

Word Counter

2021-04-24. Django 실습 session

오늘 배울 내용

1 : TV 패턴 복습

2 : Word Counter

1 : TV 패턴 복습

중간고사 기간 동안 날라간 장고 지식 되돌리기!!



(M)TV 패턴..을 기억하시나요..? 😂

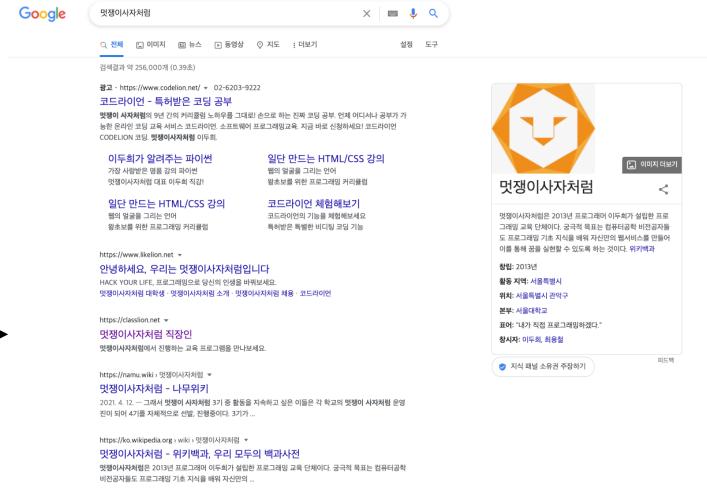


안녕, 구글 서버야!
난 '멋쟁이사자처럼' 을 검색하고 싶어!

index.html
templates

views

멋쟁이사자처럼



result.html
templates



templates

The screenshot shows the PyCharm IDE interface with two code editors open. The left editor contains the file `index.html` and the right editor contains the file `result.html`. Both files are located in the `templates` directory of a Django project named `django_word_counter`.

index.html Content:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>Document</title>
        <style>
            body {
                text-align: center;
            }
        </style>
    </head>

    <body>
        <h1>
            워드 카운터
        </h1>
        <hr/>
        <form action="{% url 'result' %}" method="POST">
            {% csrf_token %}
            <textarea name="sentence" rows="30" cols="60">
            <input type="submit"></input>
        </form>
    </body>
</html>
```

result.html Content:

```
<!DOCTYPE html>
<html lang="en">
    <head>
        <meta charset="UTF-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <title>결과</title>
        <style>
            body {
                text-align: center;
            }
        </style>
    </head>

    <body>
        <h1>
            결과
        </h1>
        <hr/>
        {% for word, count in word_counts %}
            {{word}} : {{count}} <br>
        {% endfor %}
    </body>
</html>
```

At the bottom of the interface, there is a status bar displaying the following information: Python 3.9.2 64-bit ('venv': venv) 0 △ 0 ⚡ Ln 25, Col 1 Spaces: 2 UTF-8 LF Django Template Go Live ⚡ ⚡



views.py

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following details:

- File Explorer (Left):** Shows the project structure under "OPEN EDITORS" and "DJANGO_WORD_COUNTER". The "views.py" file is currently selected.
- Code Editor (Center):** Displays the content of the "views.py" file. The code implements two views: "index" and "result". The "index" view returns an index.html template. The "result" view handles a POST request for a sentence, splits it into words, counts the occurrences of each word, sorts the results by frequency (descending), and renders a "result.html" template with the word counts.
- Status Bar (Bottom):** Shows the Python version ("Python 3.9.2 64-bit ('venv': venv)"), code analysis status ("0 △ 0"), and other settings like "Spaces: 4", "UTF-8", "LF", "Python", "Go Live", and file icons.

```
views.py — django_word_counter

word_counter > views.py > ...
1  from django.shortcuts import render
2
3  # Create your views here.
4  def index(request):
5      return render(request, 'index.html')
6
7  def result(request):
8      sentence = str(request.POST.get('sentence'))
9      words = sentence.split()
10
11     word_counts = {}
12     for word in words:
13         if word in word_counts:
14             word_counts[word] += 1
15         else:
16             word_counts[word] = 1
17     word_counts = list(word_counts.items())
18
19     word_counts.sort(key = lambda t : -t[1])
20
21
22     return render(request, 'result.html', { 'word_counts': word_counts})
```



urls.py

