

공개SW 솔루션 설치 & 활용 가이드

시스템SW > SW공학도구



django

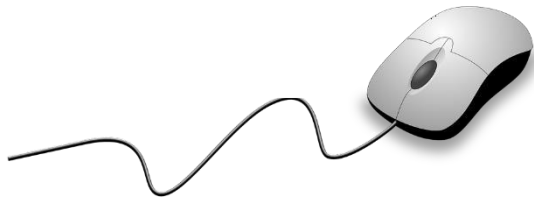
제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide



오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center



CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

1. 개요

django



소개	<ul style="list-style-type: none"> Python으로 작성된 오픈 소스 웹 프레임워크 		
주요기능	<ul style="list-style-type: none"> MVC패턴 기반의 MVT ORM(Object-Relational Mapping) 기능 지원 관리자 웹 인터페이스 제공 자체 템플릿 시스템 		
대분류	<ul style="list-style-type: none"> 시스템 SW 	소분류	<ul style="list-style-type: none"> SW공학도구
라이선스 형태	<ul style="list-style-type: none"> 3-clause BSD 	사전설치 솔루션	<ul style="list-style-type: none"> Python 3
운영체제	<ul style="list-style-type: none"> Windows, Linux, MacOS 	버전	<ul style="list-style-type: none"> 2.1.2 (2018년 10월 기준)
특징	<ul style="list-style-type: none"> django는 웹 개발에서 번거로운 요소들을 새로 개발 할 필요 없이 내장된 기능만을 이용해 빠른 개발 가능 		
보안취약점	<ul style="list-style-type: none"> 취약점 ID : CVE-2018-16984 심각도 : N/A 취약점 설명 : 2.1.2 이전 Django 2.1에서 권한 없는 사용자가 임의 계정의 암호해시를 읽을 수 있는 문제가 발견, 사용자가 전체 암호를 표시하기 위해 "view" 권한(Django 2.1의 새로운 기능)만 가진 경우 Django 관리자가 난독화된 암호 해시를 표시하기 위해 사용하는 읽기전용 암호 위젯이 무시됨 대응방안 : 2.1.2 이상으로 업그레이드 참고 경로 : https://www.djangoproject.com/weblog/2018/oct/01/security-release/ 		
개발회사/커뮤니티	<ul style="list-style-type: none"> Django Software Foundation 		
공식홈페이지	<ul style="list-style-type: none"> https://www.djangoproject.com 		





- Python으로 작성된 오픈 소스 웹 애플리케이션 프레임워크로 MVC(Model – View–Controller) 패턴을 따르고 있으며, 현재는 장고 소프트웨어 재단에 의해 관리
- 주요 목표는 복잡한 데이터베이스 기반 웹 사이트의 생성을 쉽게 함
- 컴포넌트의 재사용성과 플러그인화 가능성, 빠른 개발 강조함
- Django를 사용하는 잘 알려진 사이트는 Instagram, Mozilla, The Washington Times, Disqus 등 있음



- MVC 패턴 기반 MTV
 - django는 전통적인 MVC 디자인 패턴에서 이야기하는 컨트롤러의 기능을 프레임워크를 자체에서 하기 때문에 모델(Model), 템플릿(Template), 뷰(View)로 분류해 MTV 프레임워크라고 보기도 함
- 객체 관계 매핑(ORM; Object-Relational Mapping)
 - 다양한 데이터베이스 시스템을 지원하고 있으며, 이미 구축한 데이터베이스 시스템을 다른 데이터베이스로 변경하는 경우에도 설정을 조금만 변경하면 가능하도록 쉽고 편리함
- 프레임 워크 제한없이 원하는대로 URL을 디자인 가능
- 자체 템플릿 시스템
- Django는 양식을 HTML로 렌더링하고, 사용자가 제출한 데이터를 검증하고, 해당 데이터를 원시 Python 유형으로 변환하는 강력한 양식 라이브러리 제공
- 가장 강력한 부분 중 하나는 자동 관리 인터페이스
- 텍스트를 다른 언어로 번역하고 날짜, 시간, 숫자 및 시간대에 대한 로케일 별 형식을 완벽 지원

3. 실행환경

django



- 지원 OS
 - Windows 32bit / 64bit
 - MacOS
 - Linux
- Django는 1.11 LTS 버전이 Python 2.7을 마지막으로 지원하는 버전(2020년 4월까지 지원)
- 2.0 버전부터는 Python 3 전용 릴리스



