

Django 기반의 웹프로그래밍

이 소 영

yisy0703@naver.com

Web Frameworks for Python(<https://wiki.python.org/moin/WebFrameworks>)

장고 공식 사이트(<https://www.djangoproject.com/>)

장고 공식 소스 저장소(<http://github.com/django/django>)

장고 참조 문서 영어(<https://docs.djangoproject.com/en/5.1/>)

Agenda

Django(장고) 기반의 파이썬 웹 프로그래밍

Ch01. Django 시작하기

1. Django 란?
2. 개발 환경 구축
3. Django 구조

Ch02. Django App

1. Django Project
2. Model
3. View

Ch03. Model

1. Model 속성 및 옵션
2. Relationship
3. Migrations
4. Admin App

Ch04. Django SQL

1. Django shell
2. Manager & QuerySet
3. 조회 SQL
4. 생성/수정/삭제 SQL
5. Django-Debug-Toolbar

Ch05. Template

1. Template Loader
2. URL Dispatcher
3. Template 상속
4. Template Engines
5. Template Filter

Ch06. Django View

1. View 기본
2. View 활용

Ch07. Django Form

1. HTML form
2. CSRF
3. HttpRequest/HttpResponse
4. Django Form
5. Django Model Form
6. Form Validation

Ch08. File 관리

1. Static Files
2. Media Files
3. Image Thumbnail

Ch09. 사용자 인증

1. Auth App
2. 회원가입 구현
3. 로그인/아웃 구현
4. Oauth 라이브러리 활용

Ch 01 Django 시작하기

1. Django란?
2. 개발환경구축
3. Django 구조

Chapter 01. Django 시작하기

- Web Frameworks for Python(<https://wiki.python.org/moin/WebFrameworks>)
- 장고 공식 사이트(<https://www.djangoproject.com/>)
- **장고 공식 소스 저장소**(<http://github.com/django/django>)
- 장고 참조 문서 영어(<https://docs.djangoproject.com/en/5.1/>)
- 장고 참조 문서 한글(<https://docs.djangoproject.com/ko/5.1/intro/>)

개발 환경

- Anaconda Python 설치(<https://www.anaconda.com/download/>)
- Visual Studio Code 설치(<https://code.visualstudio.com/>) 나 PyCharm 설치(<https://www.jetbrains.com/ko-kr/pycharm/download>)



1. Django 란?

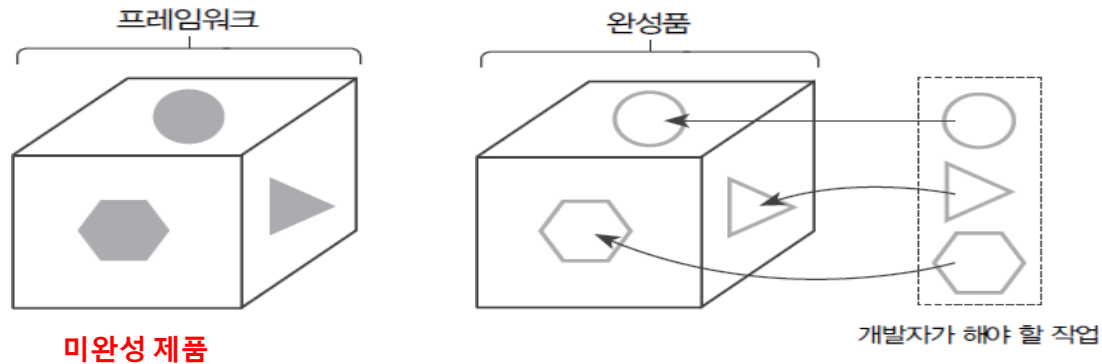


프레임워크

1. Django 란?

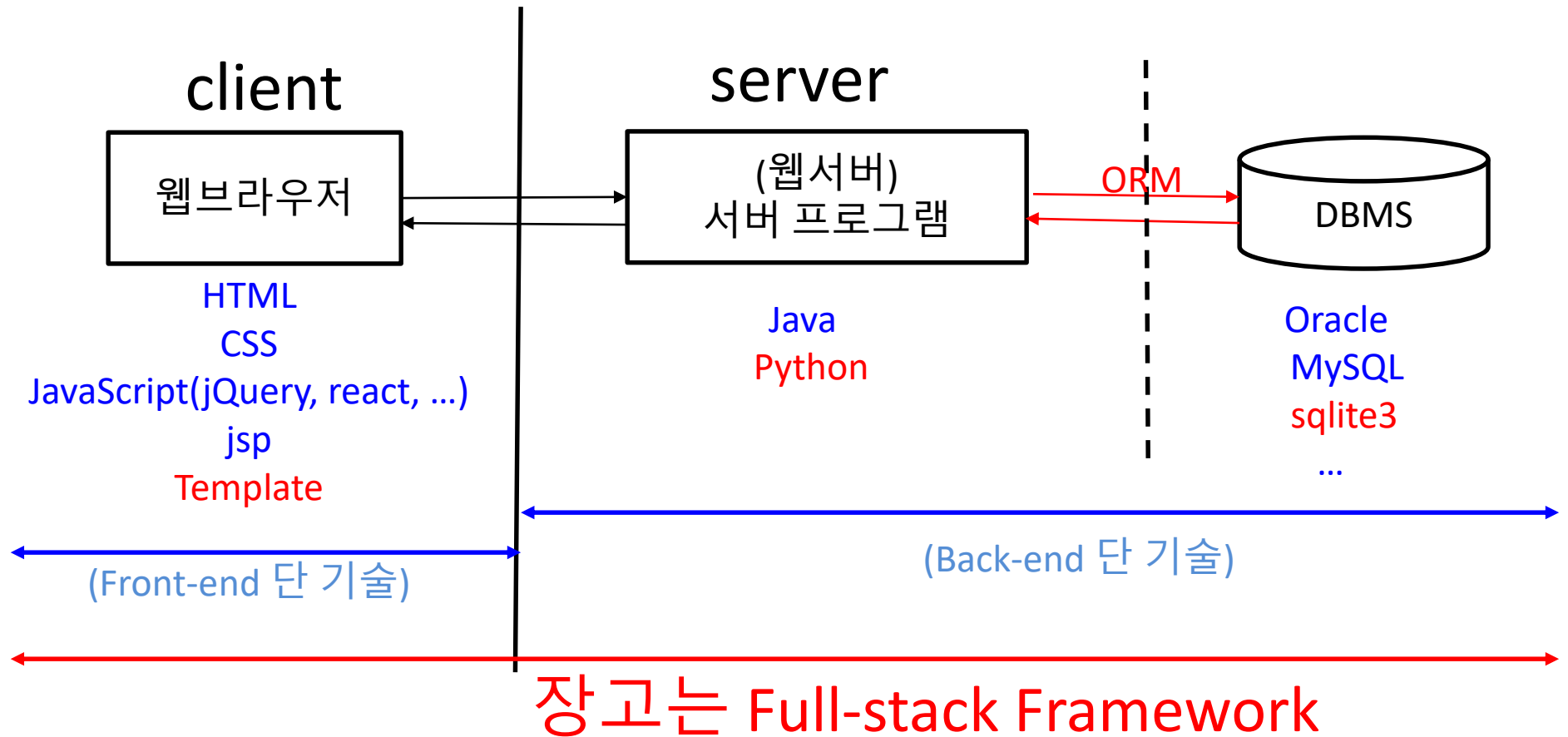
- 프레임워크와 라이브러리 차이

	프레임워크	라이브러리
코드작성	프레임워크 클래스를 서브 클래싱해서 작성	독립적으로 작성
호출흐름	프레임워크 코드가 유저 코드를 호출	유저코드가 라이브러리를 호출
실행흐름	프레임워크가 제어	유저 코드가 제어
객체연동	구조 프레임워크가 정의	독자적으로 정의



Web Framework for Python (<https://wiki.python.org/moin/WebFrameworks>)