```
word_counter > urls.py > ...
1 from django.urls import path
2 from . import views
3
4 urlpatterns = [
5     path('', views.index, name="index"),
6     path('result', views.result, name="result")
7 ]
8 |
```

The screenshot shows the VS Code interface with the following details:

- Explorer View:** Shows the project structure under "DJANGO_WORD_COUNTER". The "urls.py" file is currently selected.
- Code Editor:** Displays the content of the "urls.py" file, which defines two URL patterns: an index page and a result page.
- Status Bar:** Shows the Python version (Python 3.9.2 64-bit), the current file (urls.py), and other settings like spaces and encoding.
- Bottom Right:** Includes status icons for line numbers (Ln 8, Col 1), spaces (Spaces: 4), encoding (UTF-8 LF), and file type (Python).



settings.py

The screenshot shows a Microsoft Visual Studio Code (VS Code) interface. The title bar reads "settings.py — django_word_counter". The left sidebar (Explorer) shows the project structure:

- OPEN EDITORS: settings.py (django_word_counter)
- DJANGO_WORD_COUNTER:
 - django_word_counter
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py (selected)
 - urls.py
 - wsgi.py
 - templates
 - venv
 - word_counter
 - db.sqlite3
 - manage.py

The main editor area displays the contents of settings.py:1 """
2 Django settings for django_word_counter project.
3
4 Generated by 'django-admin startproject' using Django 3.1.7.
5
6 For more information on this file, see
7 <https://docs.djangoproject.com/en/3.1/topics/settings/>
8
9 For the full list of settings and their values, see
10 <https://docs.djangoproject.com/en/3.1/ref/settings/>
11 """
12
13 from pathlib import Path
14 import os
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
18
19
20 # Quick-start development settings - unsuitable for production
21 # See <https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/>
22
23 # SECURITY WARNING: keep the secret key used in production secret!
24 SECRET_KEY = '=o4k_t=%#brp#\$7_b-i!by*ip15\$_!m5q)!36by#08t#brn6'
25
26 # SECURITY WARNING: don't run with debug turned on in production!
27 DEBUG = True

The status bar at the bottom indicates "Python 3.9.2 64-bit ('venv': venv)" and "Ln 2, Col 49 Spaces: 4 UTF-8 LF Python".

2 : Word counter

단어 개수를 세주는 간단한 사이트를 만들어보자!



Word counter?

워드 카운터

```
Yeah, ha  
  
모두 그에게 말해 또 웃네 (새꺄 whut?)  
죽지 않고 웠다 이렇게 (새꺄 뭐?)  
Snacky chan의 라인을 빌릴래  
했어 예수처럼 이렇게 부활을  
031 팔 안쪽의 tatt freaky woah  
어전 시장님이 만나재 motown  
어수룩해 엄마 속에 걱정  
한 톤만큼 쌓고 스물한살에 독립했던 얘는  
여섯이 되었고  
발견했지 우연히 5년 전의 노트  
“정말 스타 되고 싶어  
그럴려면 가서 만나면 돼 악마?”  
노트를 뒤고  
거울에 비춰진 남자를 보니 와  
저 손목이 너무 부러워  
저 금 daydate rollie 님을 걸 천백은 더  
  
전국  
사람들이 외치네  
저 괴물체는 뭘까?  
  
Meteor  
거대 Meteor
```

제출

이 문장에서
단어들을 뽑아서
개수를 세줘!

결과

```
Yeah : 6  
ha : 6  
저 : 5  
Meteor : 4  
처박힐 : 4  
Meteor야 : 4  
내리오자 : 4  
경고 : 4  
영 : 4  
난 : 3  
그 : 3  
모두 : 2  
새꺄 : 2  
웠다 : 2  
이렇게 : 2  
전국 : 2  
사람들이 : 2  
외치네 : 2  
괴물체는 : 2  
뭘까? : 2  
기대 : 2  
네개 : 2  
별빛을 : 2  
미구 : 2  
그래도 : 2  
the : 2  
up : 2  
별이 : 2  
영원히 : 2  
제발 : 2  
그에게 : 1  
말해 : 1  
또 : 1  
와 : 1  
whut?: 1  
죽자? : 1  
않고 : 1  
뭐?: 1  
Snacky : 1  
chan의 : 1  
라인을 : 1  
빌릴래 : 1  
쳤어? : 1  
예수처럼 : 1  
부활을 : 1  
031 : 1  
팔 : 1  
안쪽의 : 1  
tatt : 1  
freaky : 1  
woah : 1  
어전 : 1  
시장님이 : 1
```



python: dictionary를 기억하시나요..?

dictionary -> 사전

key, value 관계를 나타내주는 자료구조입니다!

```
1 koa = {  
2     'name' : '김신건',  
3     'nickname' : 'koalala',  
4     'age': 22,  
5     'blood type': 'AB',  
6     'MBTI': 'ESTJ',  
7     'org': 'grepp',  
8 }
```

Q1. 아래 코드의 실행 결과는?

```
print(koa['name'])
```

Q2. 아래 코드의 실행 결과는?

```
if 'nickname' in koa:  
    print(koa['nickname'])  
if 'place' in koa:  
    print(koa['place'])  
else:  
    print('no place')
```



python: split

```
split.py
Users > shinkeonkim > split.py > ...
1 text = "안녕 나는 멋사
2 아기사자야!
3 "
4 print(text.split())
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
1: Code + ⌂ ⌄ ⌈ ⌉ ×
python3 -u "/Users/shinkeonkim/split.py"
~ python3 -u "/Users/shinkeonkim/split.py"
['안녕', '나는', '멋사', '아기사자야!']
~ |
```

Python 2.7.16 64-bit ⌂ 0 △ 0 Ln 4, Col 20 Spaces: 4 UTF-8 LF Python ⌂ ⌙ ⌚

split은 문자열을
쪼개서
리스트로 만들어주는
메서드입니다!



form 태그