4. 설치 및 실행

django



세부 목차

1. django 설치하기
2. 코드 에디터 설치하기



4. 설치 및 실행

django



4.1 django 설치하기

- django를 설치하려면 Python이 먼저 설치 필요
- Python이 설치되어있지 않은 경우 먼저 Python 설치가이드(<https://www.oss.kr/storage/app/public/oss/2d/81/%5bPython%5d%20Solution%20Guide%20V0.95.pdf>)를 보고 Python 설치
- 플랫폼에 따라서 django를 설치하는 방법 다름
- Windows - cmd 창에서
 - > pip install django
- Mac / Linux - 터미널 창에서
 - \$ pip install django

```
명령 프롬프트
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\#sbk>pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/32/ab/22530cc1b2114e6067eece94a333d6c749fa1c56a009f0721e51c181ea53/Django-2.1.2-py3-none-any.whl (7.3MB)
    100% |#####| 7.3MB 169kB/s
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/52/8b/876c5745f617630be90cfb8f8afe363c6d7204b176dc707d1805d1e9a0a35/pytz-2018.6-py2.py3-none-any.whl (507kB)
    100% |#####| 512kB 826kB/s
Installing collected packages: pytz, django
Successfully installed django-2.1.2 pytz-2018.6
You are using pip version 9.0.1, however version 18.1 is available.
You should consider upgrading using the 'pip install --upgrade pip' command.
```

```
[root@localhost tuser]# pip install django
Collecting django
  Downloading https://files.pythonhosted.org/packages/32/ab/22530cc1b2114e6067eece94a333d6c749fa1c56a009f0721e51c181ea53/Django-2.1.2-py3-none-any.whl (7.3MB)
    100% |#####| 7.3MB 167kB/s
Collecting pytz (from django)
  Downloading https://files.pythonhosted.org/packages/52/8b/876c5745f617630be90cfb8f8afe363c6d7204b176dc707d1805d1e9a0a35/pytz-2018.6-py2.py3-none-any.whl (507kB)
    100% |#####| 512kB 1.8MB/s
Installing collected packages: pytz, django
Successfully installed django-2.1.2 pytz-2018.6
```



4. 설치 및 실행

django



4.2 코드 에디터 설치하기(1/3)

- Atom
 - Atom은 GitHub에서 만든 무료 오픈소스
- Brackets
 - Brackets은 어도비에서 만든 무료 에디터
- Sublime Text3
 - 가장 널리 알려진 프로그램으로 무료로 사용할 수 있으며, 설치와 사용이 편리하고 모든 운영체제에서 사용 가능



4. 설치 및 실행



4.2 코드 에디터 설치하기(Mac /Linux)(2/3)

- Sublime Text 설치하기
 - \$ sudo rpm -v --import <https://download.sublimetext.com/sublimehq-rpm-pub.gpg>
 - \$ sudo yum-config-manager --add-repo https://download.sublimetext.com/rpm/stable/x86_64/sublime-textrepo
 - \$ sudo yum install sublime-text

```
[tuser@localhost ~]$ sudo rpm -v --import https://download.sublimetext.com/sublimehq-rpm-pub.gpg
[sudo] tuser의 암호:
[tuser@localhost ~]$ su
암호:
[root@localhost tuser]# sudo yum-config-manager --add-repo https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo
Loaded plugins: fastestmirror, langpacks
adding repo from: https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo
grabbing file https://download.sublimetext.com/rpm/stable/x86_64/sublime-text.repo to /etc/yum/repos.d/sublime-text.repo
repo saved to /etc/yum/repos.d/sublime-text.repo
[root@localhost tuser]# sudo yum install sublime-text
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.navercorp.com
* epel: mirror.premi.st
* extras: mirror.navercorp.com
* ius: mirrors.kernel.org
```



4. 설치 및 실행

django



4.2 코드 에디터 설치하기(Mac /Linux)(3/3)

- 위 방법으로 안될 시
 - \$ wget https://download.sublimetext.com/sublime_text_3_build_3176_x64.tar.bz2
 - \$ tar vxif sublime_text_3_build_3176_x64.tar.bz2