1. Django 란?



웹 서비스 개발 파트

1. Django 란?

- 백엔드 개발
 - Django, FastAPI, Flask, Spring(Java), Node.js 등
- 서버운영
 - 자체 서버, 서버/웹 호스팅
 - AWS/Azure/Google 클라우드(IaaS와 PaaS) 등
- 프론트엔드 개발
 - HTML, CSS, JavaScript 기반 라이브러리 및 프레임워크
 - Vue.js, React.js, Angular.js, jQuery, Bootstrap 등
- 앱 개발
 - Android, iOS 개발

Python web Framework

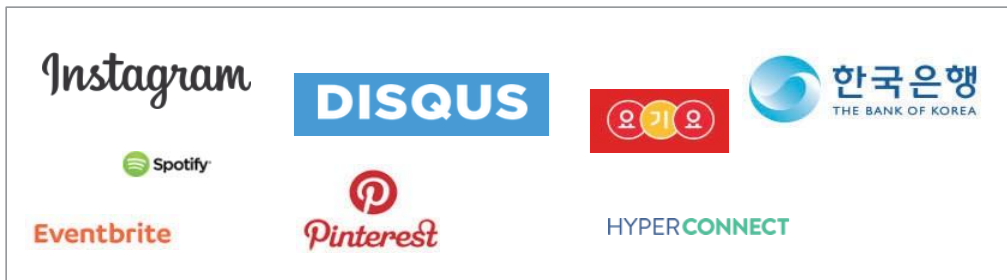
1. Django 란?

- <https://wiki.python.org/moin/WebFrameworks>
- Popular **Full-stack Frameworks**
 - Django
 - The Web Framework for perfectionists with deadlines.
- Popular Non Full-stack Frameworks
 - Flask
 - A microframework for Python based on Werkzeug.
 - FastAPI
 - A modern, fast (high-performance).

Django

1. Django 란?

- 백엔드를 담당하는 파이썬 풀스택 웹 프레임워크
- PG언어 시장 점유율 : <https://tiobe.com/tiobe-index>
 - 쉽고 빠른(풍부한 라이브러리) 인터프리터 언어의 특징으로 위부터 순서대로 실행
- Lawrence Journal-World 신문사에서 2003년부터 개발하여, 2005년에 세상에 공개
- 2008년에 1.0 릴리즈(Django Roadmap)
- 기타 연주자 Django Reinhardt 이름에서 작명
- <https://www.djangoproject.com>



Django 구성요소

1. Django 란?

Client - web Server - DB
Full stack

- Views : HTTP 요청처리 (server단 기술)
- Models : 데이터 베이스 처리 (DB 단 기술)
- Template : 사용자 인터페이스 처리 (client 측 기술)
- Form : 사용자 입력 데이터 처리 (client 측 기술)
- Admin App : 쉬운 DB 레코드 관리 UI (관리자 기능 제공)
- Loggin : 다양한 경로로 메시지 로깅
- Static files : 정적 파일 관리 (서버측에서 사용하는 파일)
- Media files : 사용자가 업로드한 파일 관리 (서버측에서 사용하는 파일)
- Message framework : 1회성 메시지 노출
- 테스트, 국제화 & 지역화, 캐싱, Send Emails
- Geographic, Syndication Feeds(RSS/Atom), Sitemaps



2. 개발 환경 구축



파이썬 설치

2. 개발 환경 구축

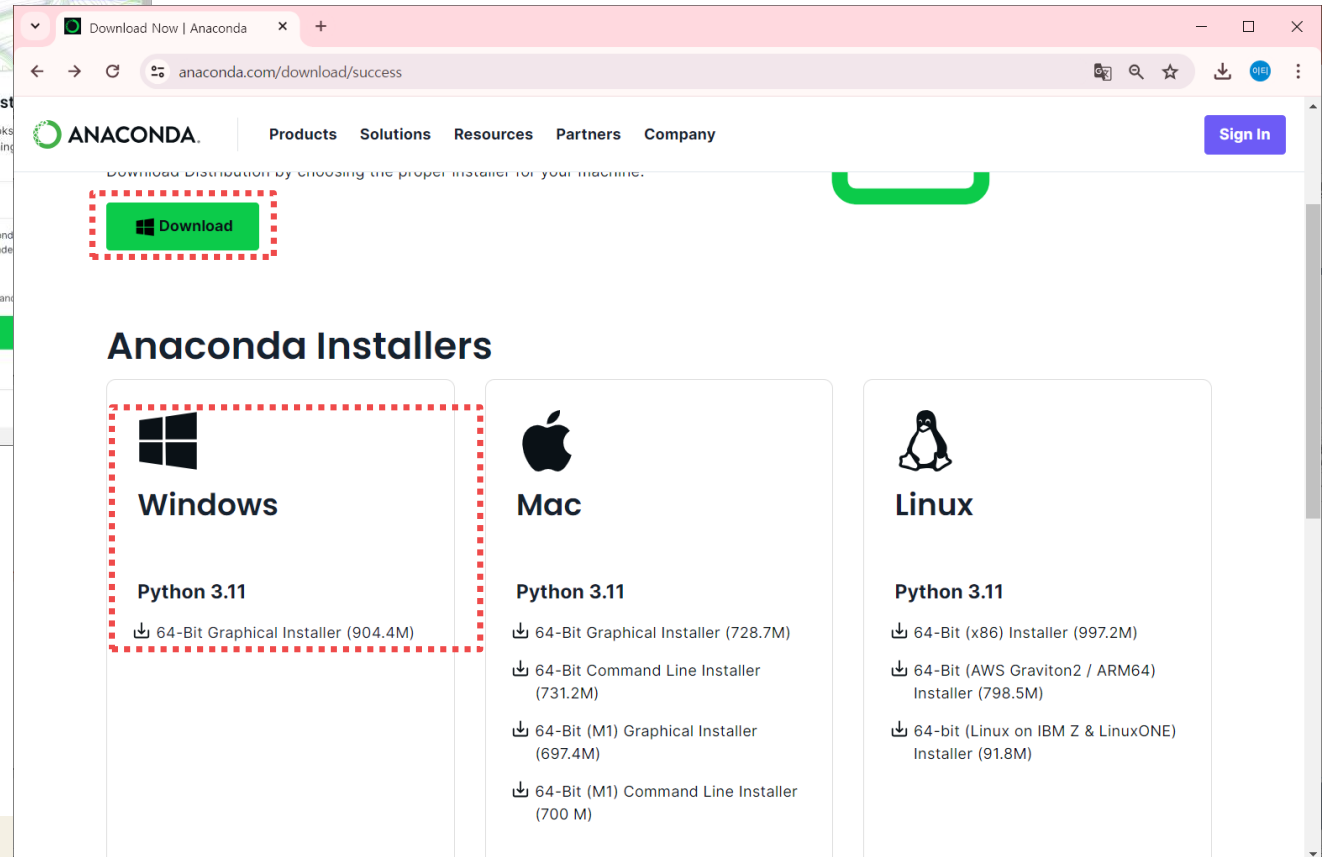
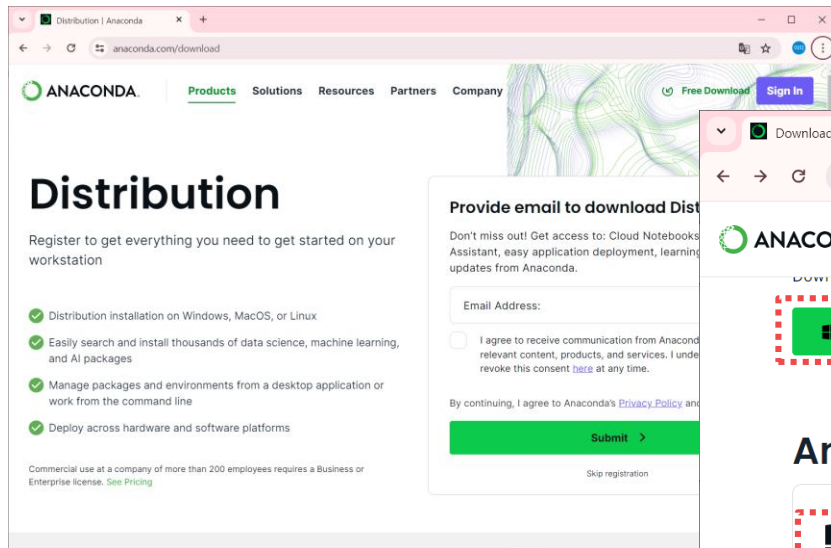
- 공식 배포판
 - <http://python.org>
- Anaconda Python
 - <https://www.anaconda.com>
 - 설치시 환경변수 PATH 옵션 체크(?)
- 파이썬은 1개만 설치
- 실제 웹서비스에서는 운영체제 배포판에서 제공하는 파이썬 사용

