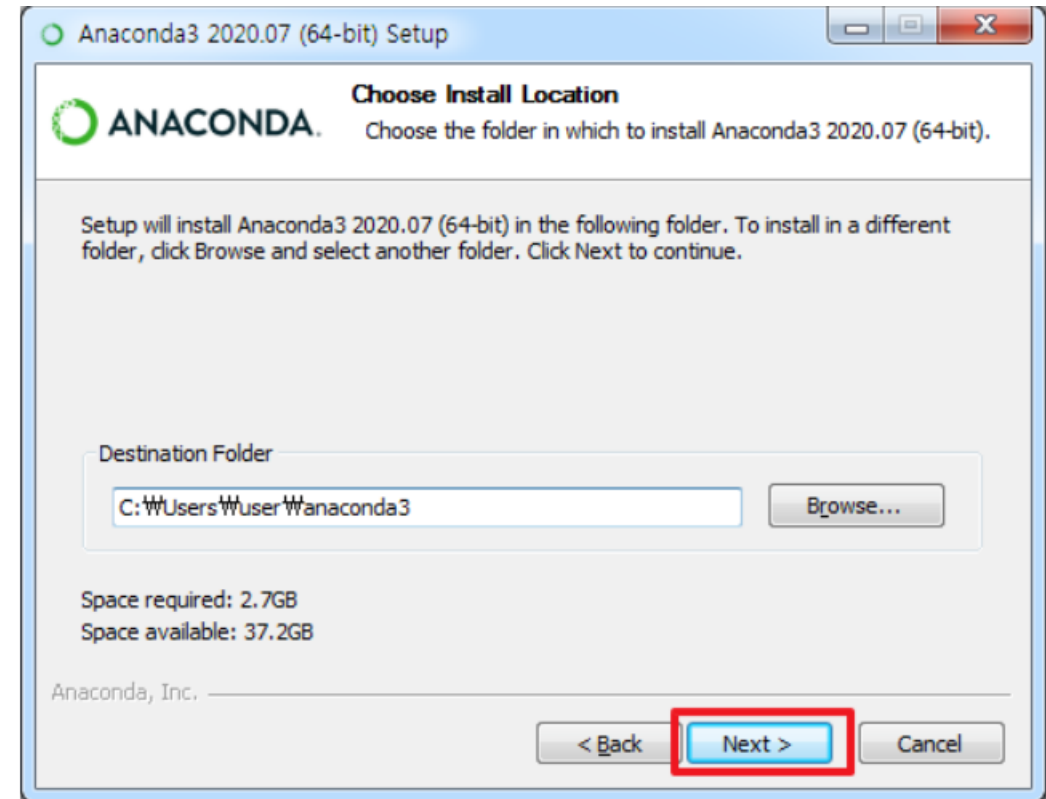


1. Anaconda 설치

해당 OS와 해당 bit를 선택하여 설치



클릭

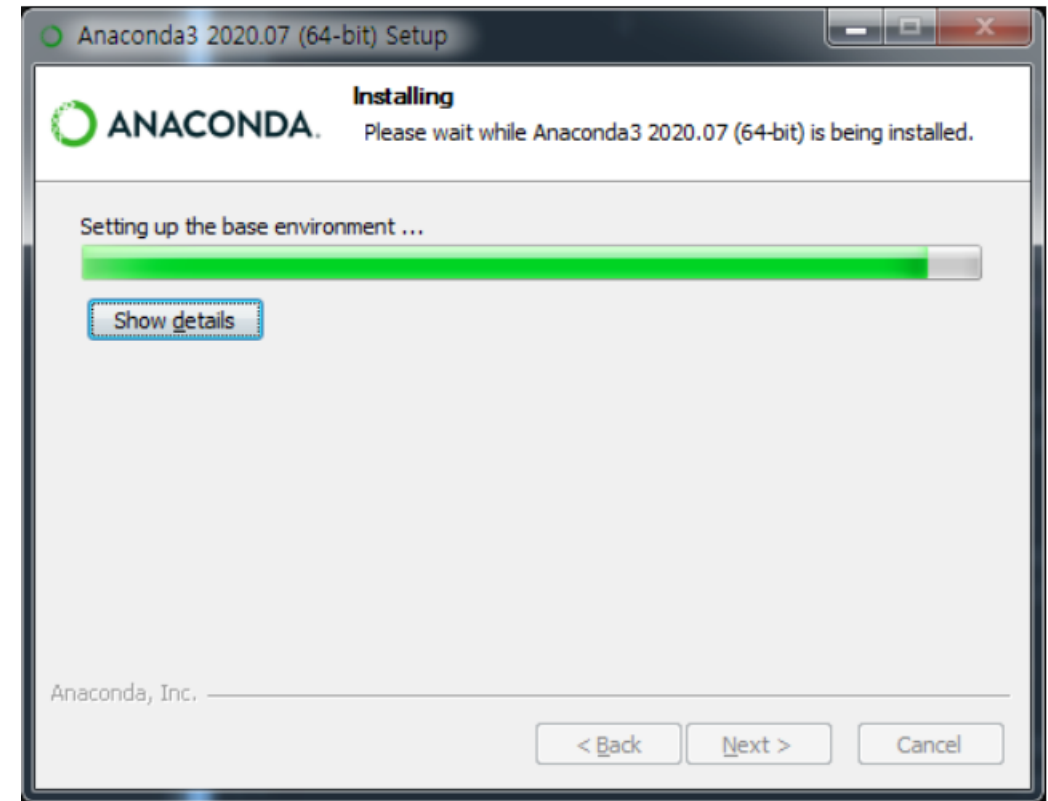
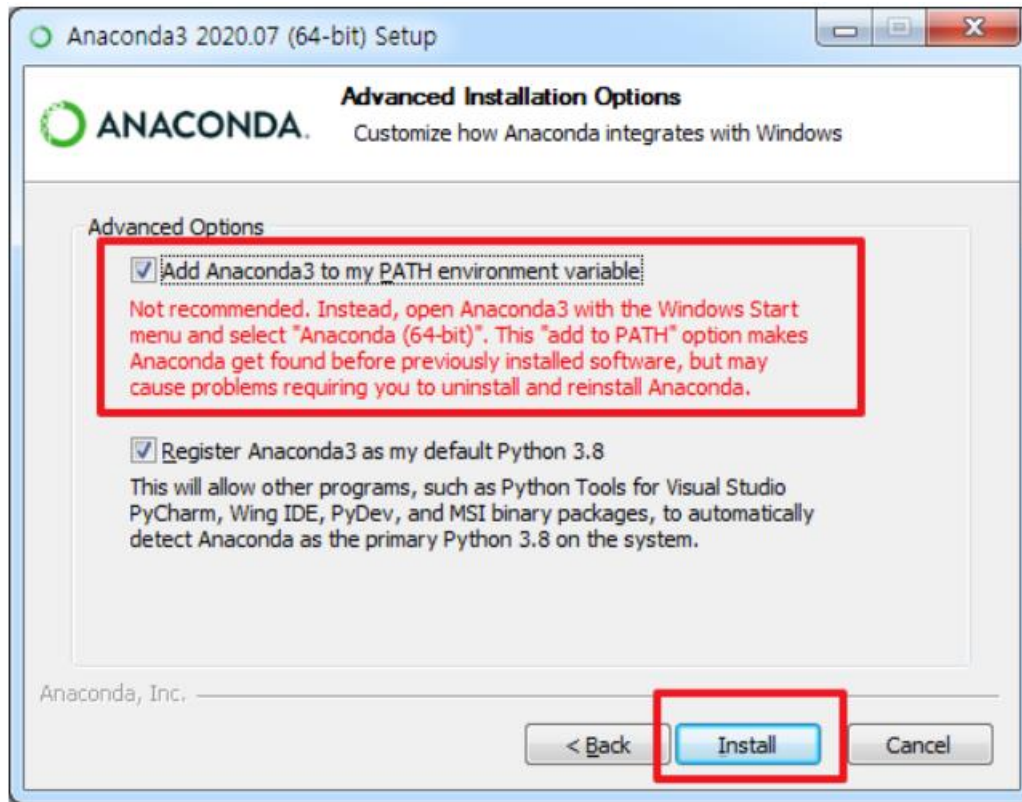


설치 폴더를 바꾸고 싶으면 Browse 버튼
클릭해서 폴더 수정 후 Next



1. Anaconda 설치

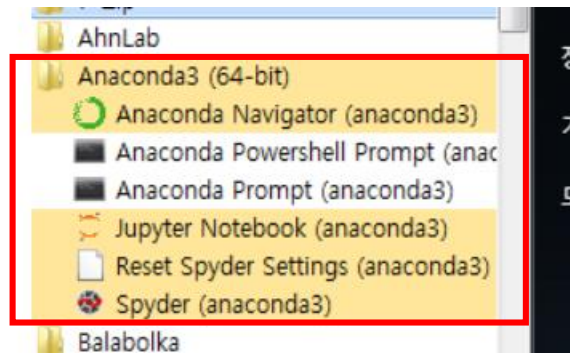
해당 OS와 해당 bit를 선택하여 설치



위의 Add ~~ 부분을 체크해야 PATH가 등록되어 작업이 편리함



1. Anaconda 설치



설치됨




1. PyCharm 설치

<https://www.jetbrains.com/ko-kr/pycharm/download/#section=windows>

PyCharm

Coming in 2020.1 새로운 기능 기능 구매 다운로드



버전: 2019.3.3
빌드: 193.6494.30
2020년 2월 7일

[시스템 요구 사항](#)
[설치 안내](#)
[기타 버전](#)

다운로드 PyCharm

[Windows](#) [Mac](#) [Linux](#) **선택**

Professional

과학 및 웹 Python 개발용. HTML, JS, SQL 지원.