5. 기능소개

django



세부 목차

1. Admin 사이트
2. 파이썬 셸
3. 템플릿 시스템
4. form
5. 클래스형 뷰
6. 로그



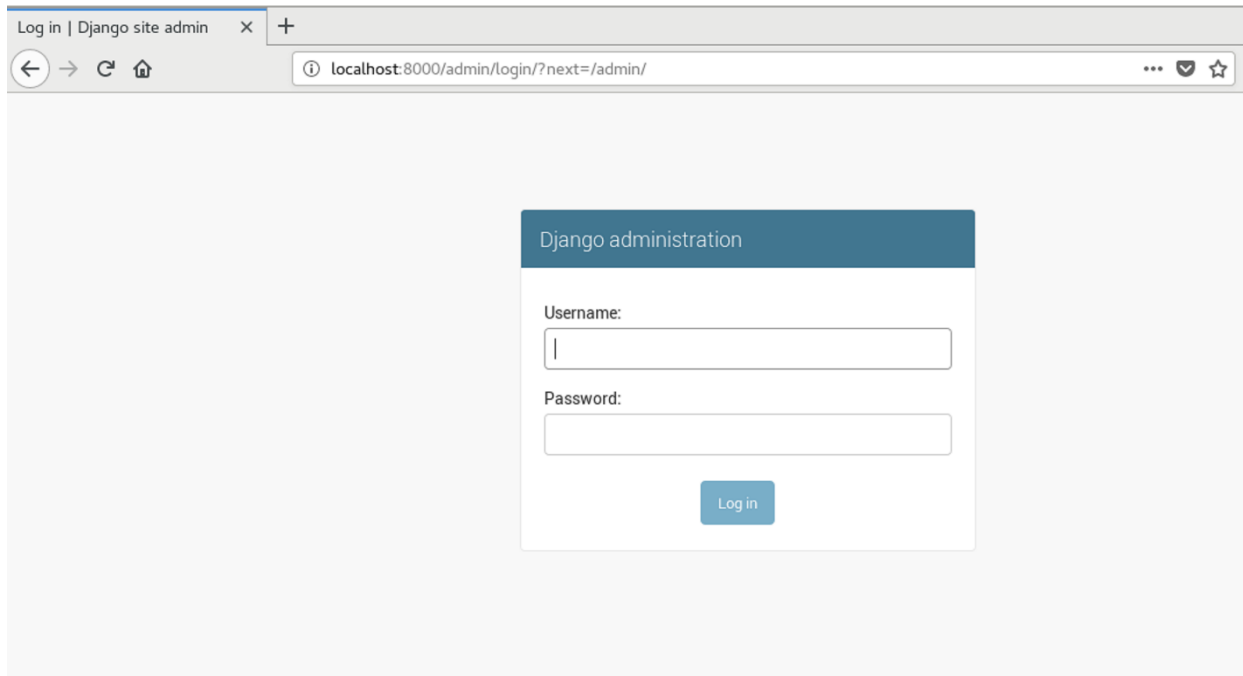
5. 기능소개

django



5.1 Admin 사이트

- 장고의 Admin 기능은 데이터 관리를 쉽게 해주고 UI도 깔끔하게 정돈된 모습을 제공
- 또한 자신의 취향에 맞게 꾸미는 것도 가능하여 많은 사람들이 장고를 사용하는 장점으로 뽑고 있음



5. 기능소개

django



5.2 파이썬 셸

- 장고는 추가로 파이썬 셸을 이용하여 데이터를 관리할 수 있는 API 제공
- 셸 데이터 처리는 인터넷 속도가 느려져 Admin 사이트에 접속이 어려울 때 사용 가능
- 복잡한 조건으로 검색하는 것처럼 Admin 사이트보다는 더 다양한 데이터 관리 명령이 가능하다는 장점 있음

```
[root@localhost mysite]# python3 manage.py shell
Python 3.6.5 (default, Apr 10 2018, 17:08:37)
[GCC 4.8.5 20150623 (Red Hat 4.8.5-16)] on linux
Type "help", "copyright", "credits" or "license" for more information.
(InteractiveConsole)
>>> █
```



5. 기능소개

django



5.3 템플릿 시스템

- MTV 모델에서 UI를 담당하고 있는 기능이 템플릿 시스템
- 템플릿 코드를 작성 시에 HTML 코드와 파이썬 코드가 섞이지만 중요한 점은 템플릿에서는 로직을 표현하는 것이 아니라 사용자에게 어떻게 보여줄지에 대한 룩앤필을 표현한다는 것이며, 따라서 템플릿 코딩은 프로그래밍 이라기 보다는 화면 구현이라고 하는 것이 더 적절한 표현
- 템플릿 코드에 if 태그, for 태그 등이 있지만, 이들은 파이썬 프로그래밍 언어의 문법과는 다른 것이며 템플릿 시스템에서만 사용되는 고유 문법
- 장고의 템플릿 시스템은 템플릿 문법으로 작성된 템플릿 코드를 해석하여 템플릿 파일로 결과물 생성
- 이렇게 템플릿 코드를 템플릿 파일로 해석하는 과정을 장고에서는 렌더링이라고 함
- 결과물인 템플릿 파일은 HTML XML CSV 등의 단순한 텍스트 파일



5. 기능소개

django



5.4 form

- 장고는 폼 처리를 위하여 다음의 3가지 기능 제공
 - 폼 생성에 필요한 데이터를 폼 클래스로 구조화하기
 - 폼 클래스의 데이터를 렌더링하여 HTML 폼 만들기
 - 사용자로부터 제출된 폼과 데이터를 수신하고 처리하기