Anaconda Python 설치

2. 개발 환경 구축

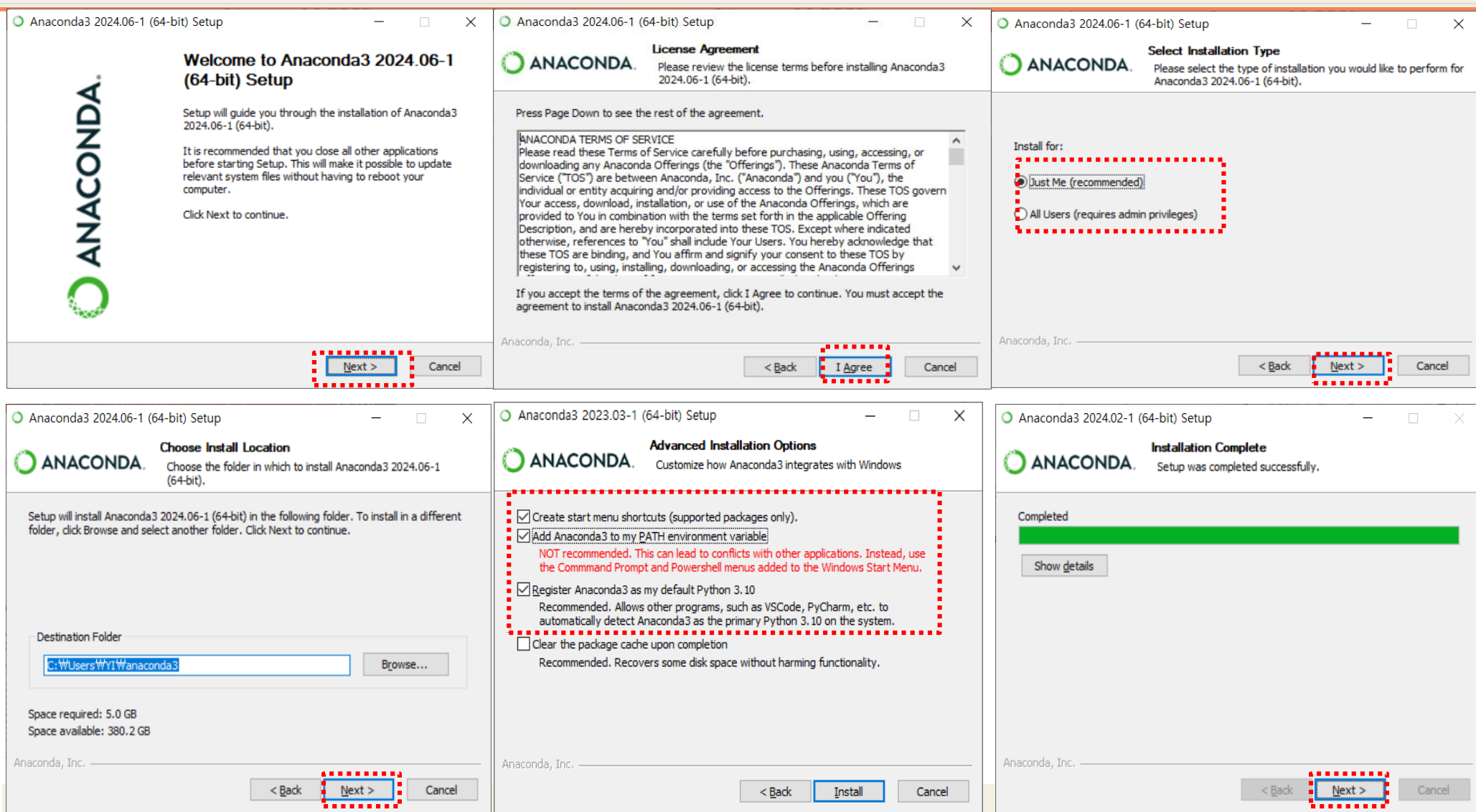
- <https://www.anaconda.com>

Anaconda3-2024.02-1-Windows-x86_64.exe 다운로드



Anaconda Python 설치

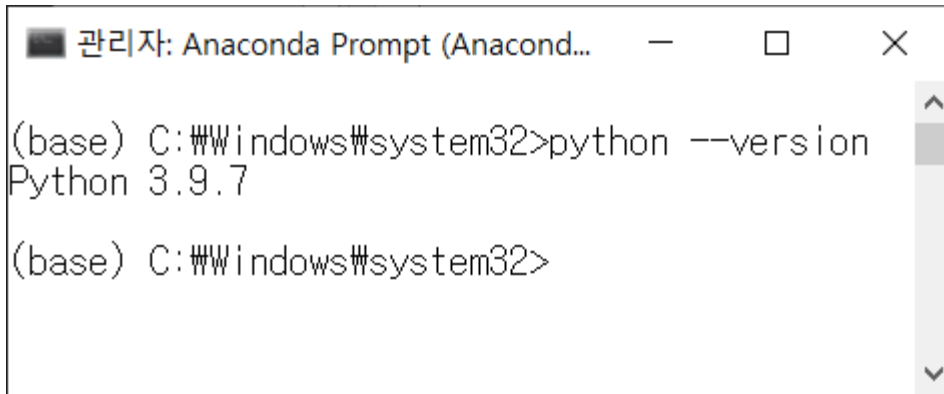
2. 개발 환경 구축



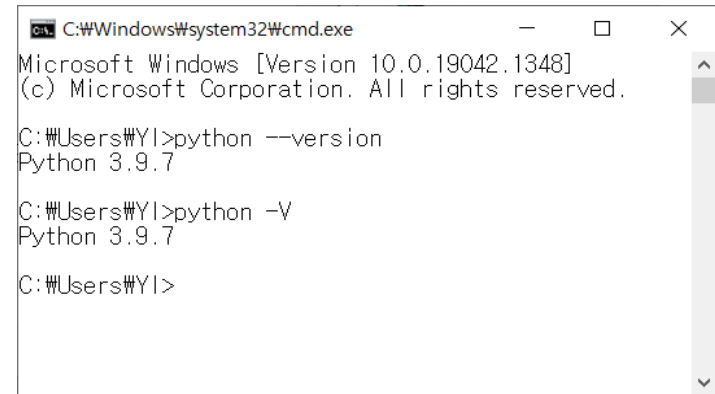
Python 버전 확인

2. 개발 환경 구축

- Python --version (window 키 + r -> cmd 나 anaconda prompt)



```
관리자: Anaconda Prompt (Anacond...  
(base) C:\Windows\system32>python --version  
Python 3.9.7  
(base) C:\Windows\system32>
```