[다운로드](#)


무료 평가판

Community

순수 Python 개발용

[다운로드](#) **클릭**

무료, 오픈 소스



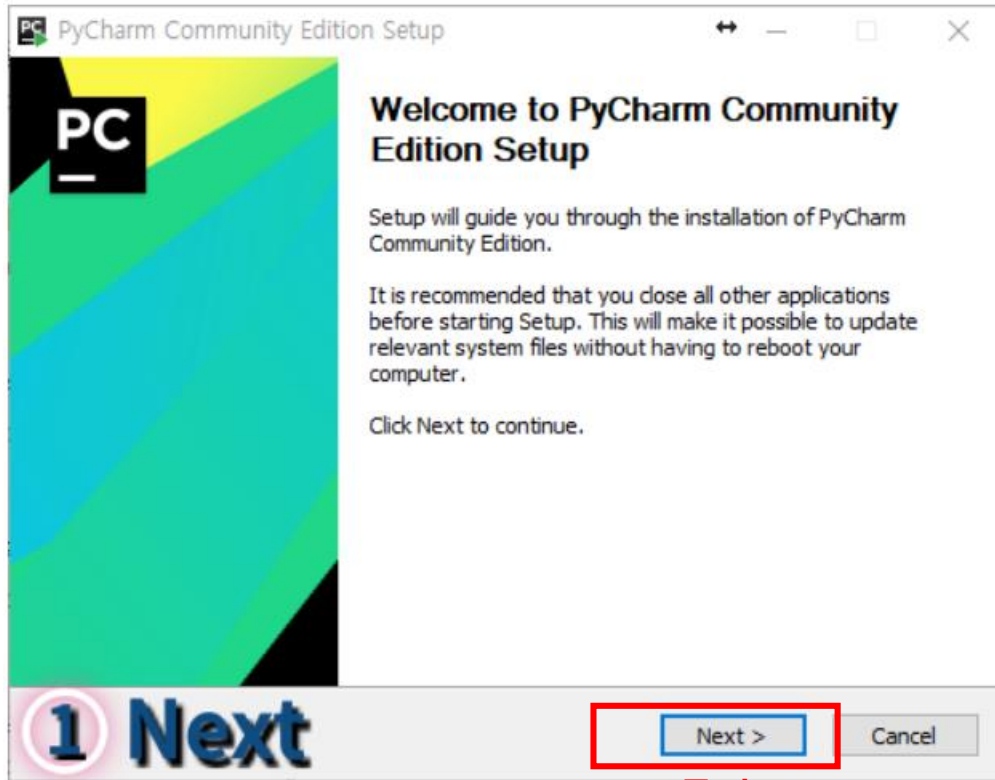
Toolbox App을 설치하여 PyCharm 및 향후 업데이트를 간편하게 다운로드하세요