5. 기능소개

django



5.5 클래스형 뷰

- 뷰는 요청을 받아서 응답을 반환해주는 호출 가능 객체
- 장고에서 뷰를 작성하는 방법은 두 가지가 있으며, 하나는 함수이고 하나는 클래스
- 프로젝트가 간단한 경우에는 함수형 뷰로 신속하게 개발하는 것도 하나의 방법임
- 클래스형 뷰를 사용하면 상속과 믹스인 기능을 사용하여 코드를 재사용할 수 있고 뷰를 체계적으로 구성 가능

```
from django.views.generic import ListView, DetailView
from .models import Question

class IndexView(ListView):
    template_name = 'cbvpolls/index.html'
    context_object_name = 'latest_question_list'

    def get_queryset(self):
        return Question.objects.order_by('-pub_date')[:5]

class DetailView(DetailView):
    model = Question
    template_name = 'cbvpolls/detail.html'

class ResultsView(DetailView):
    model = Question
    template_name = 'cbvpolls/results.html'
```

```
from django.shortcuts import render, get_object_or_404
from django.core.urlresolvers import reverse
from .models import Question

def index(request):
    latest_question_list = Question.objects.all().order_by('-pub_date')[:5]
    context = {'latest_question_list': latest_question_list}
    return render(request, 'polls/index.html', context)

def detail(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/detail.html', {'question': question})

def results(request, question_id):
    question = get_object_or_404(Question, pk=question_id)
    return render(request, 'polls/results.html', {'question': question})
```



5. 기능소개

django



5.6 로그

- 장고의 로깅은 기본적으로 파이썬의 로깅 체계를 따르면서 일부 추가
- 파이썬의 로깅 모듈을 보면 로거, 핸들러, 필터, 포맷터 4가지 주요 모듈을 정의하고 있음
- 장고의 runserver나 웹 서버에 의해 장고가 실행될 때 장고는 settings.py 파일에 정의된 LOGGING_CONFIG, LOGGING 항목을 참고하여 로깅에 관련된 설정 처리
- settings.py 파일에 관련 항목이 없더라도 디폴트 로깅 설정 처리
- 따라서 장고의 로깅은 실행되는 시점부터 준비되어 있으므로, 항상 로그를 기록하는 것이 가능 상태임



6. 활용예제

django



세부 목차

1. django 프로젝트 만들기
 1. django 프로젝트 생성
 2. django 서버 실행
 3. django 서버 접속
 4. django 서버 중단하기
2. Hello World 출력하기
 1. app 만들기
 2. hello world를 출력하는 index 함수 만들기
 3. 앱에 접근할 조건을 지정하는 함수 만들기
 4. 앞서 생성한 index 함수를 실행할 조건을 지정하는 함수 만들기
 5. Hello World 확인하기



6. 활용예제

django



6.1.1 django 프로젝트 생성(Windows)(1/2)

- > `django-admin startproject <프로젝트이름>`
- > `dir` 로 확인해보면 `mysite` 라는 폴더가 생긴 것을 확인 가능
- > `cd mysite` 디렉토리를 이동하여 `dir`로 확인해보면 `manage.py` 파일과 `mysite` 라는 폴더가 생성된 것을 확인 가능

```
ca. 명령 프롬프트
D:\#test>django-admin startproject mysite
D:\#test>dir
D 드라이브의 볼륨: 1TB
볼륨 일련 번호: 86F5-AD56

D:\#test 디렉터리

2018-10-25 오후 09:15 <DIR>          .
2018-10-25 오후 09:15 <DIR>          ..
2018-10-25 오후 09:15 <DIR>          mysite
                0개 파일              0 바이트
                3개 디렉터리  380,228,243,456 바이트 남음

D:\#test>cd mysite
D:\#test\mysite>dir
D 드라이브의 볼륨: 1TB
볼륨 일련 번호: 86F5-AD56

D:\#test\mysite 디렉터리

2018-10-25 오후 09:15 <DIR>          .
2018-10-25 오후 09:15 <DIR>          ..
2018-10-25 오후 09:15                553 manage.py
2018-10-25 오후 09:15 <DIR>          mysite
                1개 파일              553 바이트
                3개 디렉터리  380,228,243,456 바이트 남음
```