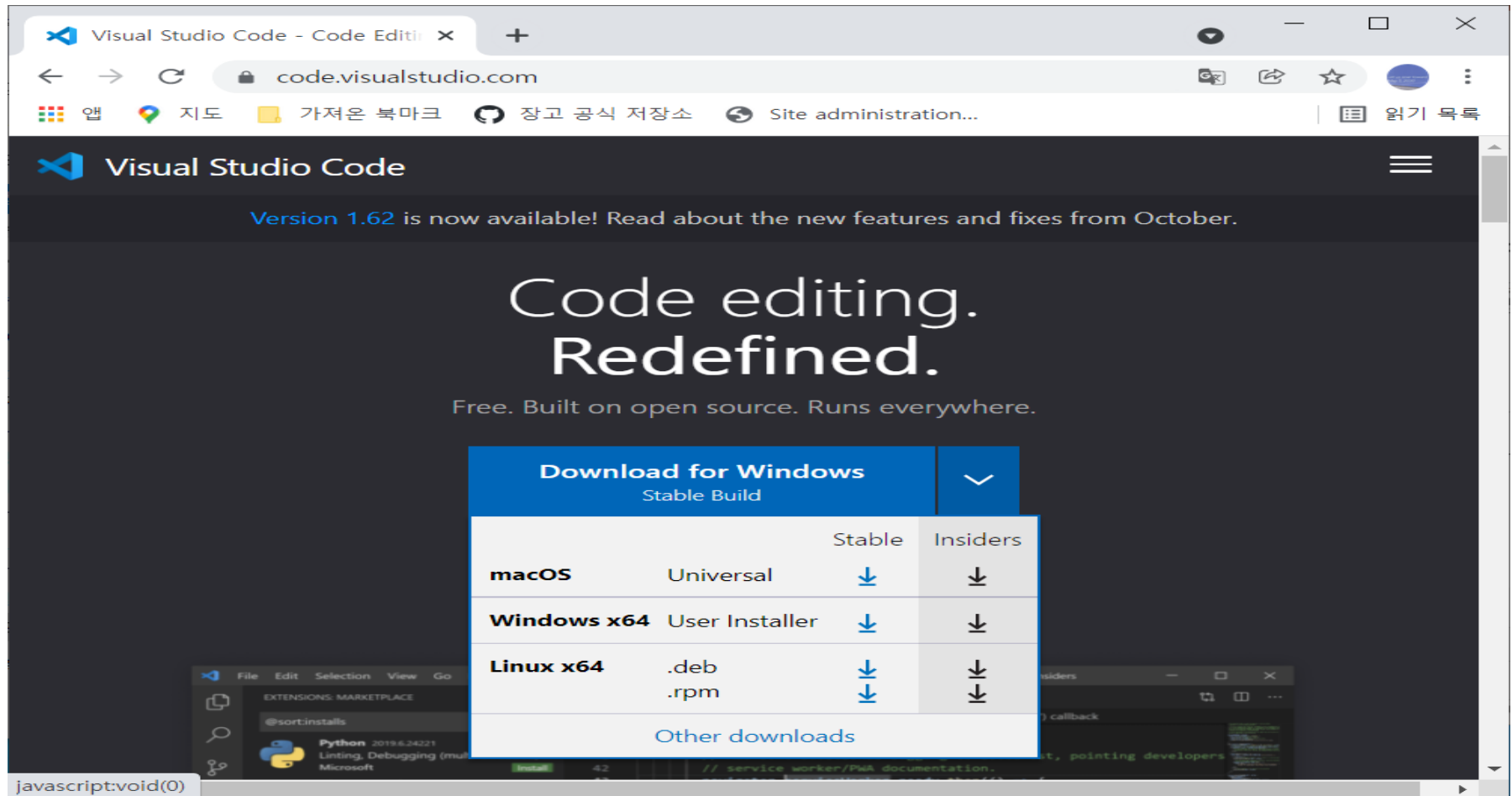
```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.19042.1348]  
(c) Microsoft Corporation. All rights reserved.  
C:\Users\YI>python --version  
Python 3.9.7  
C:\Users\YI>python -V  
Python 3.9.7  
C:\Users\YI>
```

- virtual env – 가상환경 구축 설치하면 독립적으로 작업도 가능하다.
수업시간에는 하나의 환경으로 할 예정

Visual Studio Code(편집기)

2. 개발 환경 구축

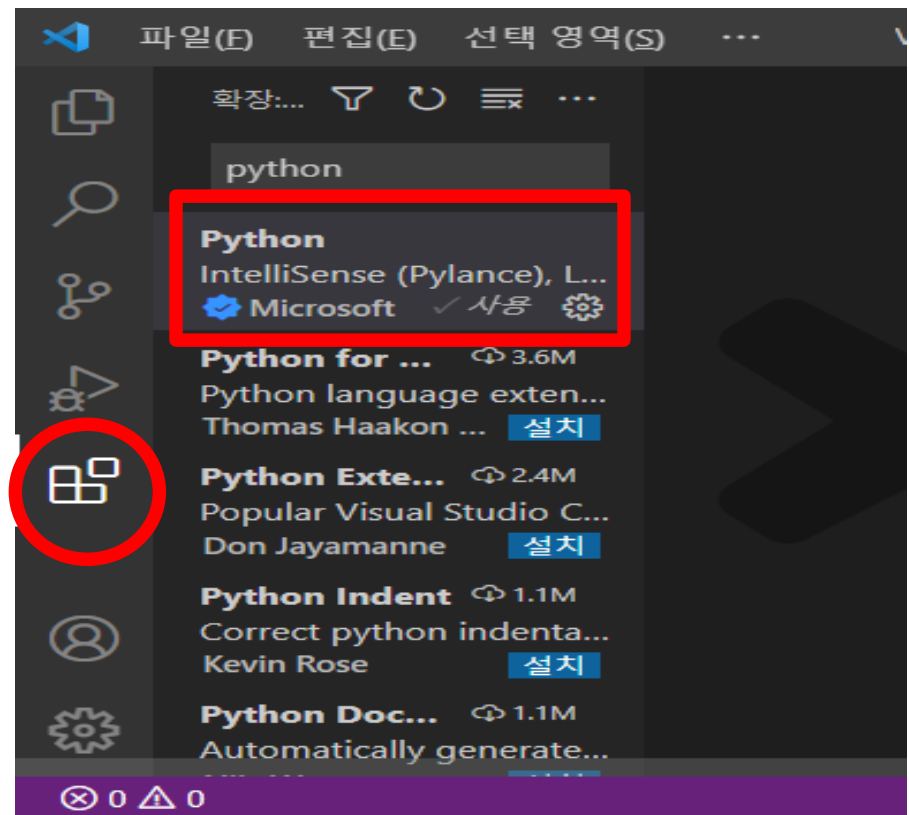
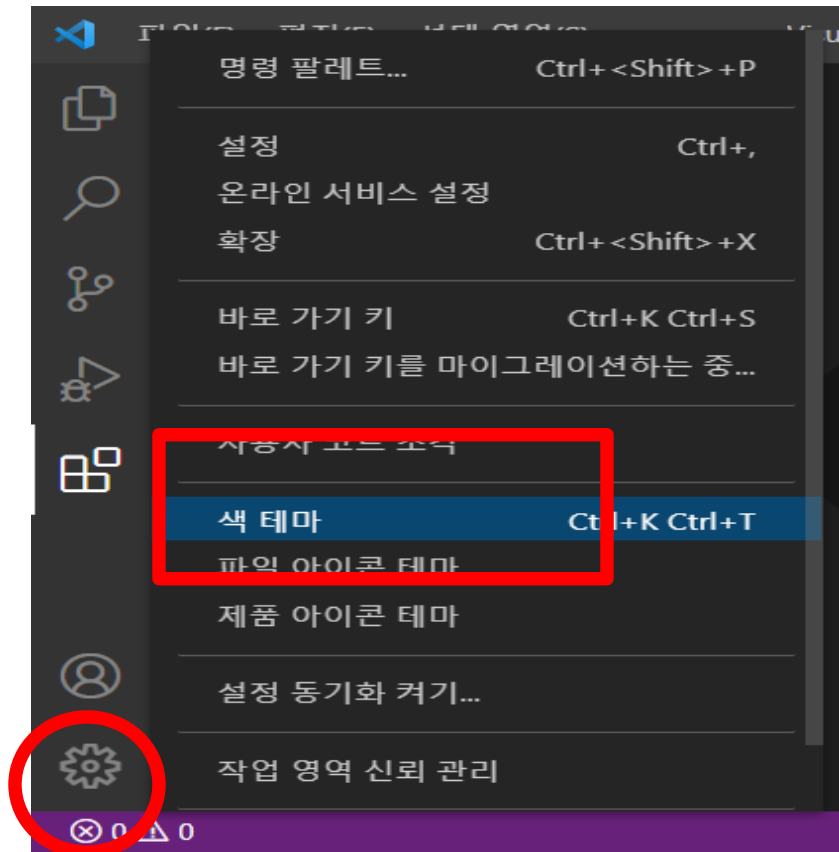
- <https://code.visualstudio.com>