[출처]<https://mingtrace.tistory.com/500>

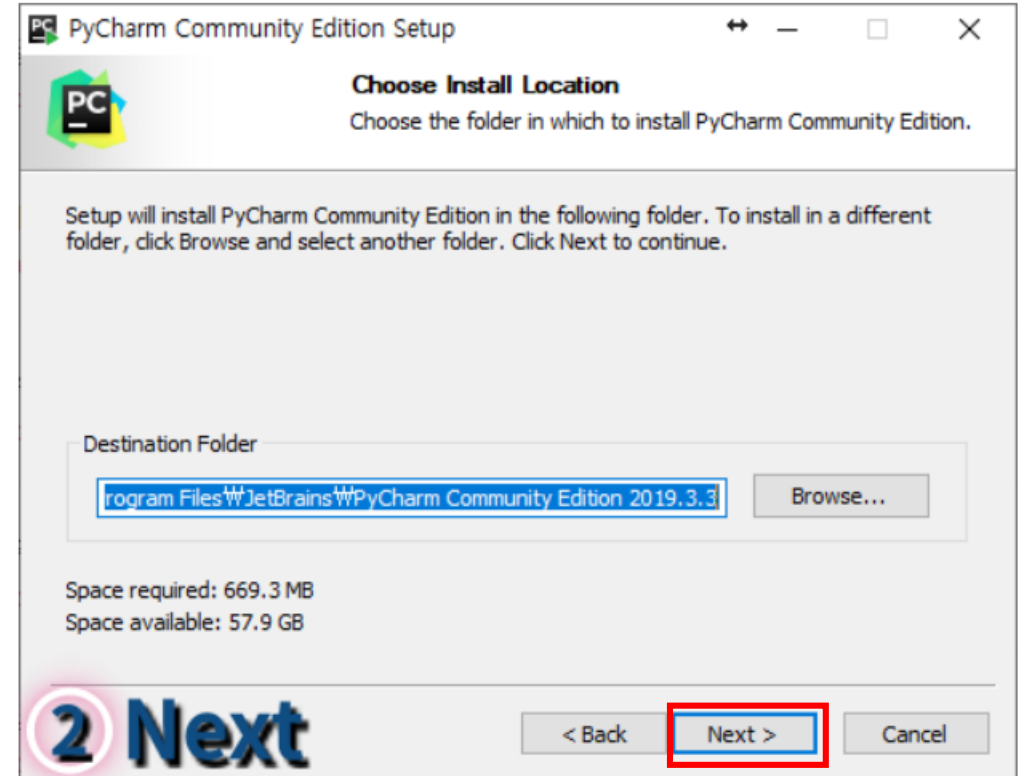


1. PyCharm 설치

파이참 설치 파일을 다운로드하여 실행



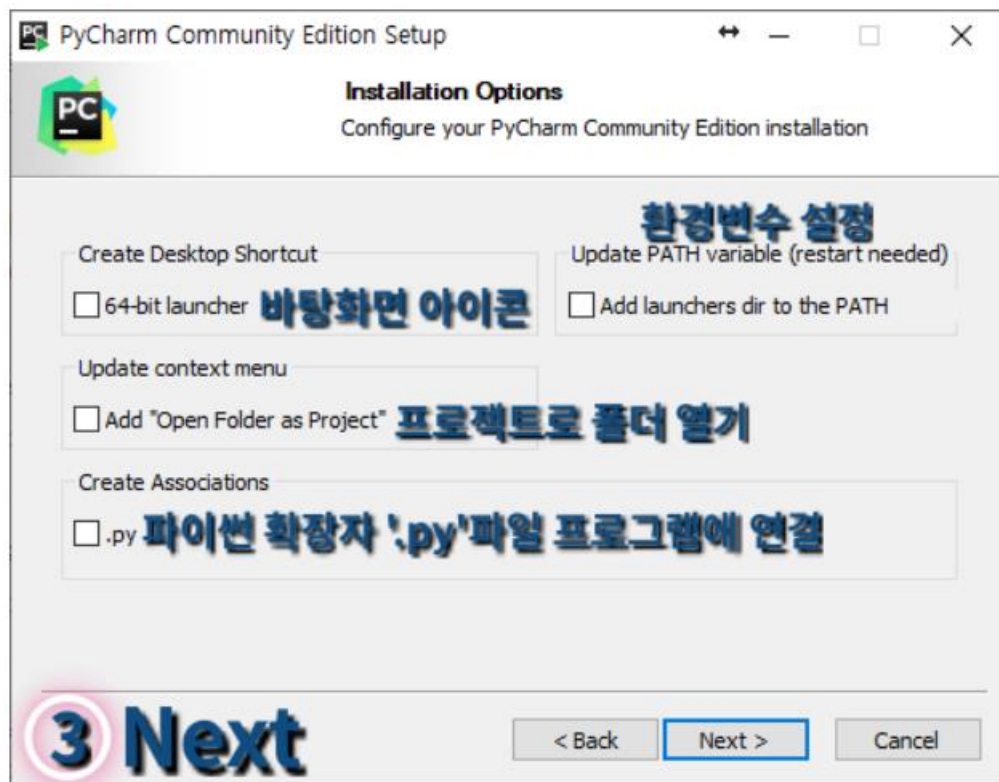
클릭



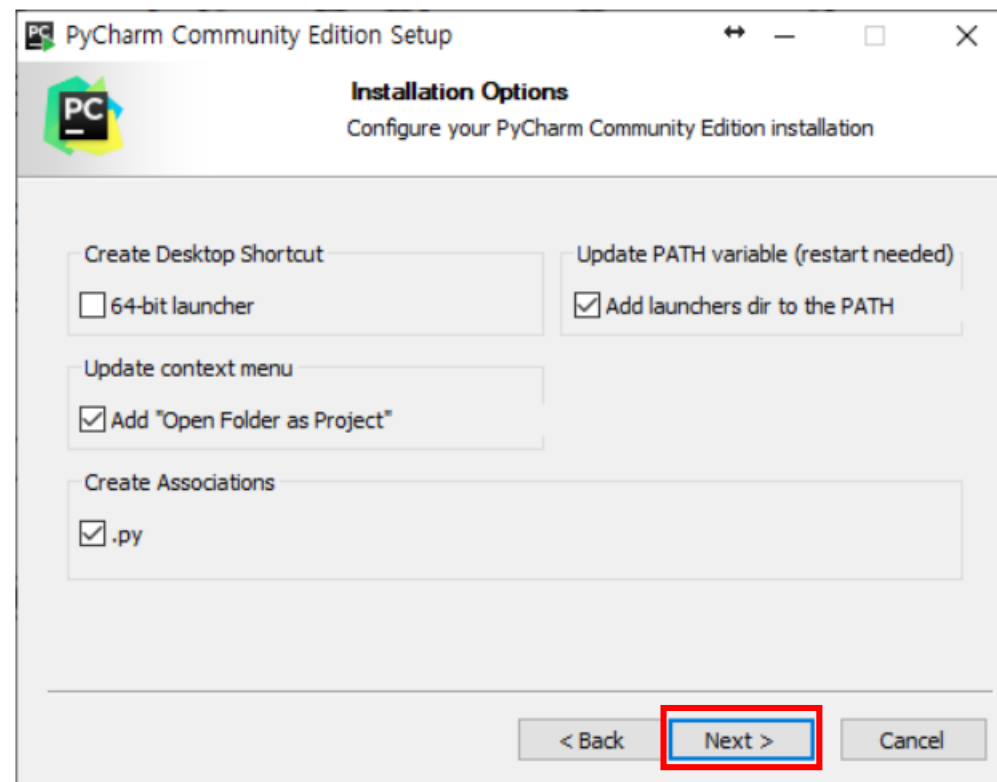
설치폴더 수정 후(원하지 않으면 그냥 Next 클릭) Next 클릭



1. PyCharm 설치



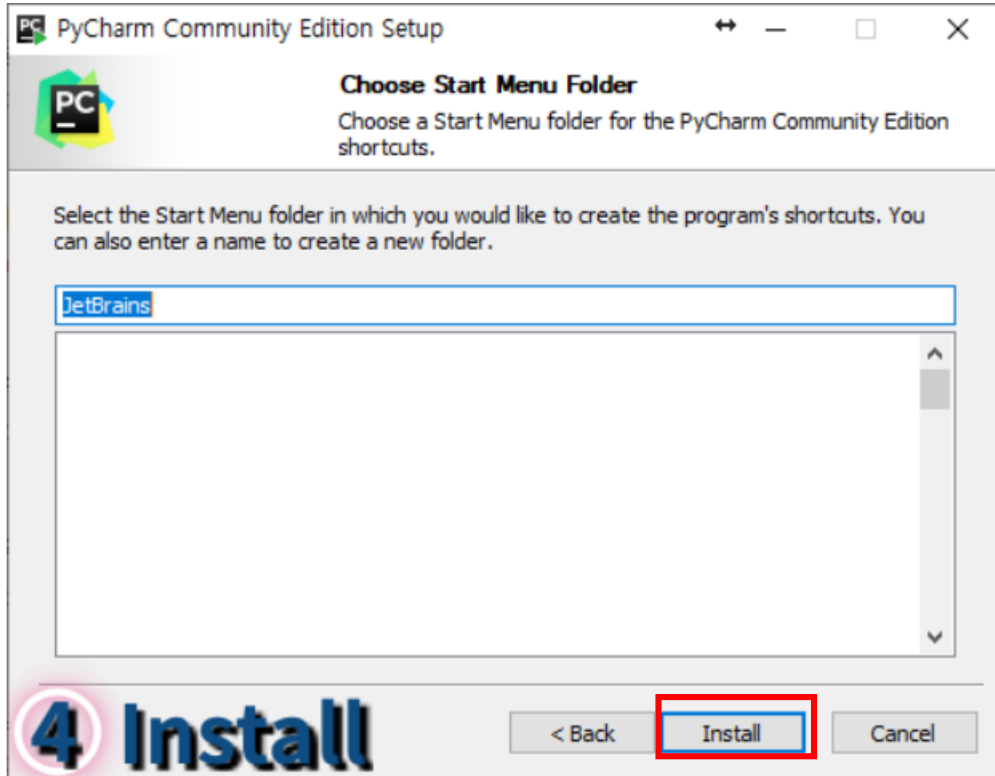
옵션들 선택
모두 선택



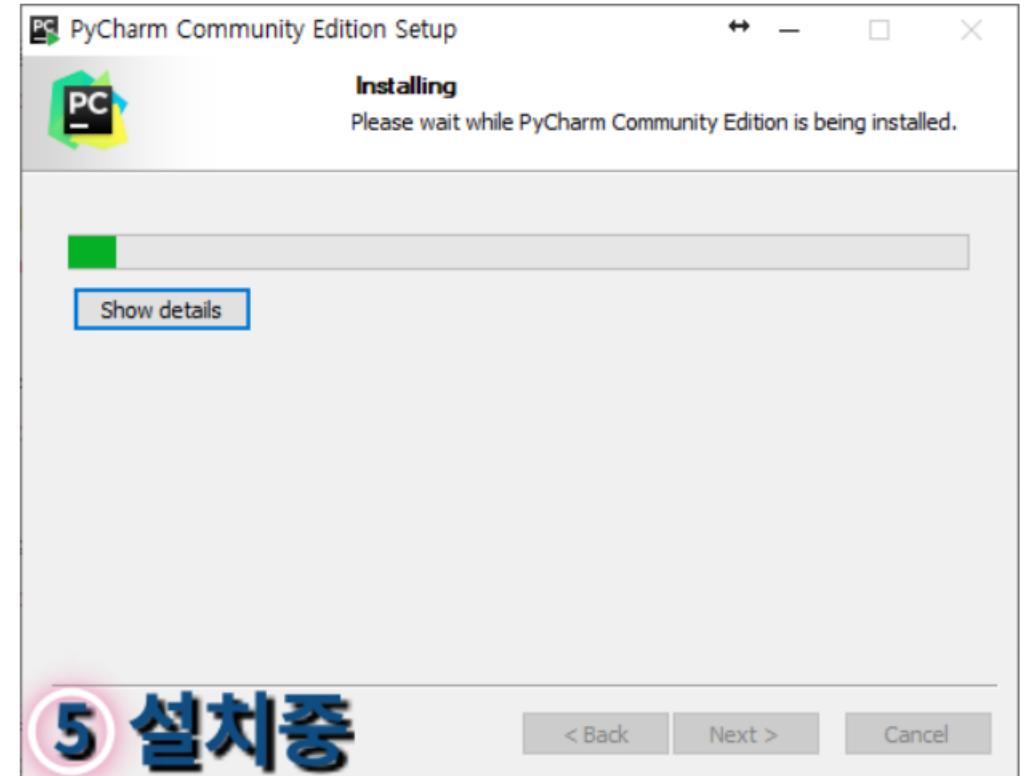
클릭



1. PyCharm 설치

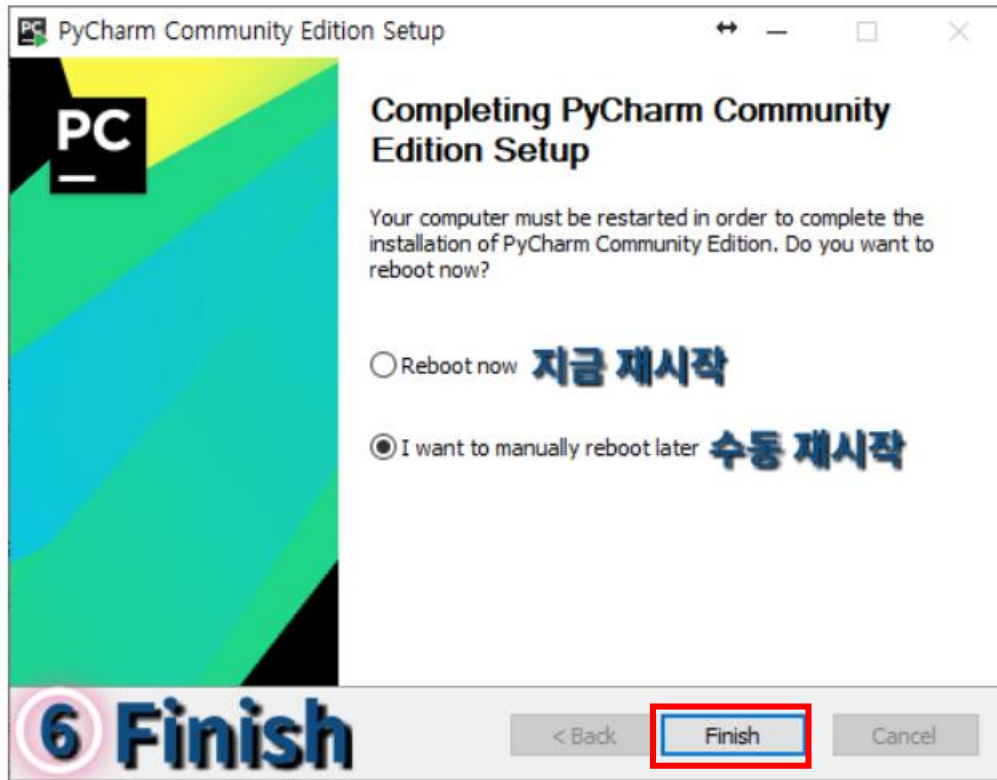


변경없이 Install 클릭

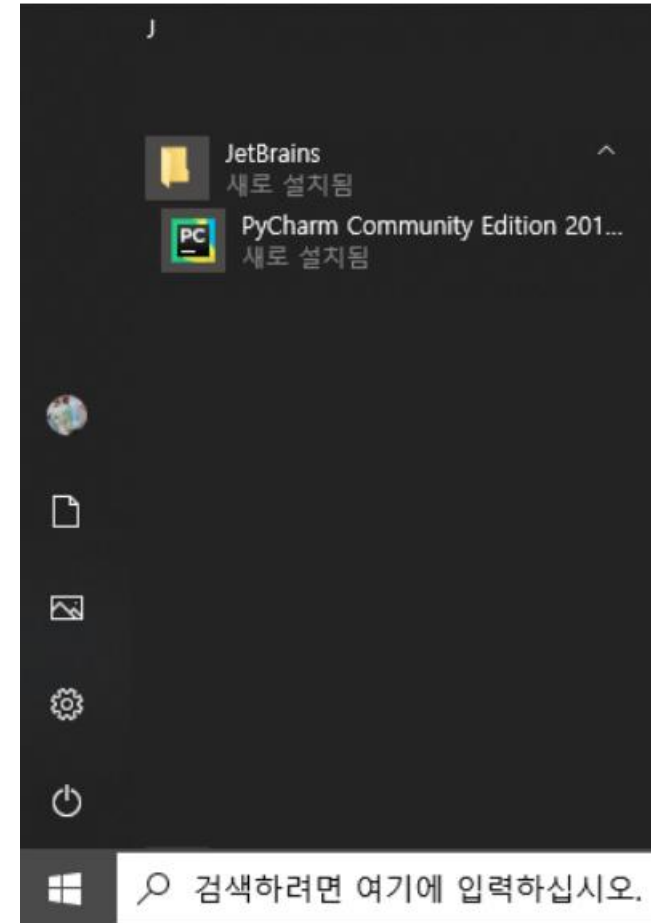




1. PyCharm 설치

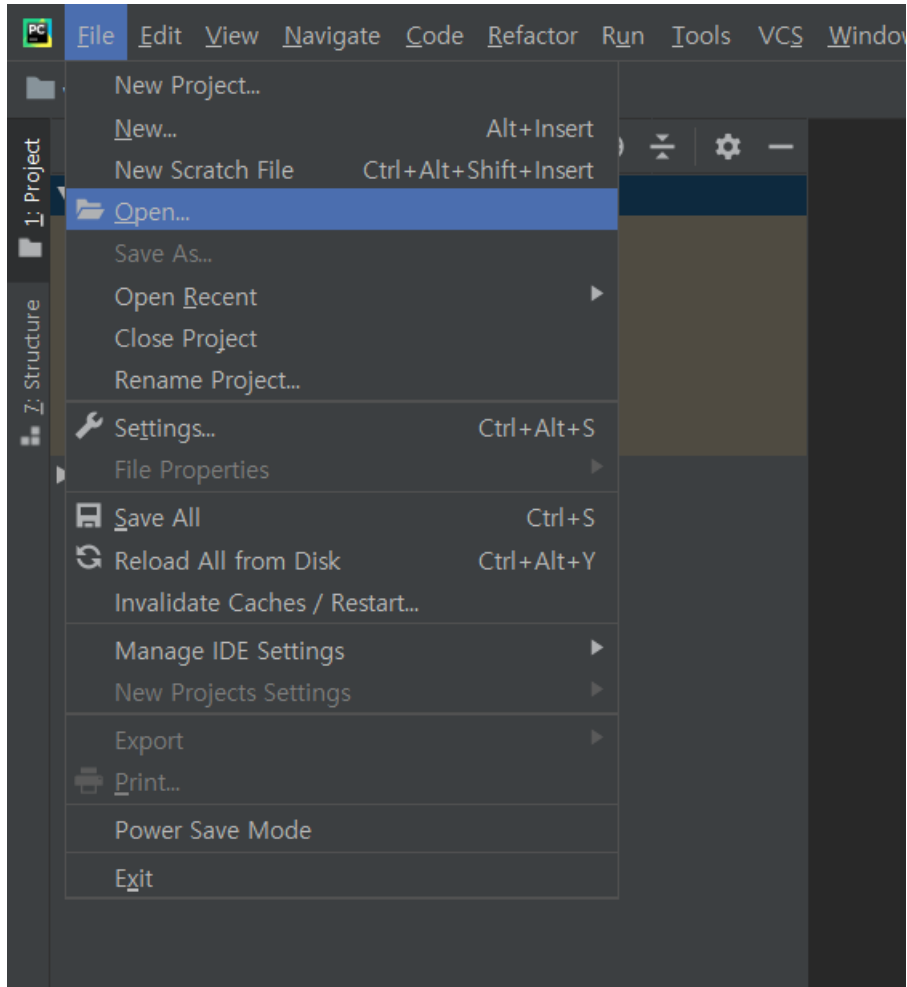


클릭





1. PyCharm 설치



후에 가상환경을 만들어서 해당 폴더를 open 하면 편집가능



2. 가상환경 만들기 - Anaconda 이용

1

2

Environments

Applications on base (root)