```
index.html - django_word_counter
index.html M X
templates > index.html
13 | </head>
14 |
15 | <body>
16 |   <h1>
17 |     워드 카운터
18 |   </h1>
19 |   <hr/>
20 |   <form action="{% url 'result' %}" method="POST">
21 |     {% csrf_token %}
22 |     <textarea name="sentence" rows="30" cols="60"></textarea><br>
23 |     <input type="submit"></input>
24 |   </form>
25 |
26 | </body>
27 | </html>
```

form 태그는 여러분들 모르게
정말 많은 웹사이트에서 접했습니다.

어떤 입력을 받아서 한번에 서버로 전송하는 페이지(ex. 회원가입, 로그인, 글작성)에서는
form 태그가 무조건 쓰이기 때문입니다.



form 태그

```
form.html — django_word_counter
...
templates > form.html > form
1 <form action="/login" name="login_form" accept-charset="utf-8" target="_blank" method="GET">
2 </form>

...
main* Python 3.9.2 64-bit ('venv': venv) ① 0 △ 0 Git Graph
Ln 2, Col 8 Spaces: 4 UTF-8 LF HTML Go Live
```

form 태그에서 사용할 수 있는 속성의 종류는 위와 같아요.

- **action:** 폼을 전송할 서버 주소(서버 파일)를 지정합니다.
- **name:** 폼의 이름을 지정합니다.
- **accept-charset:** 폼 전송에 사용할 문자 인코딩을 지정합니다.
- **target:** action에서 지정한 주소를 어떤 위치에서 열지를 지정합니다.
- **method:** 폼을 전송할 http 메소드를 정합니다. (주로 GET, POST를 사용합니다.)



HTTP method

HTTP method는 클라이언트와 서버 사이에 이루어지는 요청(request)과 응답(response) 데이터를 전송하는 방식을 말합니다.

단순하게 생각하면, 서버에게 요청을 보내는 방식입니다.

HTTP 메소드	RFC	요청에 Body가 있음	응답에 Body가 있음	안전	멱등(Idempotent)	캐시 가능
GET	RFC 7231	아니오	예	예	예	예
HEAD	RFC 7231	아니오	아니오	예	예	예
POST	RFC 7231	예	예	아니오	아니오	예
PUT	RFC 7231	예	예	아니오	예	아니오
DELETE	RFC 7231	아니오	예	아니오	예	아니오
CONNECT	RFC 7231	예	예	아니오	아니오	아니오
OPTIONS	RFC 7231	선택 사항	예	예	예	아니오
TRACE	RFC 7231	아니오	예	예	예	아니오
PATCH	RFC 5789	예	예	아니오	아니오	예

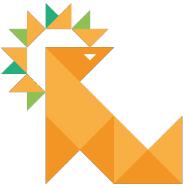


HTTP method : get

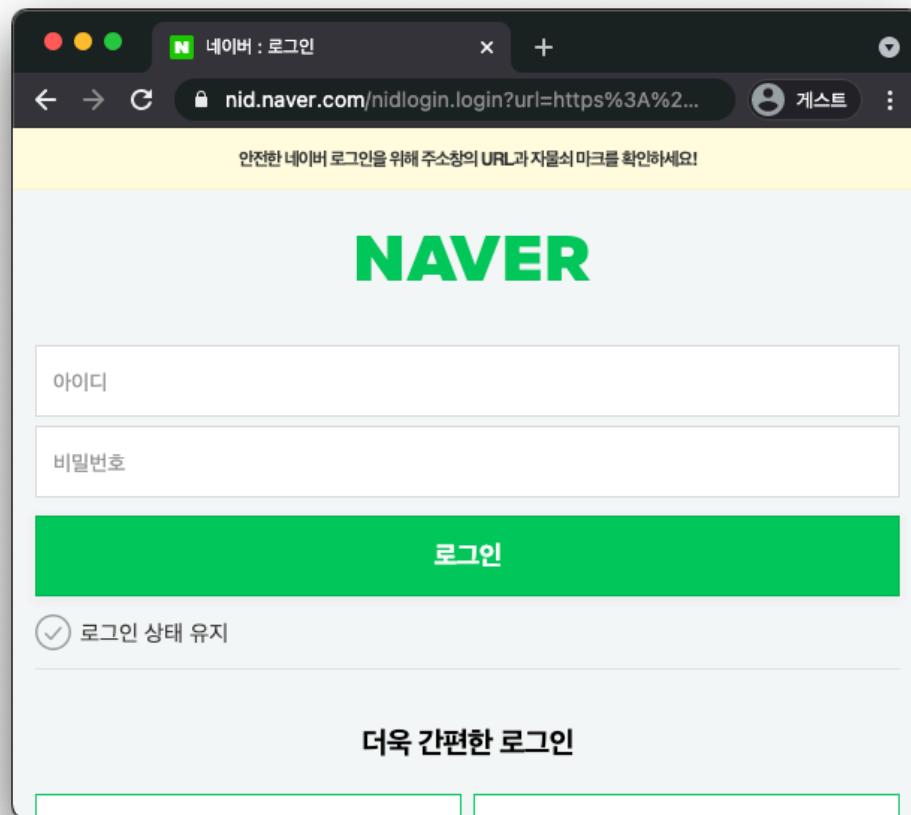
naver에 likelion을 검색했더니, 위 조소에 query = likelion 라는 식으로 보이네요?
이게 바로 GET 의 특징입니다.

GET으로 요청을 보낼 경우

URL에 <https://blah.blah.com?a=blah&b=10> 라는 식으로 요청한 내용이 드러납니다.



HTTP method : post



어, 그럼 만약!

로그인 시에도 GET 메소드를 사용한다면 어떻게 될까요?
naver.com/login?id=singun&password=qwer
라는 식으로, 비밀번호가 드러나는 URL이 될거에요!

보안에 취약하겠죠?

그래서 이럴 때, POST를 사용합니다!
URL에 드러나지 않고 따로 요청에 정보들을 담아서
보내줍니다! (자세한 내용은 컴퓨터 네트워크...)

+) 정보가 길다면...? 파일을 보내야한다면..?



실습 해볼까요?

오늘 실습할 코드는 아래 링크에서 확인할 수 있어요!

https://github.com/likelion-kookmin/django_word_counter



startproject 부터 합시다!