Python 확장

2. 개발 환경 구축

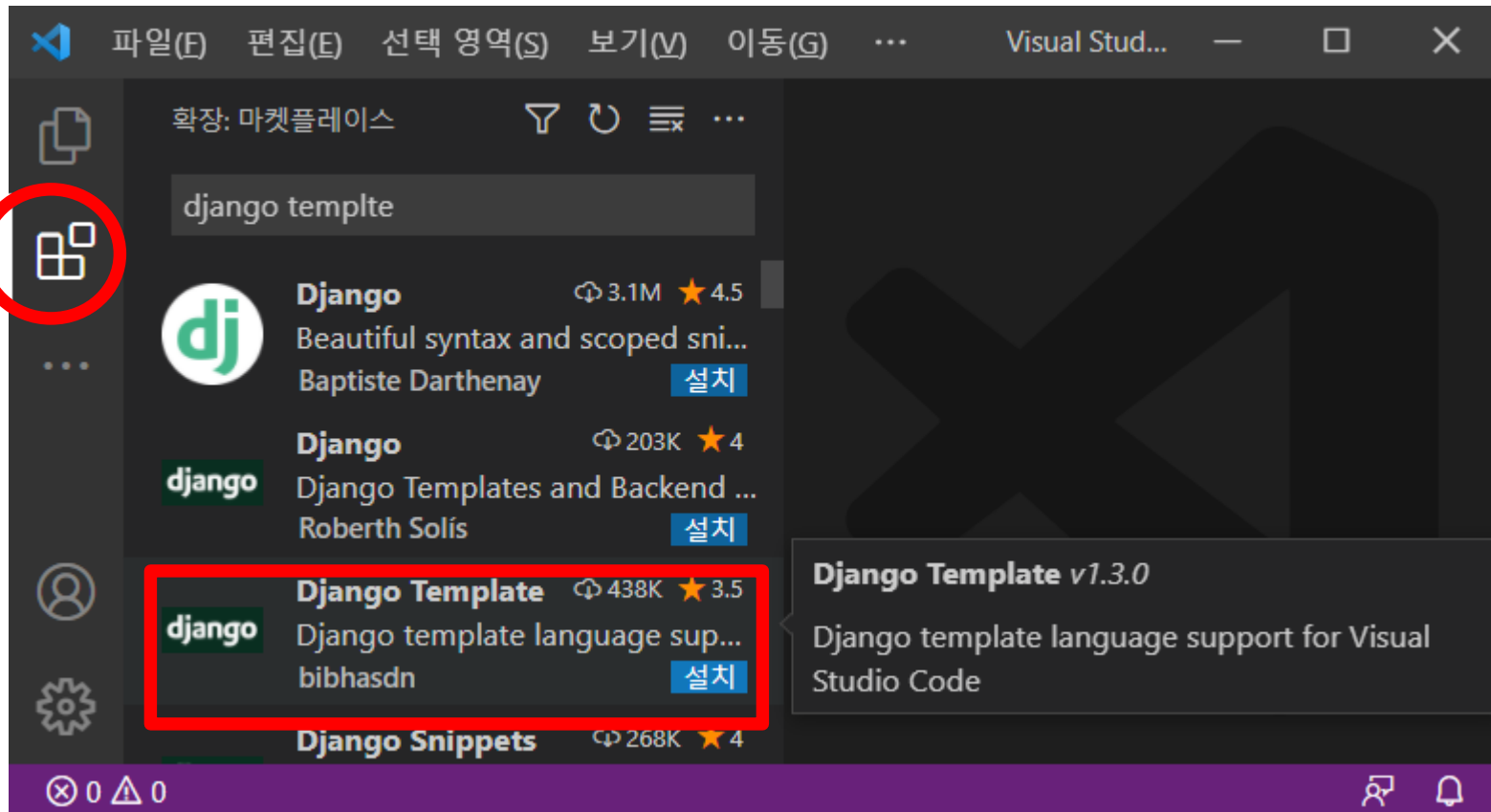
- 원하는 테마 설정
- 확장 -> python -> 설치 -> reload



Django Template

2. 개발 환경 구축

- 확장 -> django template -> 설치



- 추가하면 좋은 extenson : Korean, Prettier – code formatter, django

Django 설치

2. 개발 환경 구축

- Django 설치

```
D:\src\django>pip install django
```

최신버전

```
D:\src\django>pip install "django==3.2.5"
```

특정버전

```
D:\src\django>pip install "django~=3.2.0"
```

0끝을 최신버전

- Django 버전 확인

```
D:\src\django>django-admin --version
```

```
D:\src\django>pip show django
```

- 장고 설치된 곳 찾기

```
D:\src\django>python -c "import django; print(django.__path__)"
```

- Django 공식 소스 코드 저장소 - <http://github.com/django/django>

- Django 공식 사이트 - <https://www.djangoproject.com>



3. Django 구조



1. pip 업그레이드

- pip --version
- python -m pip install --upgrade pip
- pip --version

2. Django install

- pip install django

3. 파이참에서 프로젝트 생성후

- Django-admin startproject ch01 .

4. SECRET_KEY 숨기기(.gitignore생성)

- .env생성(.gitignore에 .env추가)
- Pip install python-decouple
- settings.py 를 수정하기
From decouple import config
SECRET_KEY = config('SECRET_KEY')

5. home app 추가하기

- python manage.py startapp home
- Settings.py에 home app 등록

- ch01/urls.py

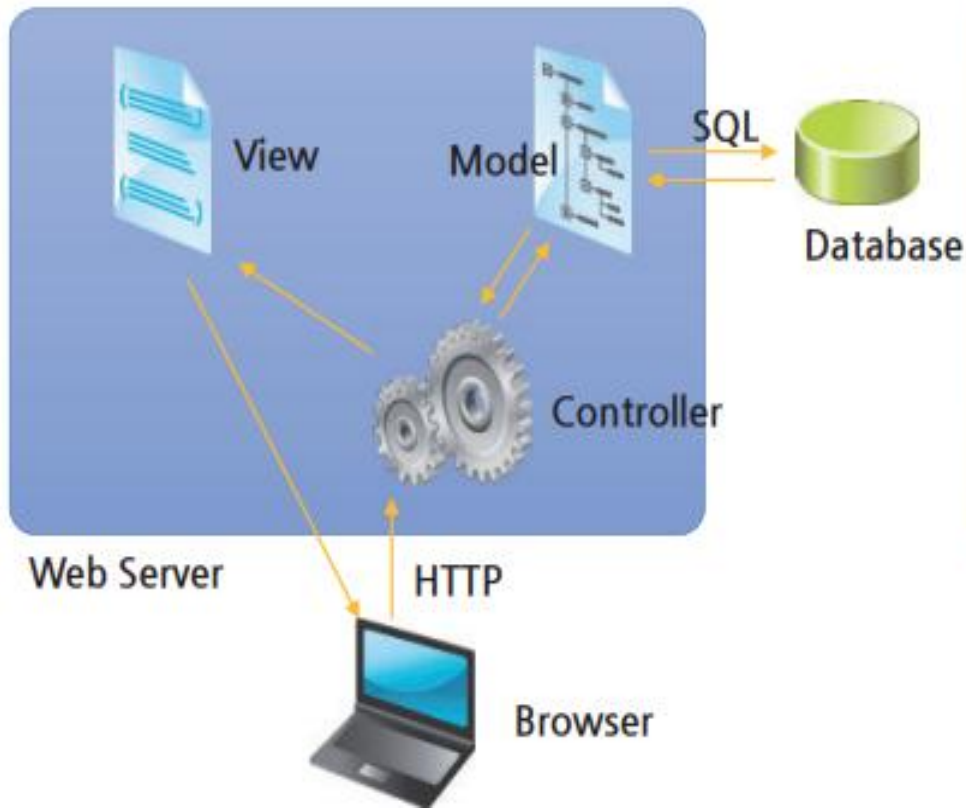
```
from django.urls import path
from hello import views
from django.urls import include
from hello import views
urlpatterns = [
    path('admin/', admin.site.urls),
    path("", views.home, name="home")
]
```