Application	Version	Action
console_shortcut	0.1.1	Launch
JupyterLab	1.2.6	Launch
Notebook	6.0.3	Launch
powershell_shortcut	0.0.1	Launch
Qt Console	4.6.0	Launch
Spyder	4.0.1	Launch
VS Code	1.47.3	Launch
Glueviz	0.15.2	Install
Orange 3	3.26.0	Install
RStudio	1.1.456	Install



2. 가상환경 만들기

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Home

Environments

Learning

Community

Documentation

Developer Blog



Create



Clone



Import



Remove

Search Environments



base (root)

django



vDjBook

Installed



Channels

Update index...

Search Packages



Name



T

Description



certifi



Python package for providing mozilla's ca bundle.



pip



Pypa recommended tool for installing python packages



python



General purpose programming language



setuptools



Download, build, install, upgrade, and uninstall python packages



vc



A meta-package to impose mutual exclusivity among software built with dif



vs2015_runtime



Msvc runtimes associated with cl.exe version 19.15.26726 (vs 2017 update 8



wheel



A built-package format for python.



wincertstore



Python module to extract ca and crt certs from windows' cert store (ctypes



8 packages available

3

2. 가상환경 만들기

base (root) ▶

Temp

django

vDjBook

Search Environments

Installed Channels Update index... Search Packages

Name T Description

✓ _anaconda_depends

✓ _ipyw_jlab_nb_ex... A configuration metapackage for enabling anaconda-bundled jupyter extensions

✓ _tflow_select

✓ absl-py Abseil python common libraries, see <https://github.com/abseil/abseil-py>.

✓ alabaster

✓ anaconda

✓ anaconda-client

✓ anaconda-project

✓ argh

✓ asn1crypto

✓ astor Read, rewrite, and write python code

✓ astroid A abstract syntax tree for python with inference support.

✓ astropy Community-developed python library for astronomy

✓ atomicwrites Atomic file writes.

✓ attrs Attrs is the python package that will bring back the joy of writing classes by relieving you from the c

✓ autopep8 A tool that automatically formats python code to conform to the pep 8 style guide

✓ babel Utilities to internationalize and localize python applications

✓ backcall Specifications for callback functions passed in to an api

✓ backports

316 packages available

Create new environment

Name: vDjBook

Location: C:\Anaconda3\envs\vDjBook

Packages: ☒ Python ☐ R

3.7

✓ 3.7

3.6

3.5

2.7

Cancel Create

4. 가상환경이름 입력

5. Python 버전 선택

가상환경생성 시간이 좀 걸림



2. 가상환경 만들기

6. Home 메뉴 클릭

The screenshot shows the Anaconda Navigator interface. The left sidebar contains the 'Home' menu item, which is highlighted with a red box. The main area displays a list of installed packages for the 'vDjBook-test' environment. The table below shows the installed packages and their versions.

Name	Description	Version
ca-certificates	Certificates for use with other packages.	2020.7.22
certifi	Python package for providing mozilla's ca bundle.	2020.6.20
openssl	Openssl is an open-source implementation of the ssl and tls protocols	1.1.1g
pip	Pypa recommended tool for installing python packages	20.2.2
python	General purpose programming language	3.7.7
setuptools	Download, build, install, upgrade, and uninstall python packages	49.6.0
sqlite	Implements a self-contained, zero-configuration, sql database engine.	3.33.0
vc	A meta-package to impose mutual exclusivity among software built with different vs versions	14.1
vs2015_runtime	Msvc runtimes associated with cl.exe version 19.15.26726 (vs 2017 update 8)	14.16.2...
wheel	A built-package format for python.	0.35.1
wincertstore	Python module to extract ca and crt certs from windows' cert store (ctypes based).	0.2
zlib	Massively spiffy yet delicately unobtrusive compression library	1.2.11

가상환경생성 후 화면

7 가상환경이름 선택

Applications

vDjBook-test

Channels

Refresh

VS Code

1.48.1

Streamlined code editor with support for development operations like debugging, task running and version control.

Launch

console_shortcut

0.1.1

Console shortcut creator for Windows (using menuinst)

8. install

Install

Glueviz

0.15.2

Multidimensional data visualization across files. Explore relationships within and among related datasets.

Install

JupyterLab

2.2.6

An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.

Install

Notebook

6.1.1

Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.

Install

Orange 3

3.26.0

Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.

Install

powershell_shortcut

0.0.1

Install

Qt Console

4.7.6

PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.

Install

RStudio

1.1.456

A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.

Install

Spyder

4.1.4

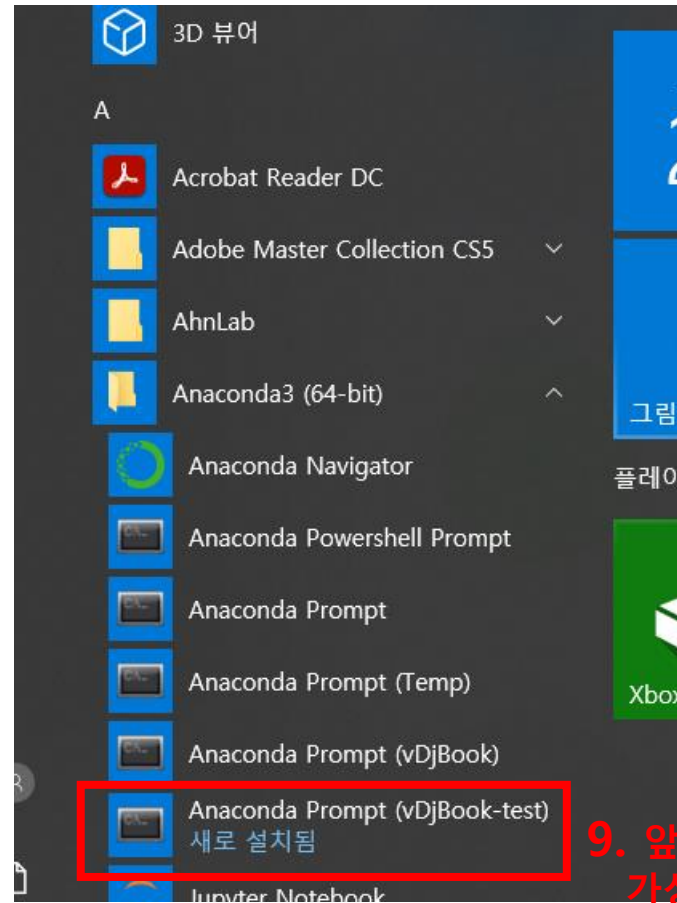
Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features

Install

python version에 따라
cmd.exe prompt 로 보인다



2. 가상환경 만들기



9. 앞페이지 8번을 하면
가상환경 prompt 메뉴 생성



3. Django 및 필요한 프로그램 설치

```
(vDjBook)C:\W\Anaconda3\envs\WvDjBook>pip install Django
```

Django 설치

```
(vDjBook)C:\W\Anaconda3\envs\WvDjBook>pip list
```

설치된 목록 확인

```
(vDjBook)C:\W\Anaconda3\envs\WvDjBook>pip install django-taggit
```

교재에서 필요한 lib 설치

```
(vDjBook) (base) C:\Users\W\BH\venv\WvDjBook>pip3 install django-taggit
Collecting django-taggit
  Downloading https://files.pythonhosted.org/packages/56/3e/dde2d7151bc0c4ac65d225e611a85e54a897c551507e8eca2c06a083f3f4/django_taggit-1.3.0-py3-none-any.whl (45kB)
    100% |#####| 51kB 184kB/s
Requirement already satisfied: Django>=1.11 in c:\users\W\BH\venv\WvDjBook\lib\site-packages (from django-taggit) (3.0.8)
Requirement already satisfied: pytz in c:\users\W\BH\venv\WvDjBook\lib\site-packages (from Django>=1.11->django-taggit) (2020.1)
Requirement already satisfied: asgiref<=3.2 in c:\users\W\BH\venv\WvDjBook\lib\site-packages (from Django>=1.11->django-taggit) (3.2.10)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\W\BH\venv\WvDjBook\lib\site-packages (from Django>=1.11->django-taggit) (0.3.1)
Installing collected packages: django-taggit
Successfully installed django-taggit-1.3.0
You are using pip version 18.1, however version 20.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

설치중 화면



3. Django 및 필요한 프로그램 설치

```
(vDjBook)C:\Anaconda3\envs\vDjBook>pip install django-taggit-templatetags2
```

교재에서 필요한 lib 설치

```
(vDjBook) (base) C:\Users\WBH\envs\vDjBook>pip3 install django-taggit-templatetags2
Collecting django-taggit-templatetags2
  Downloading https://files.pythonhosted.org/packages/97/d9/8e620d79fa9c0c2b50d1c3f7e677c8afe9236c97a7809565cc906d1c33b7/django_taggit_templatetags2-1.6.1-py2.py3-none-any.whl (67kB)
    100% |#####| 71kB 228kB/s
Requirement already satisfied: django-taggit>=0.12 in c:\users\wbh\envs\vDjBook\lib\site-packages (from django-taggit-templatetags2) (1.3.0)
Requirement already satisfied: django>=1.5 in c:\users\wbh\envs\vDjBook\lib\site-packages (from django-taggit-templatetags2) (3.0.8)
Collecting django-classy-tags>=0.5.1 (from django-taggit-templatetags2)
  Downloading https://files.pythonhosted.org/packages/19/22/39385390e854356a1267d246d9ac6ed45fd0b90c7ec422c2ce7aea647d25/django-classy-tags-1.0.0.tar.gz
Requirement already satisfied: pytz in c:\users\wbh\envs\vDjBook\lib\site-packages (from django>=1.5->django-taggit-templatetags2) (2020.1)
Requirement already satisfied: asgiref~=3.2 in c:\users\wbh\envs\vDjBook\lib\site-packages (from django>=1.5->django-taggit-templatetags2) (3.2.10)
Requirement already satisfied: sqlparse>=0.2.2 in c:\users\wbh\envs\vDjBook\lib\site-packages (from django>=1.5->django-taggit-templatetags2) (0.3.1)
Collecting six (from django-classy-tags>=0.5.1->django-taggit-templatetags2)
  Downloading https://files.pythonhosted.org/packages/ee/ff/48bde5c0f013094d729fe4b0316ba2a24774b3ff1c52d924a8a4cb04078a/six-1.15.0-py2.py3-none-any.whl
Installing collected packages: six, django-classy-tags, django-taggit-templatetags2
  Running setup.py install for django-classy-tags ... done
Successfully installed django-classy-tags-1.0.0 django-taggit-templatetags2-1.6.1 six-1.15.0
You are using pip version 18.1, however version 20.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.