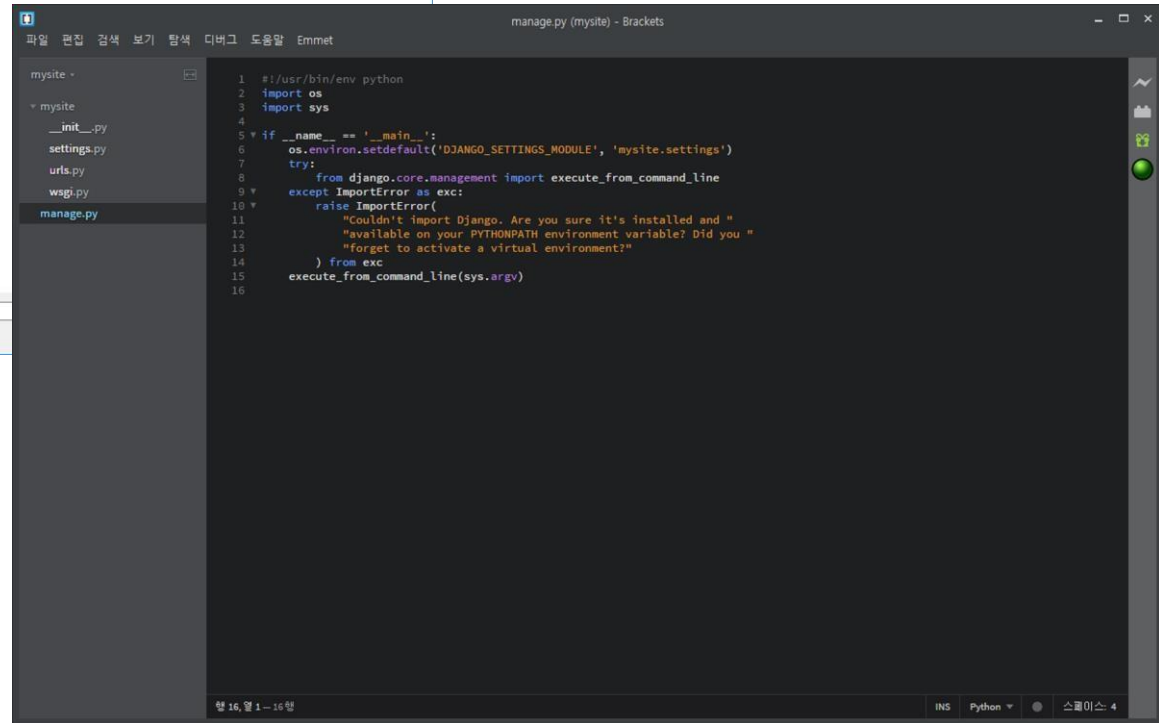
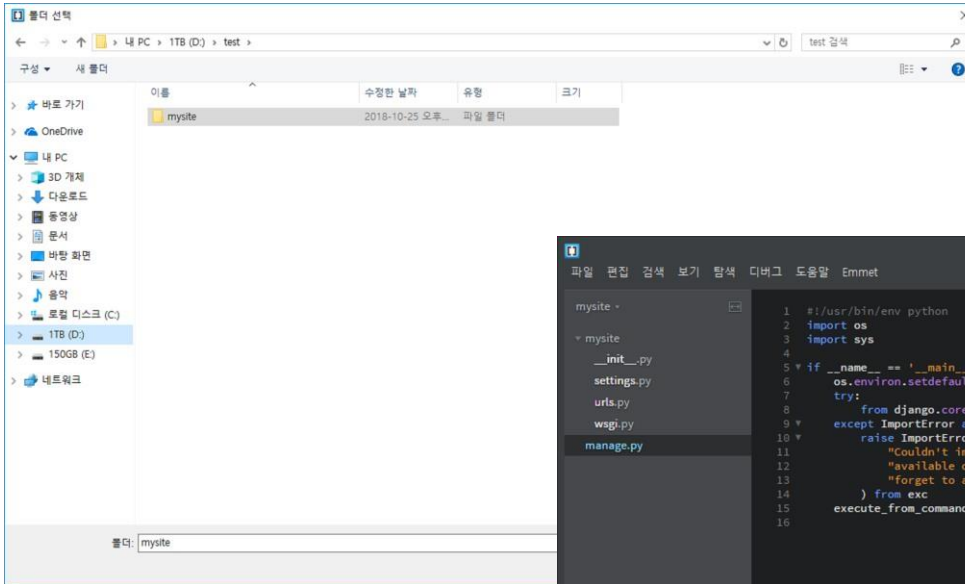
6. 활용예제

django



6.1.1 django 프로젝트 생성(Windows)(2/2)

- 에디터에서 파일 - 폴더열기 - mysite 를 하면 파일들의 구조 확인 가능



6. 활용예제

django



6.1.2 django 서버 실행(Windows)