- hello/views.py

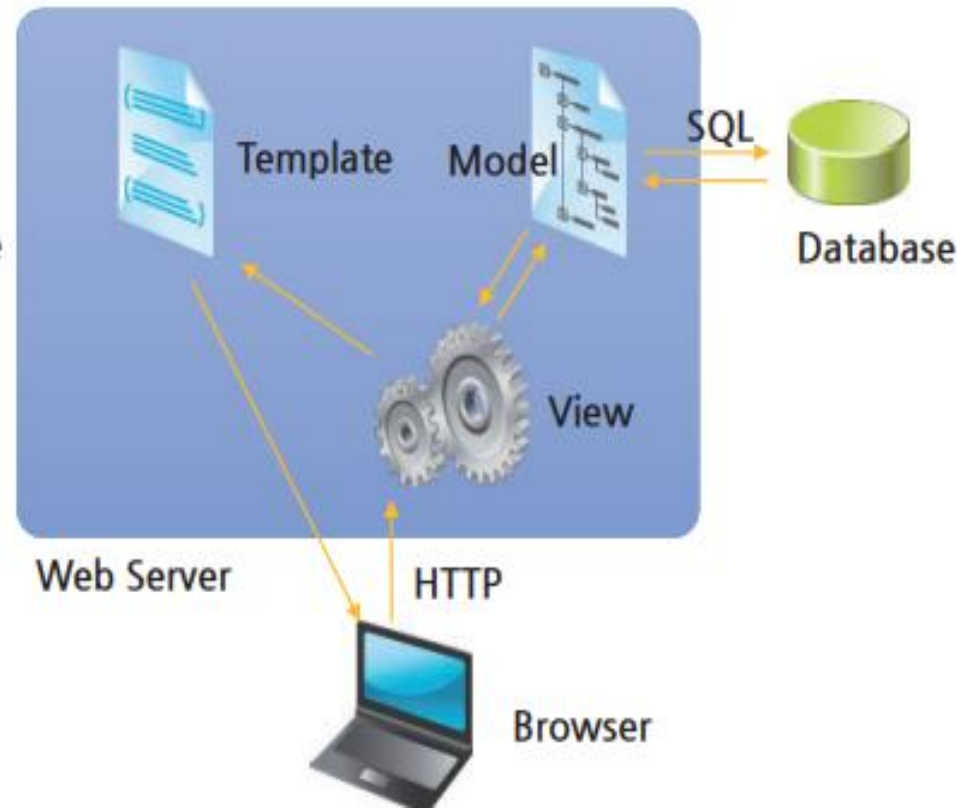
```
from django.shortcuts import render
from django.http import HttpResponse
def home(request):
    return HttpResponse('<h1>Hello, Django(장고)</h1>')
```

MVC/MVT pattern

3. Django 구조



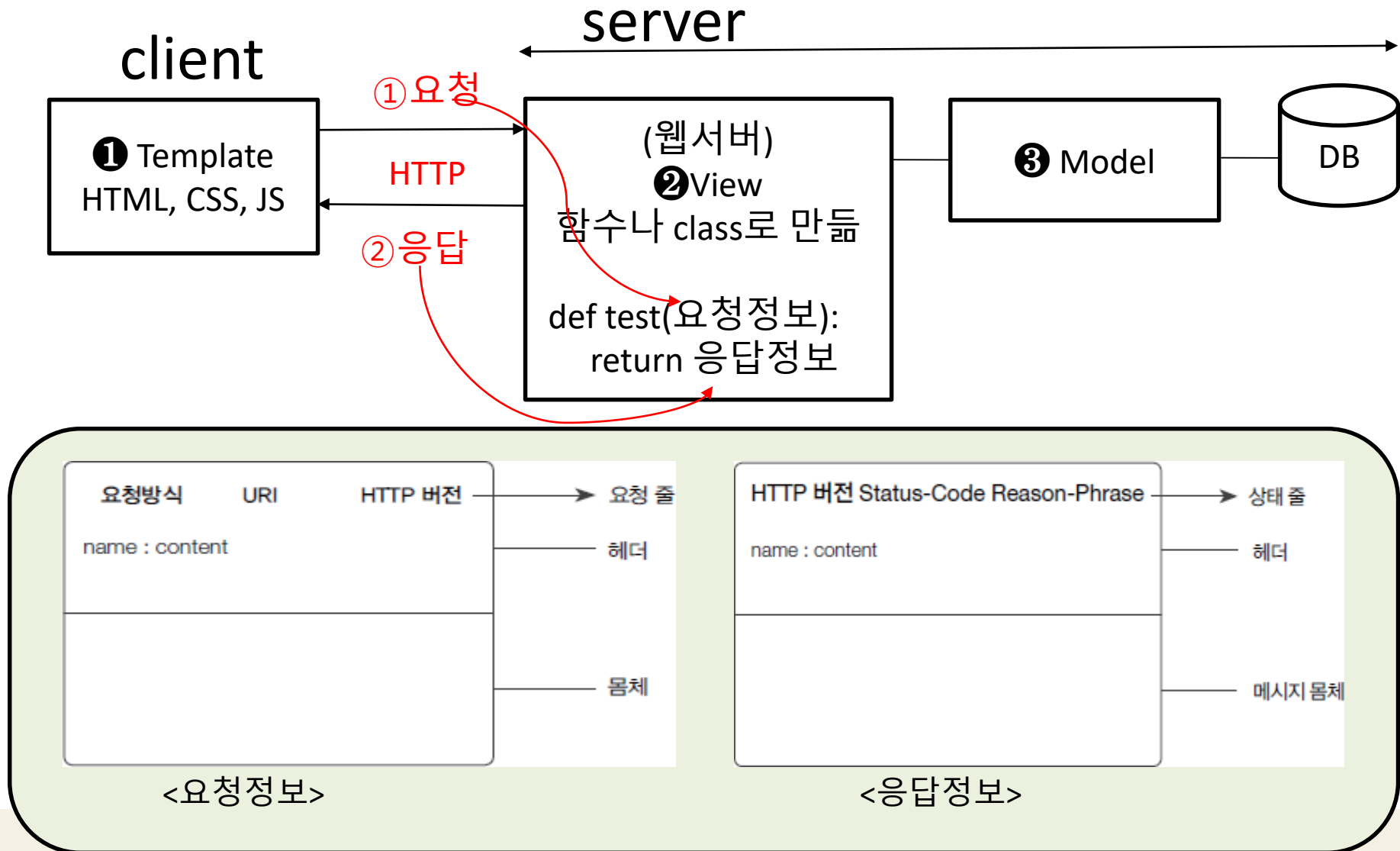
MVC (jsp, Spring)



MVT(Django)