(vDjBook) (base) C:\Users\WBH\envs\vDjBook>
```

설치중 화면



3. Django 및 필요한 프로그램 설치

```
(vDjBook)C:\Anaconda3\envs\vDjBook>pip install django-widget-tweaks
```

교재에서 필요한 lib 설치

```
(vDjBook) (base) C:\Users\WBH\envs\vDjBook>pip3 install django-widget-tweaks
Collecting django-widget-tweaks
  Downloading https://files.pythonhosted.org/packages/1d/36/d7eb0ccfe4d55d495943085be7bcb1b262ec3c5052afb18e07dcfb88ecb8/django_widget_tweaks-1.4.8-py2.py3-none-any.whl
Installing collected packages: django-widget-tweaks
Successfully installed django-widget-tweaks-1.4.8
You are using pip version 18.1, however version 20.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

설치중 화면

```
(vDjBook)C:\Anaconda3\envs\vDjBook>pip install Pillow
```

교재에서 필요한 lib 설치

```
(vDjBook) (base) C:\Users\WBH\envs\vDjBook>pip3 install Pillow
Collecting Pillow
  Downloading https://files.pythonhosted.org/packages/f0/03/bec597cb6cc3cab8e3d684c6167c3beafae753f2ed6753af6f072a0714c1/Pillow-7.2.0-cp36-cp36m-win_amd64.whl (2.0MB)
    100% |#####| 2.0MB 6.4MB/s
Installing collected packages: Pillow
Successfully installed Pillow-7.2.0
You are using pip version 18.1, however version 20.2 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```

설치중 화면



3. Django 및 필요한 프로그램 설치

```
(vDjBook)C:\Anaconda3\envs\vDjBook>pip install -U pip setuptools wheel
```

설치한 library upgrade

```
(vDjBook) (base) C:\Users\WBH\envs\vDjBook>pip3 install -U pip setuptools wheel
Collecting pip
  Downloading https://files.pythonhosted.org/packages/36/74/38c2410d688ac7b48afa07d413674afc1f903c1c1f854de51dc8eb2367a5/pip-20.2-py2.py3-none-any.whl (1.5MB)
    100% |#####| 1.5MB 7.3MB/s
Collecting setuptools
  Downloading https://files.pythonhosted.org/packages/ed/16/e9f5c5b86696da09298ea10c32d68ad8ea21f888e45b11aa9e615adda6c9/setuptools-49.2.1-py3-none-any.whl (789kB)
    100% |#####| 798kB 9.3MB/s
Collecting wheel
  Downloading https://files.pythonhosted.org/packages/8c/23/848298cccf8e40f5bbb59009b32848a4c38f4e7f3364297ab3c3e2e2cd14/wheel-0.34.2-py2.py3-none-any.whl
Installing collected packages: pip, setuptools, wheel
  Found existing installation: pip 18.1
    Uninstalling pip-18.1:
      Successfully uninstalled pip-18.1
```



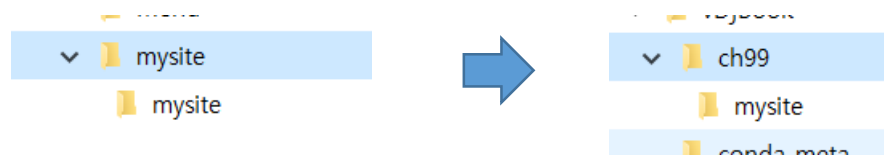
4. 프로젝트 만들기(웹서버) , 프로젝트 설정 파일 변경

```
(vDjBook)C:\Anaconda3\envs\vDjBook>django-admin startproject mysite
```

실행하면 mysite project 폴더와 그 하위에

mysite 폴더 생성

상위 mysite 폴더를 ch99로 수정 또는 다른 이름도 가능



프로젝트 설정 파일 settings.py 주요 사항

- 데이터베이스 설정
- 애플리케이션 등록
- 템플릿 항목 설정
- 정적 파일 항목 설정
- 타임존 지정



4. 프로젝트 만들기(웹서버) , 프로젝트 설정 파일 변경

ch99\mysite\settings.py 수정하기 settings.py 파일

import os 가 없다면
추가할것

```
"""
Django settings for mysite project.

Generated by 'django-admin startproject' using Django 3.1.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/
"""

import os
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve(strict=True).parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'f#e(!213^vbsmjgmh!sddb0l#(n5jf)=8rmjp8(1kl*08=73a('

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['localhost', '127.0.0.1']
```



4. 프로젝트 만들기(웹서버) , 프로젝트 설정 파일 변경

ch99\mysite\settings.py 수정하기

```
"""
Django settings for mysite project.

Generated by 'django-admin startproject' using Django 3.1.

For more information on this file, see
https://docs.djangoproject.com/en/3.1/topics/settings/

For the full list of settings and their values, see
https://docs.djangoproject.com/en/3.1/ref/settings/
"""

import os
from pathlib import Path

# Build paths inside the project like this: BASE_DIR / 'subdir'.
BASE_DIR = Path(__file__).resolve(strict=True).parent.parent

# Quick-start development settings - unsuitable for production
# See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/

# SECURITY WARNING: keep the secret key used in production secret!
SECRET_KEY = 'f#e(!213^vbsmjgmh!sddb0l#(n5jf)=8rmjp8(1kl*08=73a('

# SECURITY WARNING: don't run with debug turned on in production!
DEBUG = True

ALLOWED_HOSTS = ['localhost', '127.0.0.1']
```

※ localhost, 127.0.0.1

자기자신의 컴퓨터에 할당된 웹서버 ip 및 호스트명
내 컴퓨터에서 웹서버를 설정해서 테스트하기 위해
사용할 수 있는 IP 및 호스트명

다음과 같이 수정할 것

```
ALLOWED_HOSTS = ['localhost', '127.0.0.1']
```



ch99\mysite\settings.py 수정하기

```
TEMPLATES = [  
    {  
        'BACKEND': 'django.template.backends.django.DjangoTemplates',  
        'DIRS': [os.path.join(BASE_DIR, 'templates')], #수정 디폴드 디렉토리 설정  
        'APP_DIRS': True,  
        'OPTIONS': {  
            'context_processors': [  
                'django.template.context_processors.debug',  
                'django.template.context_processors.request',  
                'django.contrib.auth.context_processors.auth',  
                'django.contrib.messages.context_processors.messages',  
            ],  
        },  
    ],  
]
```

```
DATABASES = {  
    'default': {  
        'ENGINE': 'django.db.backends.sqlite3',  
        'NAME': os.path.join(BASE_DIR, 'db.sqlite3'),  
    }  
}
```

디폴트내용 다른 DB엔진을 사용하고 싶으면 이 부분을 수정



5. admin 관련 기본 테이블 생성 및 슈퍼유저 생성

```
(vDjBook)C:\WAnaconda3\envs\WvDjBook\ch99>python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, 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 sessions.0001_initial... OK
```

Admin 관련 DB를 생성

```
(vDjBook)C:\WAnaconda3\envs\WvDjBook\ch99>python manage.py createsuperuser
사용자 이름 (leave blank to use 'wbh'): bhwang99
이메일 주소: bhwang99@hanmail.net
Password:
Password (again):
Superuser created successfully.
(vDjBook) (base) C:\Users\W\B\H\venv\WvDjBook\ch99>.
```

Admin 접근 권한 설정

bhwang99 bhwang99@hanmail.net bh123456



5. admin 화면의 확인

웹서버 실행

```
(vDjBook)C:\WAnaconda3\envs\WvDjBook\ch99>python manage.py runserver 0.0.0.0:8000
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
August 04, 2020 - 15:24:06
Django version 3.0.8, using settings 'mysite.settings'
Starting development server at http://0.0.0.0:8000/
Quit the server with CTRL-BREAK.
```

브라우저에서 들어갈 URL <http://127.0.0.1:8000/admin>

Django 관리

사용자 이름:

비밀번호:

로그인

Django 관리

환영합니다, **BHWANG99**. [사이트 보기](#) / [비밀번호 변경](#) / [로그아웃](#)

사이트 관리

BOOKMARK	
Bookmarks	+ 추가 ✎ 변경
인증 및 권한	
그룹	+ 추가 ✎ 변경
사용자(들)	+ 추가 ✎ 변경

최근 활동

나의 활동

이용할 수 없습니다.



6. 애플리케이션 만들기

웹 사이트의 전체 프로그램 또는 모듈화된 단위 프로그램을 애플리케이션이라고 합니다. 즉, 프로그램으로 코딩할 대상을 애플리케이션이라고 부른다

사이트에 대한 전체 프로그램을 프로젝트^{Project}라 하고, 모듈화된 단위 프로그램을 애플리케이션^{Application}이라 부른다.



6. 애플리케이션 만들기