- > cd mysite 로 디렉토리 이동
- > python manage.py runserver를 입력하면 서버가 시작되는 것을 확인 가능

```
cmd 명령 프롬프트 - python manage.py runserver
3개 디렉터리 380,228,243,456 바이트 남음

D:\test>cd mysite

D:\test\mysite>dir
D 드라이브의 볼륨: 1TB
볼륨 일련 번호: 86F5-AD56

D:\test\mysite 디렉터리

2018-10-25 오후 09:15 <DIR>          .
2018-10-25 오후 09:15 <DIR>          ..
2018-10-25 오후 09:15                553 manage.py
2018-10-25 오후 09:15 <DIR>          mysite
                1개 파일                553 바이트
                3개 디렉터리 380,228,243,456 바이트 남음

D:\test\mysite>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 25, 2018 - 21:54:35
Django version 2.1.2, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```



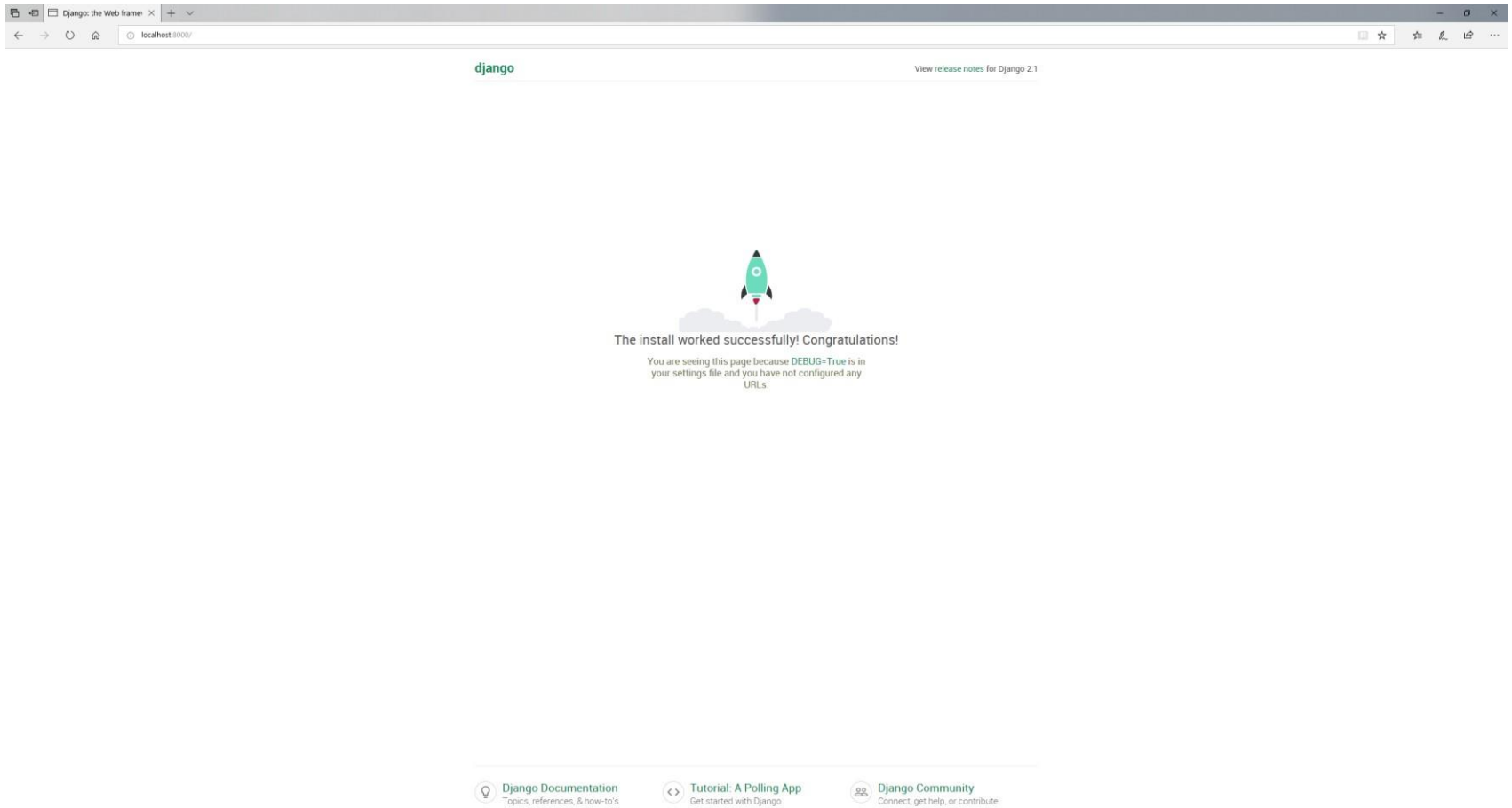
6. 활용예제

django



6.1.3 django 서버 접속(Windows)

- 주소창에 127.0.0.1:8000 또는 localhost:8000 입력
- 화면이 나오면 성공



6. 활용예제



6.1.4 django 서버 중단하기(Windows)

- Cmd에서 접속했던 로그 확인 가능
- 이 상태에서는 서버가 계속 구동 되어있기 때문에 다른 작업 불가
- 서버를 다시 시작하고 싶거나 다른 작업을 계속하고 싶으면 실행중인 서버를 종료 해야함
- 이 때 Ctrl + c 키를 누르면 됨