```
● ● ● shinkeonkim@gimsingeon-ui-MacBookPro-2:~/django_word_counter ✘ 1
Last login: Fri Apr 23 23:03:25 on ttys000
→ ~ django-admin startproject django_word_counter
→ ~ cd django_word_counter
→ django_word_counter ls
django_word_counter manage.py
→ django_word_counter └─
```



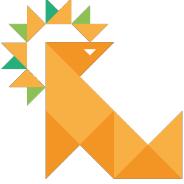
가상환경 설정!

```
● ● ● shinkeonkim@gimsingeon-ui-MacBookPro-2:~/django_word_counter ✘ 1
→ django_word_counter python3 -m venv venv
→ django_word_counter source venv/bin/activate
(venv) → django_word_counter ls
django_word_counter manage.py          venv
(venv) → django_word_counter
```



pip install django

```
● ● ● shinkeonkim@gimsingeon-ui-MacBookPro-2:~/django_word_cou... ✘ 1
(venv) ➜  django_word_counter pip install django
Collecting django
  Using cached Django-3.2-py3-none-any.whl (7.9 MB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.1-py3-none-any.whl (42 kB)
Collecting asgiref<4,>=3.3.2
  Using cached asgiref-3.3.4-py3-none-any.whl (22 kB)
Collecting pytz
  Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.3.4 django-3.2 pytz-2021.1 sqlparse-0.4.1
(venv) ➜  django_word_counter
```



word_count 앱 만들기!

```
shinkeonkim@gimsingeon-ui-MacBookPro-2:~/django_word_counter ✘ 1

(venv) ➜  django_word_counter python manage.py startapp word_counter
(venv) ➜  django_word_counter ls
django_word_counter manage.py          venv                  word_counter
(venv) ➜  django_word_counter
```



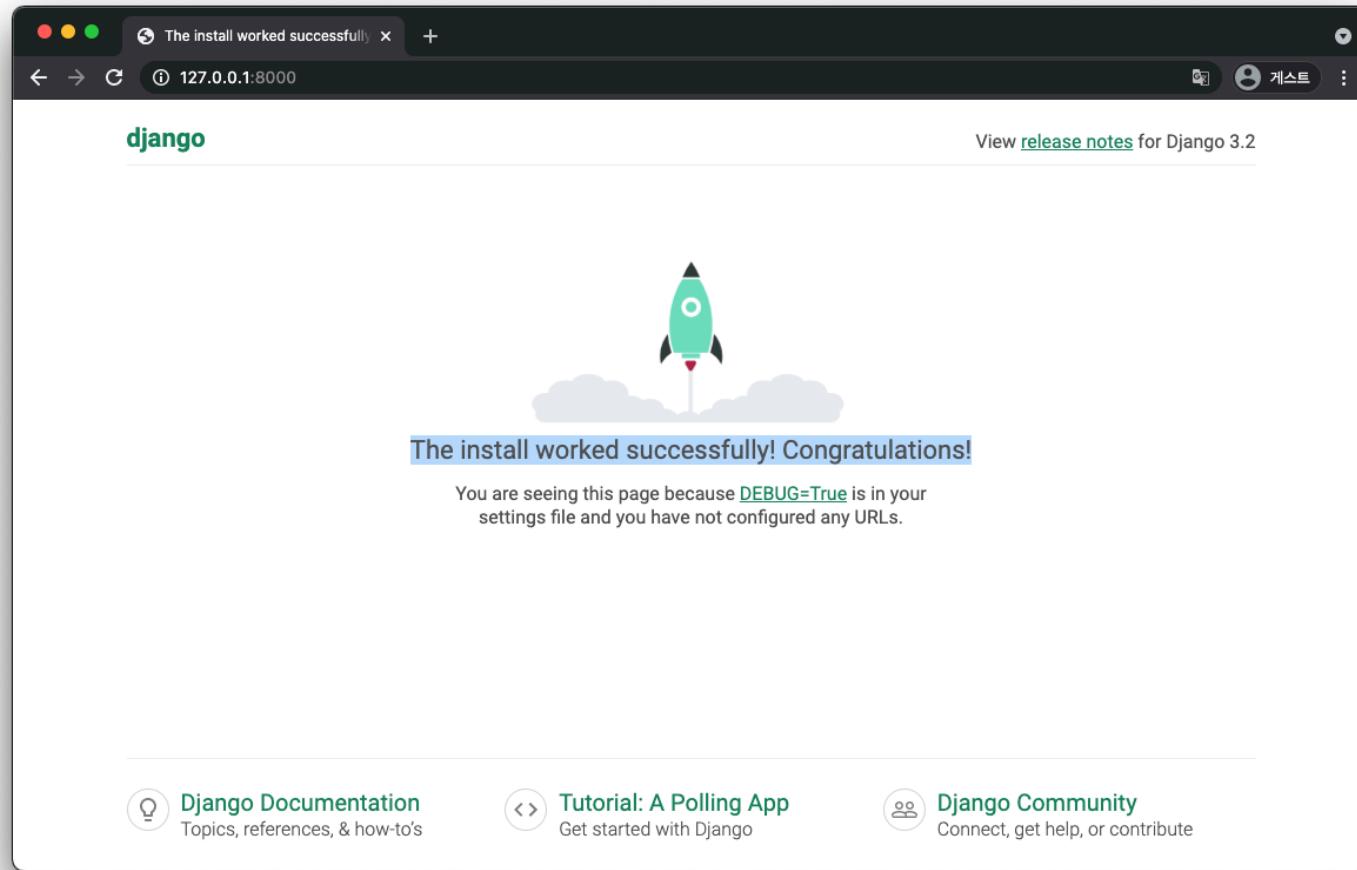
migrate & runserver