```
(vDjBook)C:\Anaconda3\envs\vDjBook\ch99>python manage.py startapp bookmark
```

(C:) > 사용자 > WBH > venv > vDjBook > ch99 >

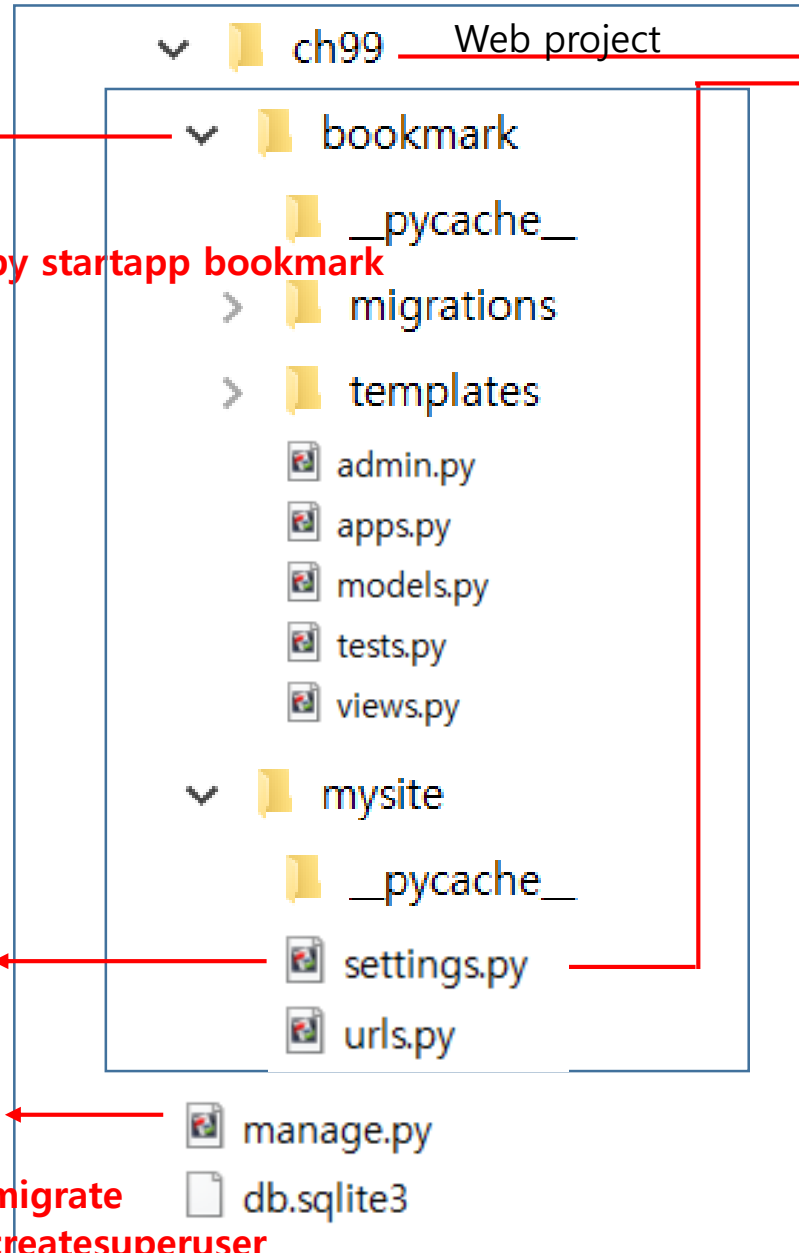
이름	수정한 날짜	유형	크기
bookmark	2020-08-04 오전 10:54	파일 폴더	
mysite	2020-08-04 오전 10:48	파일 폴더	
db.sqlite3	2020-08-04 오전 10:52	SQLITE3 파일	128KB
manage.py	2020-08-03 오후 8:53	PY 파일	1KB

Application 이름

settings.py 파일에 application 등록

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

가상환경 vDjBook → 1. Anaconda Navigator를 이용하여 생성



2. (vDjBook) > **django-admin startproject mysite**

실행하면 mysite project 폴더와 그 하위에
mysite 폴더 생성

상위 mysite 폴더를 ch99로 수정 또는 다른 이름도 가능

5. (vDjBook) ch99> **python manage.py startapp bookmark**

application 폴더 bookmark 생성

3. settings.py 파일 변경

4. 기본 테이블 생성 및
슈퍼유저 생성

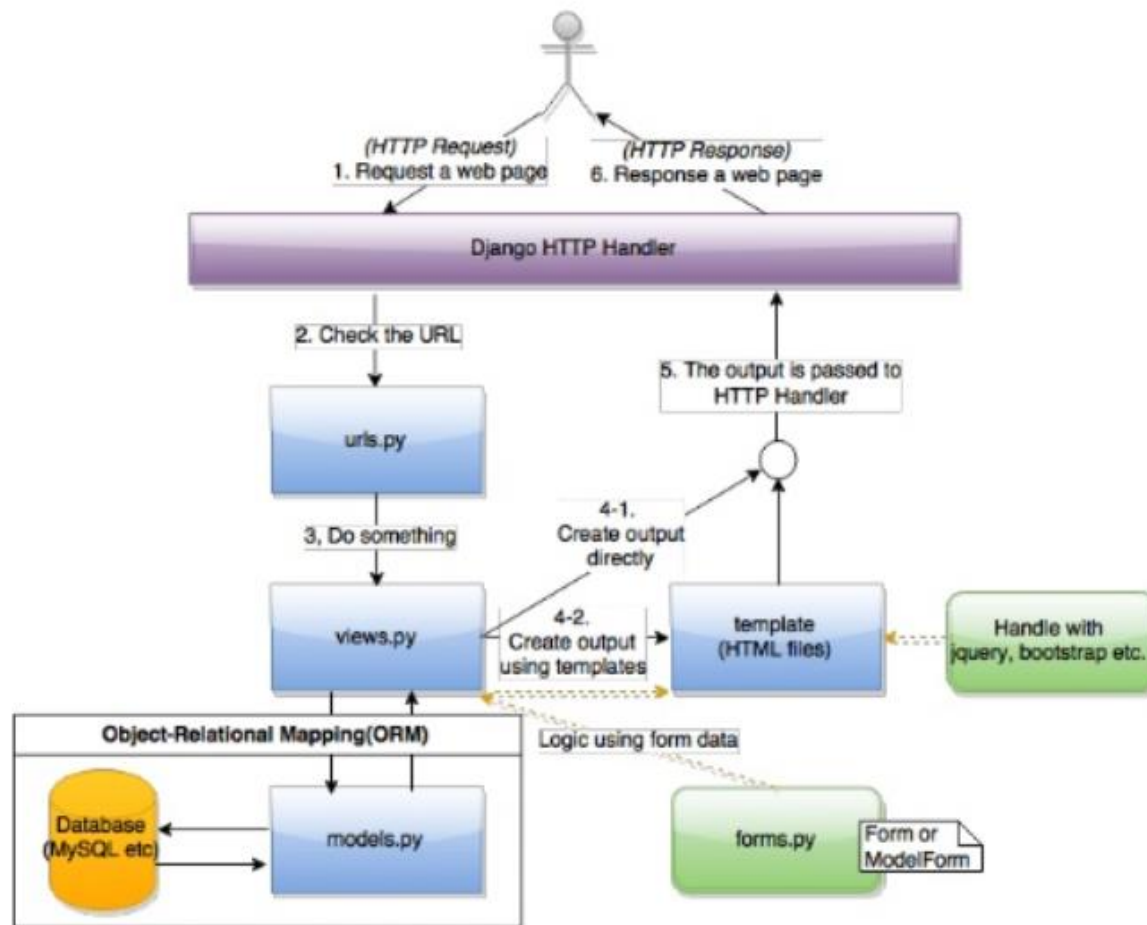
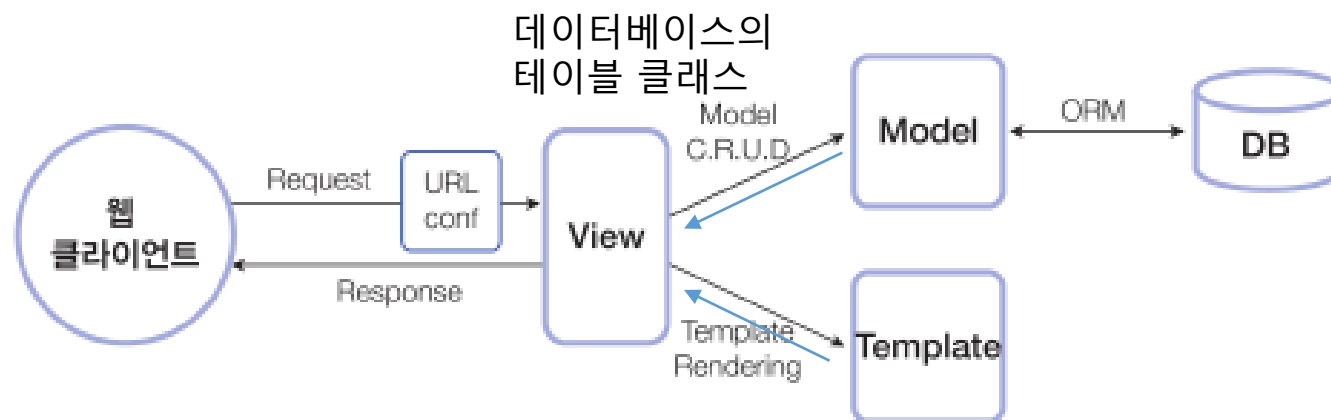
ch99> **python manage.py migrate**

ch99> **python manage.py createsuperuser**

6. settings.py 파일에 application 등록

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

2.3 개발 코딩하기



ORM: Class의 Instance를 만들어 DB에 접근

[Model Template View 소프트웨어 아키텍처 패턴]

[출처] <https://www.slideshare.net/DustinJunginSeoul/qna-blog-using-django-orm>



2.3 개발 코딩하기 - 모델

Model - 데이터베이스 정의

모델이란 사용될 데이터에 대한 정의를 담고 있는 장고의 클래스입니다. 장고는 ORM 기법을 사용하여 애플리케이션에서 사용할 데이터베이스를 클래스로 매핑해서 코딩할 수있음

```
from django.db import models

class Person(models.Model):
    first_name = models.CharField(max_length=30)
    last_name = models.CharField(max_length=30)
```

models.py 예제

```
CREATE TABLE myapp_person (
    "id" serial NOT NULL PRIMARY KEY,
    "first_name" varchar(30) NOT NULL,
    "last_name" varchar(30) NOT NULL
);
```

sql 문장



2.3 개발 코딩하기 - 모델

Model - 데이터베이스 정의

장고에서는 ORM(Object Relation Mapping) 기법에 따라 테이블을 하나의 클래스로 정의하고, 테이블의 컬럼은 클래스의 변수로 매핑합니다. 테이블 클래스는 `django.db.models.Model` 클래스를 상속받아 정의하고, 각 클래스 변수의 타입도 장고에서 미리 정의해 놓은 필드 클래스를 사용합니다.

테이블의 신규 생성, 테이블의 정의 변경 등 `models.py` 파일에서 데이터베이스 변경 사항이 발생하면, 이를 데이터베이스에 실제로 반영해주는 작업을 `makemigrations` 및 `migrate` 명령을 통해서 해야 합니다.

7. 모델 → URL → VIEW → TEMPLATE 코딩



2.3 개발 코딩하기 - 모델

2.3.1 테이블 정의 2.3.2 Admin 사이트에 테이블 반영

테이블을 새로 만들 때는 models.py와 admin.py 두 개의 파일을 함께 수정해야 한다

bookmark>models.py