```
명령 프롬프트 - python manage.py runserver
d:\test\mysite>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 15 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin,
auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
October 25, 2018 - 22:27:38
Django version 2.1.2, using settings 'mysite.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
[25/Oct/2018 22:27:55] "GET / HTTP/1.1" 200 16348
[25/Oct/2018 22:27:55] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 304 0
[25/Oct/2018 22:27:55] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 304 0
[25/Oct/2018 22:27:56] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 304 0
[25/Oct/2018 22:29:08] "GET / HTTP/1.1" 200 16348
[25/Oct/2018 22:29:08] "GET /static/admin/css/fonts.css HTTP/1.1" 200 423
[25/Oct/2018 22:29:08] "GET /static/admin/fonts/Roboto-Regular-webfont.woff HTTP/1.1" 200 80304
[25/Oct/2018 22:29:08] "GET /static/admin/fonts/Roboto-Bold-webfont.woff HTTP/1.1" 200 82564
[25/Oct/2018 22:29:08] "GET /static/admin/fonts/Roboto-Light-webfont.woff HTTP/1.1" 200 81348
```



6. 활용예제

django



6.2.1 app 만들기

- cmd에서 > python manage.py startapp <앱이름> (Windows)
- 터미널에서 \$ python3 manage.py startapp <앱이름> (Mac / Linux)
- <앱이름> 이라는 앱을 만들라는 명령임
- Hi 라는 폴더가 mysite 라는 프로젝트 안에 생성됨
- 프로젝트명과 같은 이름을 가진 폴더는 프로젝트의 기본 설정을 담고 있음

```
명령 프롬프트
d:\test\mysite>python manage.py startapp hi
d:\test\mysite>
```

```
mysite/settings.py (mysite) - Brackets
현재 파일
settings.py
mysite
  hi
  migrations
  __init__.py
  admin.py
  apps.py
  models.py
  tests.py
  views.py
mysite
  __pycache__
  __init__.py
  settings.py
  urls.py
  wsgi.py
  db.sqlite3
  manage.py
1 """
2 Django settings for mysite project.
3
4 Generated by 'django-admin startproject' using Django 2.1.2.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/2.1/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/2.1/ref/settings/
11 """
12
13 import os
14
15 # Build paths inside the project like this: os.path.join(BASE_DIR, ...)
16 BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
17
18
19 # Quick-start development settings - unsuitable for production
20 # See https://docs.djangoproject.com/en/2.1/howto/deployment/checklist/
21
22 # SECURITY WARNING: keep the secret key used in production secret!
23 SECRET_KEY = '61dh3x%lh6j48g$+xpq&4dn(9(y%&d+l1m2c5mnc8+dcfo)9'
24
25 # SECURITY WARNING: don't run with debug turned on in production!
26 DEBUG = True
27
28 ALLOWED_HOSTS = []
29
30
31 # Application definition
32
33 INSTALLED_APPS = [
34     'django.contrib.admin',
35     'django.contrib.auth',
36     'django.contrib.contenttypes',
37     'django.contrib.sessions',
38     'django.contrib.messages',
39     'django.contrib.staticfiles',
40 ]
41
42 MIDDLEWARE = [
43     'django.middleware.security.SecurityMiddleware',
44     'django.contrib.sessions.middleware.SessionMiddleware',
```



6. 활용예제

django



6.2.2 Hello World를 출력하는 index 함수 만들기

- hi – views.py 로 이동한뒤 from django.http import HttpResponse 추가
 - def index(request):
 - return HttpResponse("Hello World")
- index 함수 추가

```
1 from django.shortcuts import render
2 from django.http import HttpResponse
3
4 # Create your views here.
5 def index(request):
6     return HttpResponse("Hello World")
```

The screenshot shows a code editor window titled 'hi/views.py (mysite) - Brackets'. The left sidebar shows a file explorer with 'views.py' selected. The main editor area contains the Python code for the 'index' function, which is highlighted with a red box. The status bar at the bottom indicates '행 6, 열 39 - 6 행', 'INS', 'UTF-8', 'Python', and '스페이스: 4'.



6. 활용예제

django



6.2.3 앱에 접근할 조건을 지정하는 함수 만들기

- mysite – mysite - urls.py 이동
- path 뒤에 include 추가
- path("", include('hi.urls')) 추가

```
1 """mysite URL Configuration
2
3 The `urlpatterns` list routes URLs to views. For more information please see:
4     https://docs.djangoproject.com/en/2.1/topics/http/urls/
5 Examples:
6 ▼ Function views
7     1. Add an import: from my_app import views
8     2. Add a URL to urlpatterns: path('', views.home, name='home')
9 ▼ Class-based views
10    1. Add an import: from other_app.views import Home
11    2. Add a URL to urlpatterns: path('', Home.as_view(), name='home')
12 ▼ Including another URLconf
13    1. Import the include() function: from django.urls import include, path
14    2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
15 """
16 from django.contrib import admin
17 from django.urls import path, include
18
19 urlpatterns = [
20     path('', include('hi.urls')),
21     path('admin/', admin.site.urls),
22 ]
23 ]
```



6. 활용예제

django



6.2.4 앞서 생성한 index 함수를 실행할 조건을 지정하는 함수만들기

- hi 안에 urls.py 라는 파일 생성
 - From Django.urls import path
 - From . Import views
 - urlpatterns = [path(' ' , views.index),]
- 내용 추가