```
python manage.py runserver
(venv) ➜ django_word_counter 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 auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
(venv) ➜ django_word_counter python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 24, 2021 - 10:37:48
Django version 3.2, using settings 'django_word_counter.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
```

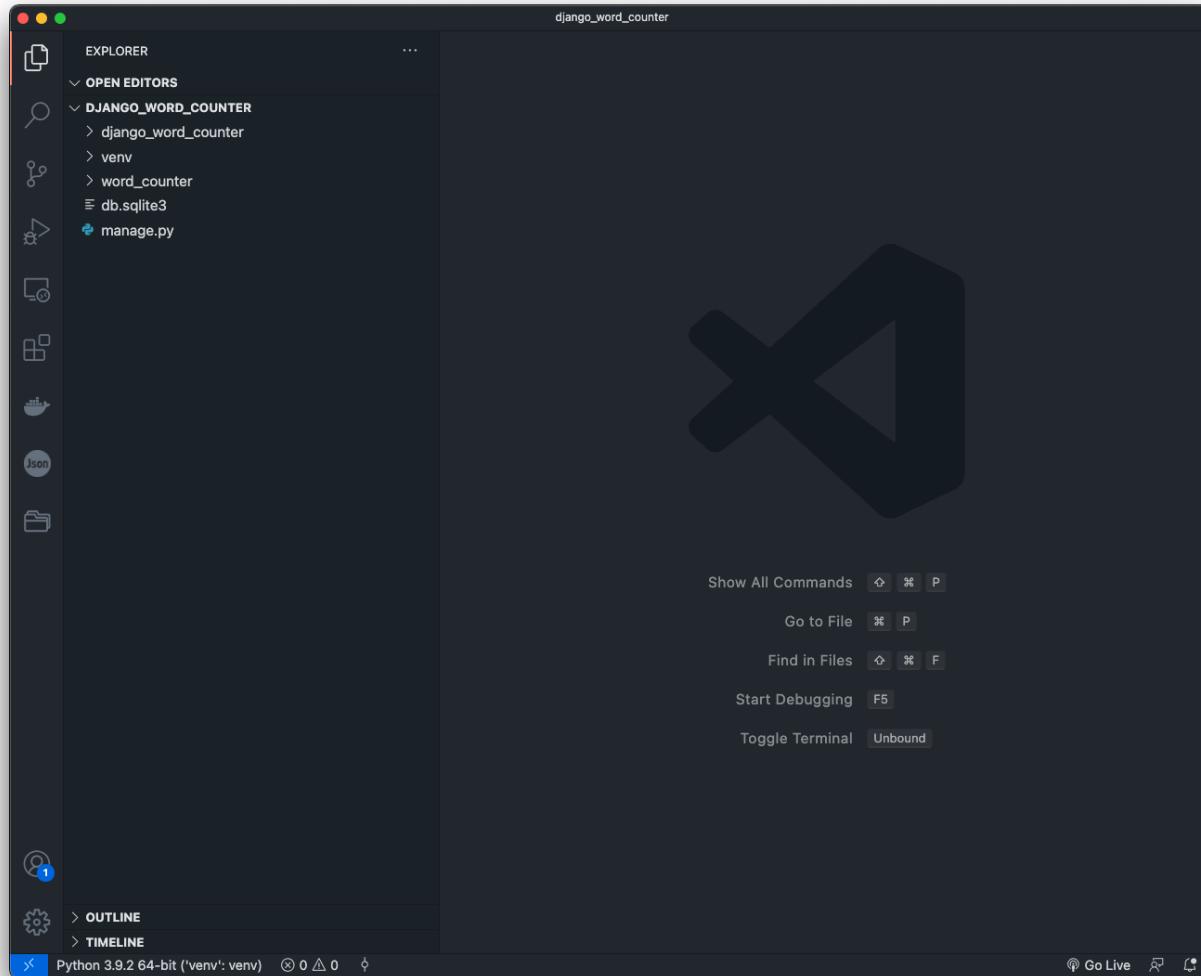


runserver 확인하기!





vscode 알기!



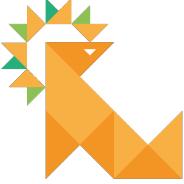


templates 만들기!

The screenshot shows a dark-themed code editor interface, likely PyCharm, displaying a Django project structure. The left sidebar is titled 'EXPLORER' and lists the following directory structure:

- OPEN EDITORS
- index.html templates
- result.html templates
- DJANGO_WORD_COUNTER
- django_word_counter
 - templates
 - index.html
 - result.html
 - venv
 - word_counter
 - db.sqlite3
 - manage.py

The 'templates' folder under 'django_word_counter' is currently selected. The main editor window shows the file 'index.html' with the number '1' at the top. The status bar at the bottom indicates 'Python 3.9.2 64-bit ('venv': venv)' and other details like line count and encoding.



templates에 임시 내용 작성하기

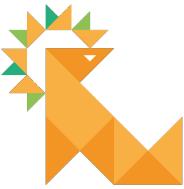
The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the Explorer sidebar, which lists the project structure:

- OPEN EDITORS
- GROUP 1: index.html templates
- GROUP 2: result.html templates
- DJANGO_WORD_COUNTER
 - django_word_counter
 - templates
 - index.html
 - result.html
 - venv
 - word_counter
 - db.sqlite3
 - manage.py

Two code editors are open side-by-side:

- index.html**: Contains the text "여긴 index야!"
- result.html**: Contains the text "여긴 result야!"

The status bar at the bottom indicates Python 3.9.2 64-bit ('venv': venv) and shows the file is a Django Template.



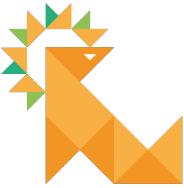
templates 등록하기!