```
from django.db import models

# Create your models here.
class Bookmark(models.Model):
    title = models.CharField('TITLE', max_length=100, blank=True)
    url = models.URLField('URL', unique=True)

    def __str__(self):
        return self.title
```

이 테이블의 객체 변수를 읽을때 출력하는 컬럼

테이블 : 클래스

각 레코드 : 클래스의 객체

객체를 저장하는 변수 → 객체변수

admin.py 에 등록을 해야 admin 화면에서 관리할 수 있다.

BookmarkAdmin class : Bookmark 내용을 보여줄 때 Admin 사이트에서 어떤 모습으로 보여줄지 정의

bookmark>admin.py

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

# Register your models here.

@admin.register(Bookmark)
class BookmarkAdmin(admin.ModelAdmin):
    list_display=('id','title','url')
```

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.3 개발 코딩하기 - 모델

2.3.1 테이블 정의 2.3.2 Admin 사이트에 테이블 반영

bookmark>admin.py

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

# Register your models here.

@admin.register(Bookmark)
class BookmarkAdmin(admin.ModelAdmin):
    list_display=('id','title','url')
```

변경할 bookmark 선택



액션: 실행 3 중 아무것도 선택되지 않았습니다.

<input type="checkbox"/>	ID	TITLE	URL
<input type="checkbox"/>	3	Google First Page	https://www.google.co.kr
<input type="checkbox"/>	2	Daum Homepage	https://www.daum.net
<input type="checkbox"/>	1	Naver Homepage	https://www.naver.com

3 bookmarks

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.3 개발 코딩하기 - 모델

2.3.3 데이터베이스 변경 사항 반영



```
(vDjBook)C:\Anaconda3\envs\vDjBook\ch99>python manage.py makemigrations bookmark
```

```
Migrations for 'bookmark':  
bookmark\migrations\0001_initial.py  
- Create model Bookmark
```

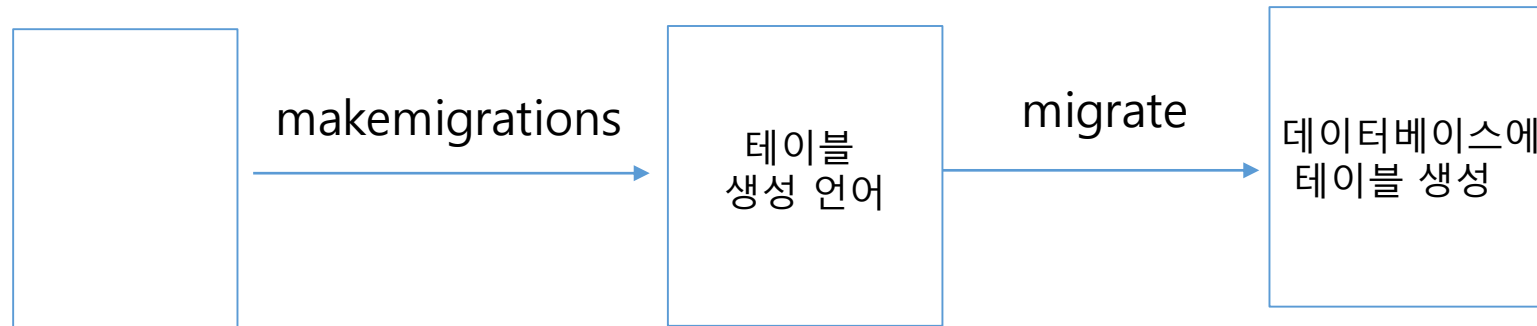
```
(vDjBook)C:\Anaconda3\envs\vDjBook\ch99>python manage.py migrate
```

```
Operations to perform:  
Apply all migrations: admin, auth, bookmark, contenttypes, sessions  
Running migrations:  
Applying bookmark.0001_initial... OK
```

bookmark/migrations 디렉토리의 마이그레이션 파일들을 이용해 DB에 테이블을 생성

bookmark/migrations 디렉토리 하위에 마이그레이션 파일들이 생김. DB에 테이블을 생성할 명령어가 담긴 파일

models.py



7. 모델 → URL → VIEW → TEMPLATE 코딩

2.3 개발 코딩하기 - 모델

2.3.3 데이터베이스 변경 사항 반영 - 테이블 생성 후 데이터 추가



① (vDjBook)C:\Anaconda3\envs\WvDjBook\ch99>python manage.py runserver 0.0.0.0:8000

② localhost:8000/admin

③ Django 관리

사용자 이름:

비밀번호:

로그인

④

BOOKMARK	
Bookmarks	+ 추가 변경
인증 및 권한	
그룹	+ 추가 변경
사용자(들)	+ 추가 변경

나의 글

- + Bohyun Wang Positive Post
- + aa Post
- + Google First Page Bookmark
- + Daum Homepage Bookmark
- + Naver Homepage Bookmark

⑤ Django 관리

환영합니다, BHWANG99. [사이트 보기](#) / [비밀번호 변경](#) / [로그아웃](#)

홈 > Bookmark > Bookmarks > bookmark 추가

bookmark 추가

TITLE:

URL:

저장 및 다른 이름으로 추가 저장 및 편집 계속 **저장**



2.4 개발 코딩하기 – URLconf

파이썬의 URL 정의 방식은 전통적인 자바나 PHP 계열의 URL보다 직관적이고 이해하기가 쉬움.

그래서 이런 방식을 우아한 Elegant URL이라고 부르는 것.

URL을 정의하기 위해서는 다음 예제처럼 **urls.py** 파일에 URL과 처리 함수(뷰 View라고 부름)를 매핑하는 파이썬 코드를 작성하면 됨

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.4 개발 코딩하기 – URLconf

ch99>mysite>urls.py

```
from django.contrib import admin
from django.urls import path
from bookmark.views import BookmarkLV, BookmarkDV
```

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

ch99>bookmark>views.py

```
class BookmarkLV(ListView):
    model = Bookmark

class BookmarkDV(DetailView):
    model = Bookmark
```

int형 매개변수 pk는 매개변수명 view 메소드나 함수에 매개변수로 넘겨 줌
BookmarkDV.as_view(pk=값) 과 같이 실행

예) 만약 url bookmark/1/ 이 요청되면 BookmarkDV.as_view(pk=1) 이 호출

URL과 뷰 메소드 매핑 , URL을 클릭하면 매핑된 메소드(뷰 파일에 정의된 메소드) 실행



2.5 개발 코딩하기 - 뷰

View - 로직 정의

뷰는 웹 요청을 받아서 데이터베이스 접속 등 해당 애플리케이션의 로직에 맞는 처리를 하고, 그 결과 데이터를 HTML로 변환하기 위하여 템플릿 처리를 한 후에, 최종 HTML로 된 응답 데이터를 웹 클라이언트로 반환하는 역할

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.5 개발 코딩하기 - 뷰

ch99>bookmark>views.py

```
from django.shortcuts import render
from django.views.generic import ListView, DetailView
from bookmark.models import Bookmark
```

Create your views here.

```
class BookmarkLV(ListView):
```

```
    model = Bookmark
```

뷰에서 사용하게 될 데이터 클래스
(model 파일에 정의된 테이블)

```
class BookmarkDV(DetailView):
```

```
    model = Bookmark
```

models.py

```
# Create your models here.
class Bookmark(models.Model):
    ...
    def __str__(self):
        return self.title
```

object_list

id	title	url

bookmark list.html

```
<ul>
    {% for bookmark in object_list %}
    <li><a href="{% url 'detail' bookmark.id %}">{{ bookmark }}</a></li>
    {% endfor %}
</ul>
```

템플릿 렌더링

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Django Bookmark List</title>
5   </head>
6   <body>
7     <div id="content">
8       <h1>Bookmark List</h1>
9       <ul>
10
11         <li><a href="/bookmark/1/">Naver Homepage</a></li>
12
13         <li><a href="/bookmark/2/">Daum Homepage</a></li>
14
15         <li><a href="/bookmark/3/">Google First Page</a></li>
16
17       </ul>
18     </div>
19   </body>
20 </html>
```

BookmarkLV 클래스에서 제공하는 속성 2가지

컨텍스트 변수 **object_list** : DB에서 가져온 data set 저장

템플릿 파일명 : 디폴트로 모델명소문자_list.html 여기서는

bookmark_list.html

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.5 개발 코딩하기 - 뷰

ch99>bookmark>views.py

```
from django.shortcuts import render
from django.views.generic import ListView, DetailView
from bookmark.models import Bookmark

# Create your views here.
class BookmarkLV(ListView):
    model = Bookmark

class BookmarkDV(DetailView): pk=? 인 레코드를 select
    model = Bookmark
```

BookmarkDV 클래스에서 제공하는 속성 2가지

컨텍스트 변수 **object** : DB에서 가져온 특정 레코드 저장

템플릿 파일명 : 디폴트로 모델명소문자_detail.html 여기서는

bookmark_detail.html

```
# Create your models here.
class Bookmark(models.Model):
    ...
    def __str__(self):
        return self.title
```

object

id	title	url
?		

bookmark_detail.html

```
<h1>{{ object.title }}</h1>
<ul>
    <li>URL : <a href="{{ object.url }}">{{ object.url }}</a></li>
</ul>
```

↓ 템플릿 렌더링

```
<!DOCTYPE html>
<html>
  <head>
    <title>Django Bookmark Detail</title>
  </head>
  <body>
    <div id="content">
      <h1>Naver Homepage</h1>
      <ul>
        <li>URL : <a href="https://www.naver.com">https://www.naver.com</a></li>
      </ul>
    </div>
  </body>
</html>
```



2.6 개발 코딩하기 - 템플릿

Template - 화면 UI 정의

개발자가 응답에 사용할 *.html 파일을 작성하면, 장고는 이를 해석해서 최종 HTML 텍스트 응답을 생성하고(이 과정을 템플릿 렌더링이라함), 이를 클라이언트에게 보내줍니다. 클라이언트(보통 웹 브라우저)는 응답으로 받은 HTML 텍스트를 해석해서 우리가 보는 웹 브라우저 화면에 UI를 보여주는 것

템플릿 파일은 *.html 확장자를 가지며, 장고의 템플릿 시스템 문법에 맞게 작성

장고에서 템플릿 파일을 찾을 때는 settings.py 파일에 등록된 TEMPLATES 및 INSTALLED_APPS에서 지정된 앱의 디렉토리를 검색.

MVT 방식에서 사용자에게 보여주는 화면, 즉 UI_{User Interface}를 담당하고 있는 기능이 템플릿 시스템.