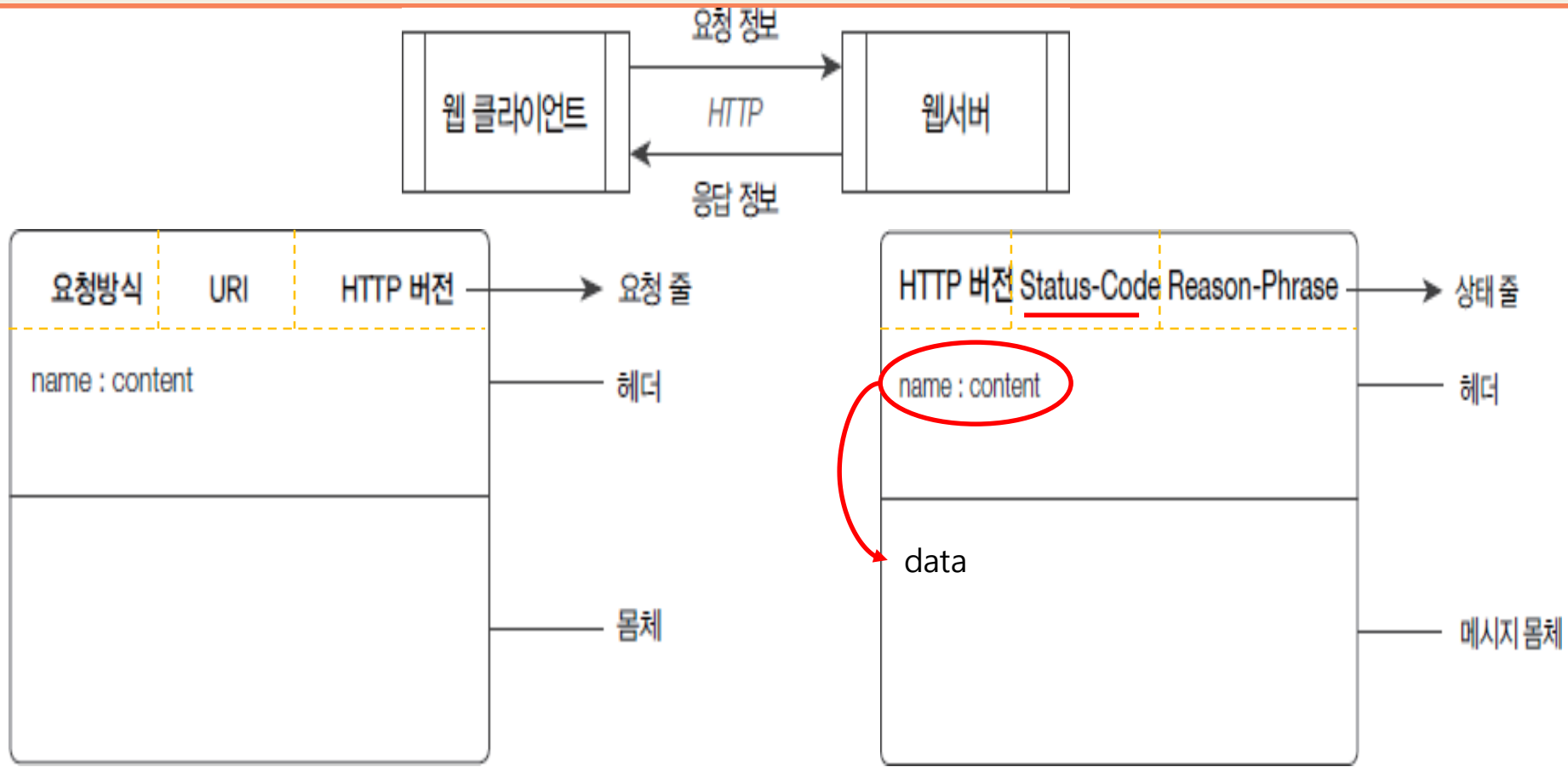
Django 구조 ; MVT 패턴

3. Django 구조



HTTP Protocol; 서버 연결 요청->클라이언트 응답->연결해제 (연결유지를 위해 쿠키 또는 세션 이용)

3. Django 구조

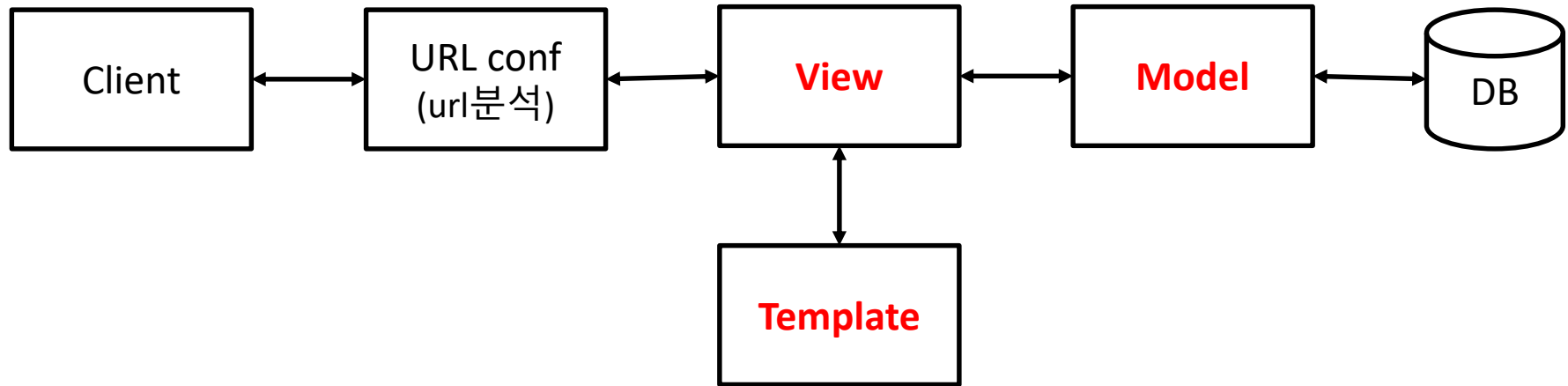


- 1. 요청방식 : GET/POST
(GET방식은 몸체없이 헤더만 넘어옴)
- 2. URI : url 뒷부분
url : <http://www.naver.com:80/show/index.html>
uri : /show/index.html

- 1. Status-code(응답코드, 상태코드)
 - 200 : success code
 - 4xx : 인증관련error
 - 500 : 서버 프로그램 error