```
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index),
6 ]
```



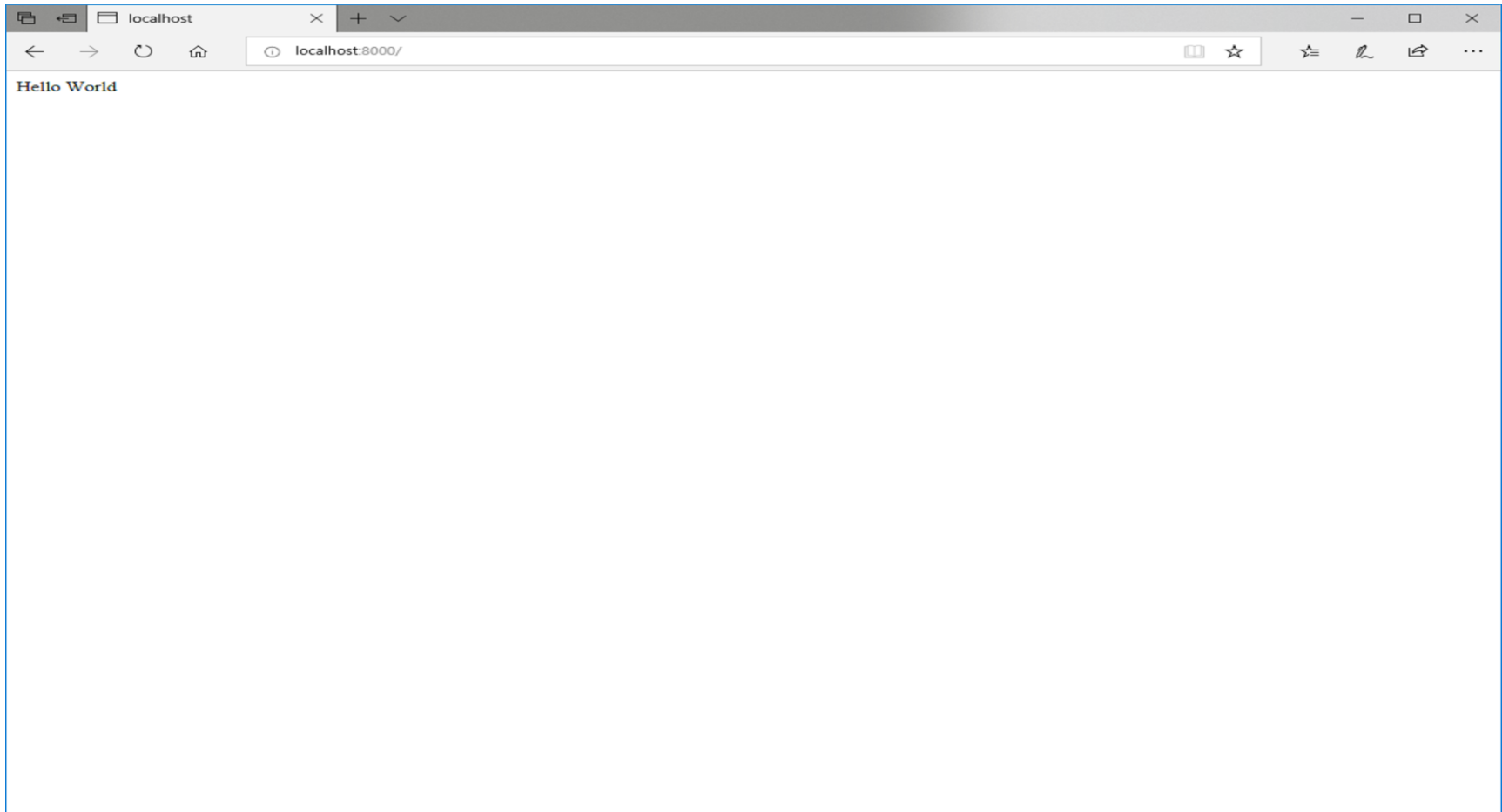
6. 활용예제

django



6.2.5 Hello World 확인하기

- localhost:8000 에 접속하면 Hello World 확인 가능





Q 왜 CentOS 에서는 `$ python3 manage.py runserver python3` 라고 하나요?

A CentOS 를 기본적으로 설치하면 python2 가 설치되어 있습니다. Python3를 설치하여도 기본설정은 python2로 되어있기때문에 숫자를 붙여주는것입니다.

Q 그렇다면 기본 설정을 python3 로 바꾸면 안 되나요?

A 그렇게 하였을 경우 yum 실행 시 에러가 발생하므로 다시 되돌려야하는 경우가 생깁니다.



8. 용어정리

django



용어	설명
MVC	모델-뷰-컨트롤러(Model-View-Controller, MVC)는 소프트웨어 공학에서 사용되는 소프트웨어 디자인 패턴
ORM	객체 관계 매핑(Object-relational mapping; ORM)은 데이터베이스와 객체 지향 프로그래밍 언어 간의 호환되지 않는 데이터를 변환하는 프로그래밍 기법



Open Source Software Installation & Application Guide



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.