템플릿 코드를 작성 시에 HTML 코드와 장고의 템플릿 코드가 섞이지만, 중요한 점은 템플릿에서는 로직을 표현하는 것이 아니라 사용자에게 어떻게 보여줄지에 대한 룩앤필을 표현한 다는 것임

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.6 개발 코딩하기 - 템플릿

ch99>bookmark>templates>bookmark>bookmark_list.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Django Bookmark List</title>
  </head>
  <body>
    <div id="content">
      <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>
```

BookmarkLV 클래스에서 제공하는 리스트형 변수

객체 변수: 모델파일에서 return self.title

path('bookmark/<int:pk>/', BookmarkDV.as_view(), name='detail')

다시쓰면 title <a>



object_list

id	title	url

bookmark

bookmark.id
bookmark.title
bookmark.url

bookmark : 객체 변수

```
def __str__(self):
    return self.title
```

객체 변수를 출력하면 위에 정의된 대로 title을 출력

테이블 : 클래스

각 레코드 : 클래스의 객체

객체를 저장하는 변수 → 객체변수

7. 모델 → URL → VIEW → TEMPLATE 코딩



2.6 개발 코딩하기 - 템플릿

ch99>bookmark>templates>bookmark>bookmark_list.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>Django Bookmark List</title>
  </head>
  <body>
    <div id="content">
      <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>
```

실행결과

Bookmark List

- [Naver Homepage](#)
- [Daum Homepage](#)
- [Google First Page](#)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Django Bookmark List</title>
5   </head>
6   <body>
7     <div id="content">
8       <h1>Bookmark List</h1>
9       <ul>
10
11         <li><a href="/bookmark/1/">Naver Homepage</a></li>
12
13         <li><a href="/bookmark/2/">Daum Homepage</a></li>
14
15         <li><a href="/bookmark/3/">Google First Page</a></li>
16
17       </ul>
18     </div>
19   </body>
20 </html>
```

7. 모델 → URL → VIEW → TEMPLATE 코딩

2.6 개발 코딩하기 - 템플릿

ch99>bookmark>templates>bookmark>bookmark_detail.html

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Django Bookmark Detail</title>
```

```
  </head>
```

```
  <body>
```

```
    <div id="content"> detail_view를 상속받은 view  
                        클래스에서 제공하는 변수
```

```
      <h1>{{ object.title }}</h1>
```

```
      <ul>
```

```
        <li>URL : <a href="{{ object.url }}">{{ object.url }}</a></li>
```

```
      </ul>
```

```
    </div>
```

```
  </body>
```

```
</html>
```

실행결과

Naver Homepage

- URL : <https://www.naver.com>

id	title	url

```
<!DOCTYPE html>
```

```
<html>
```

```
  <head>
```

```
    <title>Django Bookmark Detail</title>
```

```
  </head>
```

```
  <body>
```

```
    <div id="content">
```

```
      <h1>Naver Homepage</h1>
```

```
      <ul>
```

```
        <li>URL : <a href="https://www.naver.com">https://www.naver.com</a></li>
```

```
      </ul>
```

```
    </div>
```

```
  </body>
```

```
</html>
```

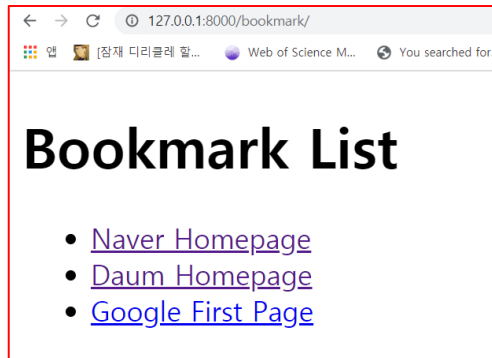
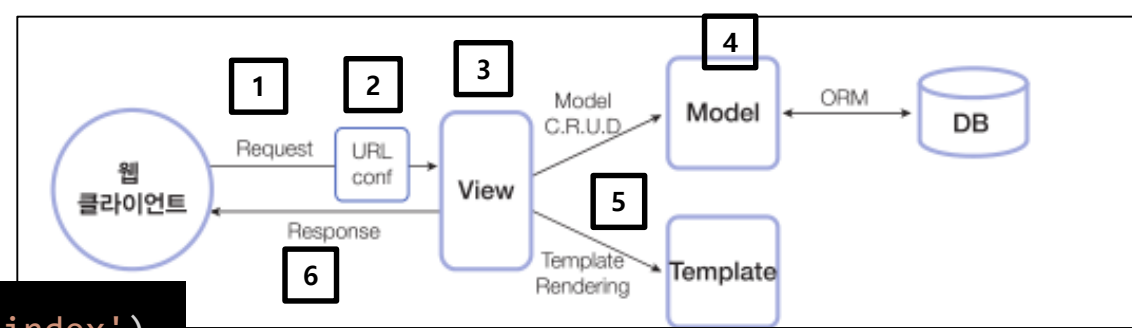
1 request



ch99>mysite>urls.py

```
path('bookmark/', BookmarkLV.as_view(), name='index'),
```

2 URLconf



6

views.py

```
class BookmarkLV(ListView):
    model = Bookmark
```

object_list

4

※object_list

id	title	url



models.py

```
class Bookmark(models.Model):
    title = models.CharField('TITLE', max_length=100, blank=True)
    url = models.URLField('URL', unique=True)
```

bookmark_list.html

```
<ul>
  {% for bookmark in object_list %}
  <li><a href="{% url 'detail' bookmark.id %}">{{ bookmark }}</a></li>
  {% endfor %}
</ul>
```

ListView 클래스를 상속받은 뷰 클래스 : **object_list** 라는 멤버변수 제공
object_list : Bookmark 테이블의 모든 레코드가 저장되어 있음



```
<ul>
  {% for bookmark in object_list %}
  <li><a href="{% url 'detail' bookmark.id %}">{{ bookmark }}</a></li>
```

Bookmark List

- [Naver Homepage](#)
- [Daum Homepage](#)
- [Google First Page](#)

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>Django Bookmark List</title>
5   </head>
6   <body>
7     <div id="content">
8       <h1>Bookmark List</h1>
9       <ul>
10
11         <li><a href="/bookmark/1/">Naver Homepage</a></li>
12         <li><a href="/bookmark/2/">Daum Homepage</a></li>
13         <li><a href="/bookmark/3/">Google First Page</a></li>
14
15       </ul>
16     </div>
17   </body>
18 </html>
```

1 request

2 URLconf

```
path('bookmark/<int:pk>/', BookmarkDV.as_view(), name='detail'),
```

urls.py

3

views.py

```
class BookmarkDV(DetailView):
  model = Bookmark
```



models.py

4

```
class Bookmark(models.Model):
  title = models.CharField('TITLE', max_length=100, blank=True)
  url = models.URLField('URL', unique=True)
```

object

<int:pk>에 해당하는
한 개의 레코드

id	title	url

5

```
<h1>{{ object.title }}</h1>
<ul><li>URL : <a href="{% url 'detail' object.url %}">{{ object.url }}</a></li></ul>
```

bookmark_detail.html

Naver Homepage

- URL : <https://www.naver.com>



템플릿 변수 : Variables are surrounded by `{{` and `}}` like this:

템플릿 시스템은 변수를 평가해서 변수값으로 출력.
변수명은 일반 프로그래밍의 변수명처럼 문자, 숫자, 밑줄(_)을 사용하여 이름을 정의

```
{{ my_dict.key }}
```

```
{{ my_object.attribute }}
```

```
{{ my_list.0 }}
```




템플릿 필터

아래의 예시처럼 파이프(|) 문자를 사용.
name 변수값의 모든 문자를 소문자로 바꿔주는 필터

```
{{ name | lower }}
```

필터를 체인으로 연결할 수도 있음

```
{{ text | escape | linebreaks }}
```

다음은 bio 변수값 중에서 앞에 30개의 단어만 보여 주고,
줄 바꿈 문자는 모두 없애줌. 필터에 30과 같은 인자를 줄 수 있음

```
{{ bio|truncatewords:30 }}
```



템플릿 필터

필터의 인자에 빈칸이 있는 경우는 따옴표로 묶어줌

```
{{ list | join : " // " }}
```

value 변수값이 False이거나 없는 경우, “nothing”으로 보여줌

```
{{ value | default : "nothing" }}
```

value 변수값의 길이를 반환함

```
{{ value | length }}
```

value 변수값에서 HTML 태그를 모두 없애줌

```
{{ value | striptags }}
```



템플릿 태그

provide arbitrary logic in the rendering process.
surrounded by {% and %} like this

{% if %} 태그

```
{% if athlete_list %}
    Number of athletes: {{ athlete_list|length }}
{% elif athlete_in_locker_room_list %}
    Athletes should be out of the locker room soon!
{% else %}
    No athletes.
{% endif %}
```

{% for %} 태그

```
<ul>
{% for athlete in athlete_list %}
    <li>{{ athlete.name }}</li>
{% endfor %}
</ul>
```

{% csrf_token %} 태그

{% url %} 태그



템플릿 태그

표 . for 태그에 사용되는 변수들

Variable	Description
<code>forloop.counter</code>	The current iteration of the loop (1-indexed)
<code>forloop.counter0</code>	The current iteration of the loop (0-indexed)
<code>forloop.revcounter</code>	The number of iterations from the end of the loop (1-indexed)
<code>forloop.revcounter0</code>	The number of iterations from the end of the loop (0-indexed)
<code>forloop.first</code>	True if this is the first time through the loop
<code>forloop.last</code>	True if this is the last time through the loop
<code>forloop.parentloop</code>	For nested loops, this is the loop surrounding the current one



템플릿 주석 첫번째는 한 줄 주석문으로, 아래처럼 `{# #}` 형식을 따름. 한 문장의 전부 또는 일부를 주석 처리하는 방법

```
{# greeting #}hello
```

`{# #}` 주석문 내에 템플릿 코드가 들어있어도 정상적으로 주석 처리됨

```
{# {% if foo %} bar {% else %} #}
```

여러 줄의 주석문으로, 아래처럼 `{% comment %}` 태그를 사용합니다

```
{% comment "optional note " %}
```

```
<p><commented out text here</p>
```

```
{% endcomment %>
```



HTML 이스케이프

사용자가 입력한 데이터를 그대로 렌더링하는 것은 위험
장고는 자동 이스케이프 기능을 제공.
즉 장고는 디폴트 로 HTML에 사용되는 예약 문자들을 아래처럼 예약 의미를
제거한 문자로 변경해주는 기능을 제공

< (less than) 문자는 < 로 변경함

> (greater than) 문자는 > 로 변경함

' (single quote) 문자는 ‘ 로 변경함

“ (double quote) 문자는 " 로 변경함

& (ampersand) 문자는 & 로 변경함