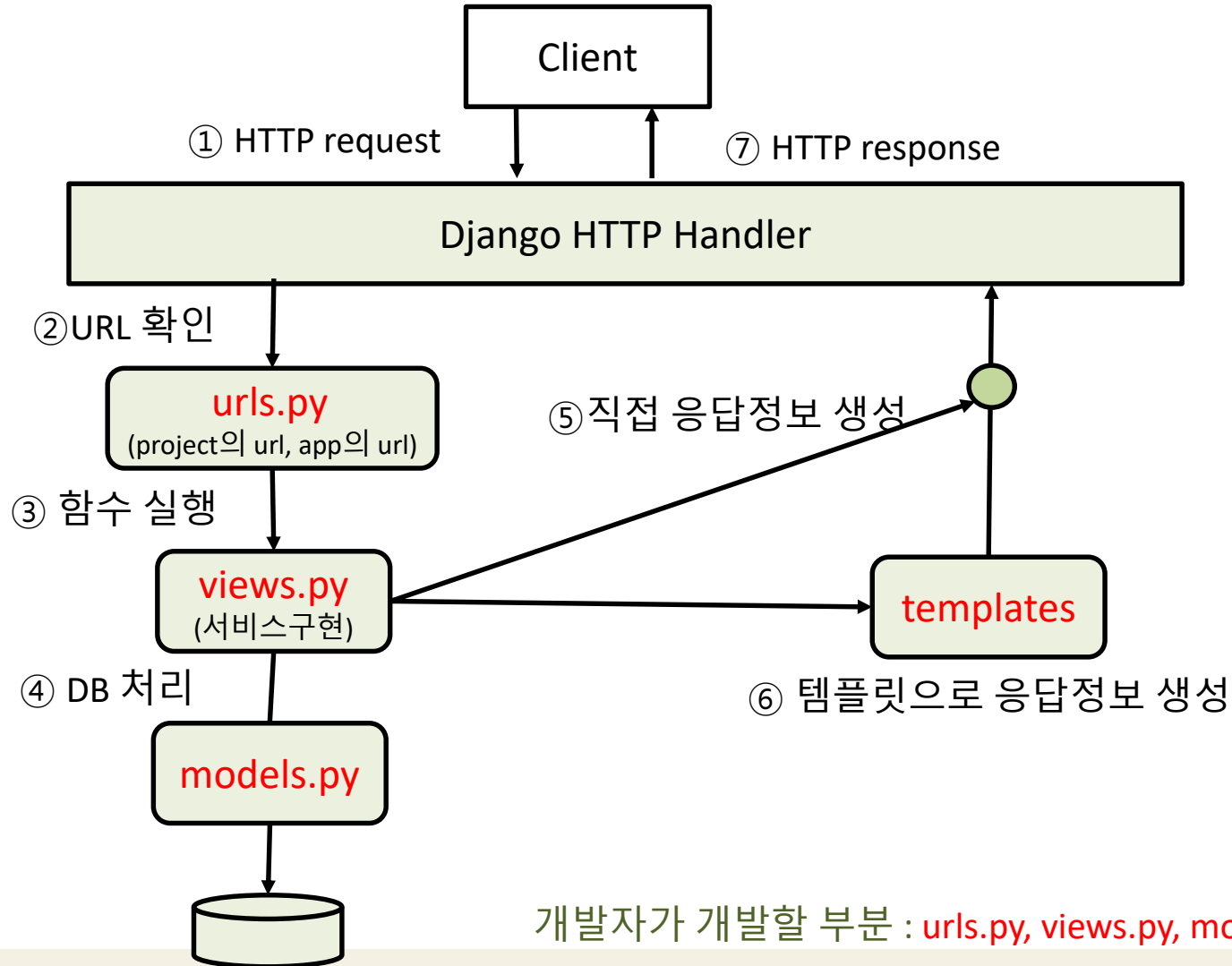
Django Process Structure

3. Django 구조



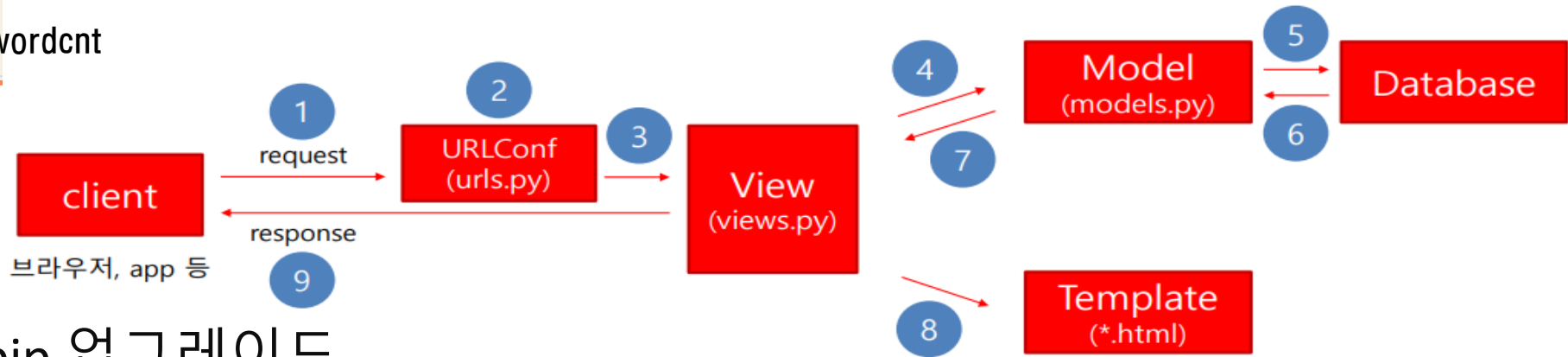
Django Process Structure

3. Django 구조



개발자가 개발할 부분 : **urls.py, views.py, models.py, templates**

Django framework flow



1. pip 업그레이드

- pip --version
- python -m pip install --upgrade pip
- pip --version

2. Django install

- pip install django

3. 파이참에서 프로젝트 생성 후

- Django-admin startproject ch02 .

4. SECRET_KEY 숨기기(.gitignore생성)

- .env생성(.gitignore에 .env추가)
- Pip install python-decouple
- settings.py 를 수정하기

From decouple import config

SECRET_KEY = config('SECRET_KEY')

5. Home, wordcnt app 추가하기

- `python manage.py startapp home`
- `python manage.py startapp wordcnt`
- `Settings.py`에 `home`과 `wordcnt` app 등록

6. URLconf

- `ch02/urls.py`

```
urlpatterns = [  
    path("admin/", admin.site.urls),  
    path("", views.index, name='index'),  
    path('test/', views.test, name='test'),  
    path('showId/<int:id>/', views.intId, name='showIntId'),  
    path('showId/<str:id>/', views.strId, name='showStrId'),  
    path('wordcnt/', include('wordcnt.urls')),  
]
```
- `Wordcnt/urls.py`

- Wordcnt/urls.py
wordcnt 패키지 안의 urls.py :
/wordcnt/ : text 입력
/wordcnt/result : 입력된 text wordcount
/wordcnt/about : 도움말 페이지
from django.urls import path
import wordcnt.views
app_name = 'wordcnt'
urlpatterns = [
 path('', wordcnt.views.wordinput, name='wordinput'),
 path('about/', wordcnt.views.about, name='about'),
 path('result/', wordcnt.views.result, name='result'),
]

7. Views

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

# Create your views here.
def index(request):
    context = {'msg': 'wordCount welcome page'}
    return render(request,
                  'home/index.html',
                  context=context)

def test(request):
    return HttpResponse('<h1>테스트 페이지</h1>'+
                       '<button onclick="location=W'/W'">뒤로</button>')

def intId(request, id):
    msg = '숫자 ID는 ' + str(id)
    type = '숫자'
    return render(request,
                  template_name='home/showId.html',
                  context={'msg': msg, 'type': type})

def strId(request, id):
    msg = '문자 ID는 ' + str(id)
    type = '문자'
    return render(request,
                  template_name='home/showId.html',
                  context={'msg': msg, 'type': type})
```


7. Views

```
from django.shortcuts import render

# text 입력
def wordinput(request):
    return render(request, 'wordcnt/wordinput.html')
def about(request):
    return render(request, 'wordcnt/about.html')

def result(request):
    #print(request.POST)
    # full = request.POST['fulltext']
    #full = request.POST.get('fulltext', '')
    full = request.GET['fulltext']
    strlength = len(full) # 글자수
    words = full.split()
    wordcnt = len(words) # 단어 갯수
    words_dic = dict() # 빈 딕셔너리
    for word in words:
        if word in words_dic.keys():
            words_dic[word] += 1 # words_dic['hong'] = 2
        else:
            words_dic[word] = 1
    print('★', full, wordcnt)
    context = {
        'full':full,
        'strlength':strlength,
        'wordcnt':wordcnt,
        'dict':words_dic.items() # [('hong',2),('good',1),('luck',1)]
    }
    return render(request,
                    'wordcnt/result.html',
                    context=context)
```

8. templates/home

```

<head>
  <meta charset="UTF-8">
  <!-- 파비콘 설정 : 아이콘 설정 (http://www.flaticon.com) -->
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css" rel="stylesheet">
  {% load static %}
  <link href="{% static 'img/brand.png' %}" rel="icon">
  <link href="{% static 'css/ex.css' %}" rel="stylesheet">
  <title>Title</title>
</head>
<body>
<div class="p-5 mb-4 bg-body-tertiary rounded-3">
  <h2>템플릿을 index.html로 사용해 보았어요</h2>
  <h1 onclick="location='{% url 'wordcnt:wordinput' %}'">{{msg}}</h1>
  <br>
  <a href="test">TEST</a><br>
  <a href="{% url 'test' %}">TEST</a><br>
  <button class="btn btn-outline-danger" onclick="location='{% url 'test' %}'">TEST</button>
  <button class="btn btn-outline-danger" onclick="location='/test'">TEST</button>
  <hr>
  <a href="showId/123">showIntId/123</a><br>
  <a href="{% url 'showIntId' 124 %}">showIntId/124</a><br>
  <button class="btn btn-outline-warning" onclick="location='showId/123'">숫자ID</button>
  <button class="btn btn-outline-warning" onclick="location='{% url 'showIntId' 124 %}'">숫자ID</button>
  <hr>
  <a href="showId/aab">showIntId/123</a><br>
  <a href="{% url 'showStrId' 'abc' %}">showIntId/124</a><br>
  <button class="btn btn-outline-info" onclick="location='showId/abc'">문자ID</button>
  <button class="btn btn-outline-info" onclick="location='{% url 'showStrId' 'abc' %}'">문자ID</button>
</div>
</body>

```