```
settings.py -- django_word_counter
settings.py x
django_word_counter > settings.py > ...
1 """
2 Django settings for django_word_counter project.
3
4 Generated by 'django-admin startproject' using Django 3.1.7.
5
6 For more information on this file, see
7 https://docs.djangoproject.com/en/3.1/topics/settings/
8
9 For the full list of settings and their values, see
10 https://docs.djangoproject.com/en/3.1/ref/settings/
11 """
12
13 from pathlib import Path
14 import os
15
16 # Build paths inside the project like this: BASE_DIR / 'subdir'.
17 BASE_DIR = Path(__file__).resolve().parent.parent
18
19
20 # Quick-start development settings - unsuitable for production
21 # See https://docs.djangoproject.com/en/3.1/howto/deployment/checklist/
22
23 # SECURITY WARNING: keep the secret key used in production secret!
24 SECRET_KEY = '=>k_t=&#brp#+$7_b=i!by*ip15$_!m5q)!36by#08t#brn6'
25
26 # SECURITY WARNING: don't run with debug turned on in production!
27 DEBUG = True
28
29 ALLOWED_HOSTS = []
30
31
32 # Application definition
33
34 INSTALLED_APPS = [
35     'django.contrib.admin',
36     'django.contrib.auth',
37     'django.contrib.contenttypes',
38     'django.contrib.sessions',
39     'django.contrib.messages',
40     'django.contrib.staticfiles',
41 ]
42
43 MIDDLEWARE = [
```

Ln 14, Col 1 (9 selected) Spaces: 4 UTF-8 LF Python ⌂ Go Live ⌂

```
settings.py -- django_word_counter
settings.py x
django_word_counter > settings.py > ...
52
53 ROOT_URLCONF = 'django_word_counter.urls'
54
55 TEMPLATES = [
56     {
57         'BACKEND': 'django.template.backends.django.DjangoTemplates',
58         'DIRS': [os.path.join(BASE_DIR, 'templates')],
59         'APP_DIRS': True,
60         'OPTIONS': {
61             'context_processors': [
62                 'django.template.context_processors.debug',
63                 'django.template.context_processors.request',
64                 'django.contrib.auth.context_processors.auth',
65                 'django.contrib.messages.context_processors.messages',
66             ],
67         },
68     },
69 ]
70
71 WSGI_APPLICATION = 'django_word_counter.wsgi.application'
72
73
74 # Database
75 # https://docs.djangoproject.com/en/3.1/ref/settings/#databases
76
77 DATABASES = {
78     'default': {
79         'ENGINE': 'django.db.backends.sqlite3',
80         'NAME': BASE_DIR / 'db.sqlite3',
81     }
82 }
83
84
85 # Password validation
86 # https://docs.djangoproject.com/en/3.1/ref/settings/#auth-password-validation
87
88 AUTH_PASSWORD_VALIDATORS = [
89     {
90         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
91     },
92     {
93         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
94     },
95 ]
```

Ln 58, Col 18 (35 selected) Spaces: 4 UTF-8 LF Python ⌂ Go Live ⌂



views.py 임시 내용 작성하기!

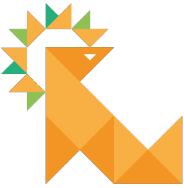
The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following details:

- Explorer View:** Shows the project structure under "OPEN EDITORS".
 - The current file, `views.py`, is highlighted with a blue selection bar.
 - Other files include `word_counter`, `migrations`, `admin.py`, `apps.py`, `models.py`, `tests.py`, `db.sqlite3`, and `manage.py`.
- Code Editor View:** The file `views.py` is open, showing the following Python code:

```
from django.shortcuts import render

# Create your views here.
def index(request):
    return render(request, 'index.html')

def result(request):
    return render(request, 'result.html')
```
- Bottom Status Bar:** Displays the Python version (`Python 3.9.2 64-bit ('venv': venv)`), line and column information (`Ln 9, Col 1`), character encoding (`Spaces: 4`), and file type (`UTF-8 LF Python`).



urls.py 만들고! 작성하기!

The screenshot shows a Visual Studio Code (VS Code) interface with a dark theme. On the left is the Explorer sidebar, which lists the project structure:

- OPEN EDITORS: urls.py (selected)
- DJANGO_WORD_COUNTER:
 - django_word_counter
 - templates
 - venv
 - word_counter
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py (selected)
 - views.py
 - db.sqlite3
 - manage.py

The main editor area displays the contents of urls.py:

```
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name="index"),
    path('result', views.result, name="result")
]
```

At the bottom of the interface, status bar information includes:

 - Python 3.9.2 64-bit ('venv': venv)
 - Ln 8, Col 1
 - Spaces: 4
 - UTF-8
 - LF
 - Python
 - Go Live



만든 urls.py 연결하기!

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists the project structure:

- OPEN EDITORS: urls.py (django_word_counter)
- DJANGO_WORD_COUNTER:
 - django_word_counter
 - __pycache__
 - __init__.py
 - asgi.py
 - settings.py
 - urls.py (highlighted with a blue selection bar)
 - wsgi.py
 - templates
 - venv
 - word_counter
 - db.sqlite3
 - manage.py

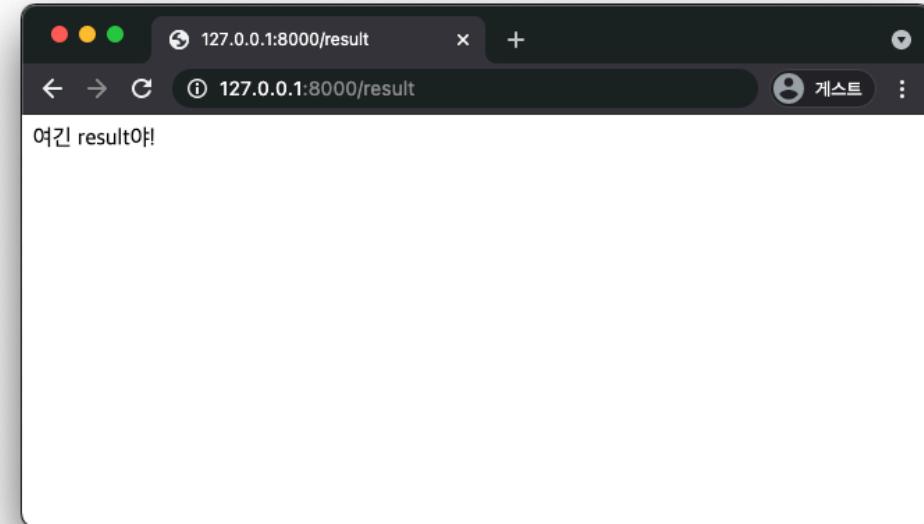
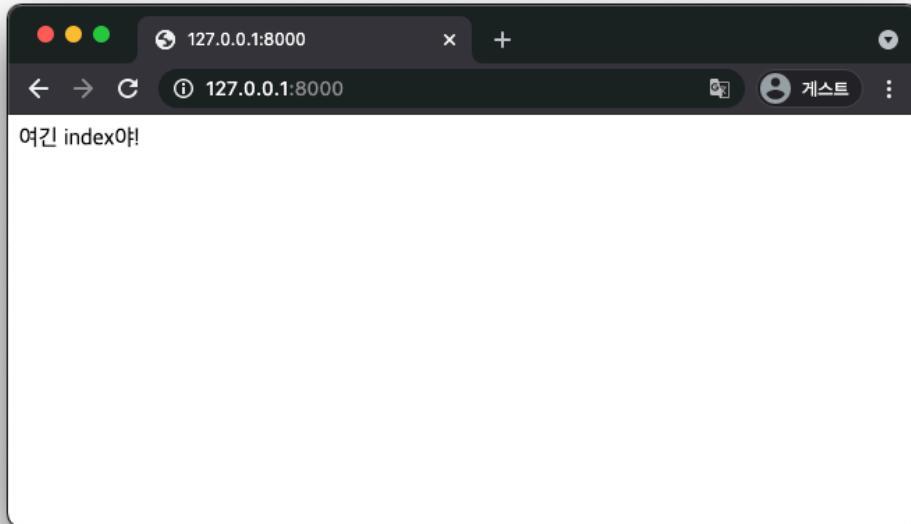
The status bar at the bottom indicates "Python 3.9.2 64-bit ('venv': venv)" and "Ln 8, Col 1".

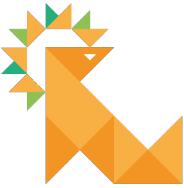
The main editor window displays the contents of urls.py:

```
urls.py — django_word_counter
django_word_counter > urls.py > ...
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('', include('word_counter.urls')),
6     path('admin/', admin.site.urls),
7 ]
8
```



한번 화면이 나오는지 확인해볼까요?





index.html 작성하기!

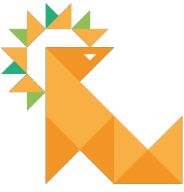
The screenshot shows the VS Code interface with the following details:

- EXPLORER** sidebar:
 - OPEN EDITORS: index.html templates
 - DJANGO...: django_word_counter (selected)
 - templates: index.html (highlighted)
 - result.html
 - venv
 - word_counter
 - db.sqlite3
 - manage.py
- index.html** editor tab: index.html – django_word_counter
- Content of index.html:

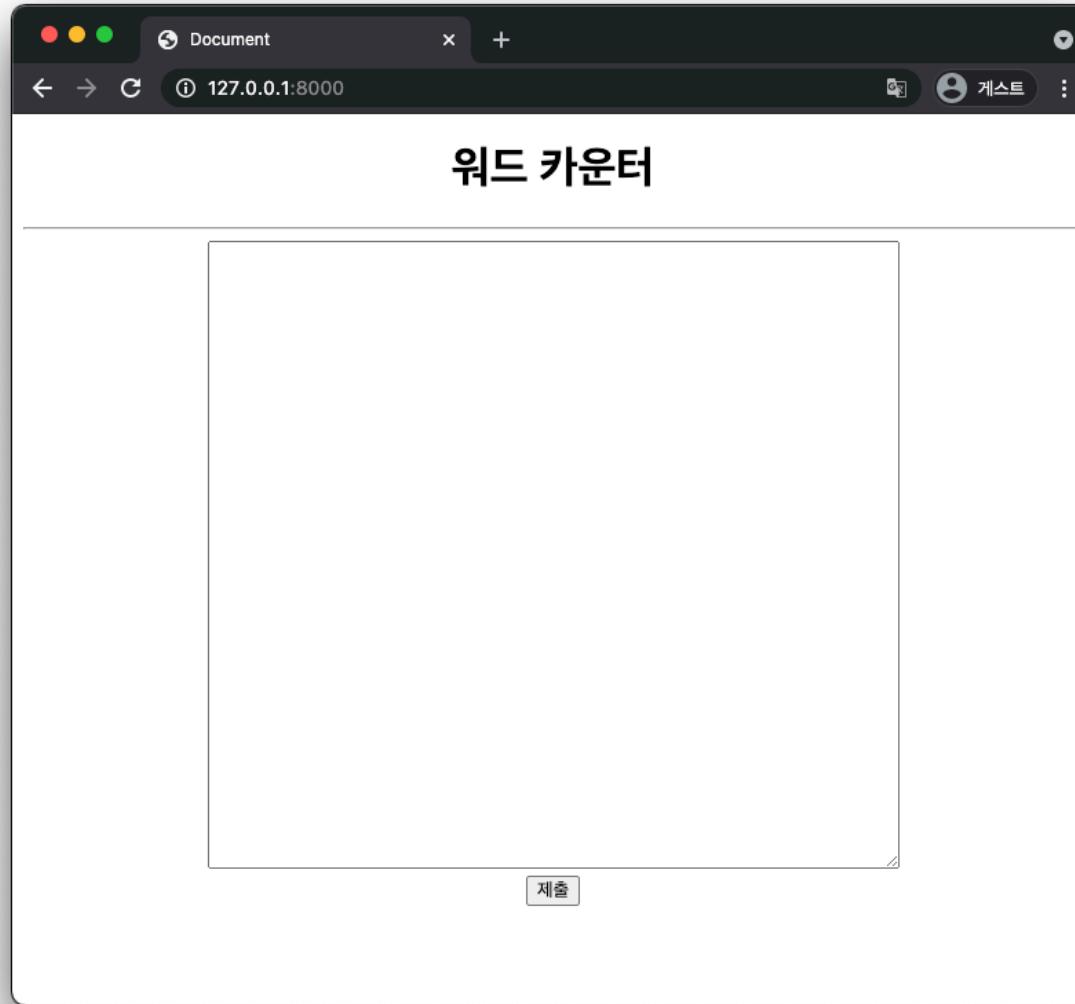
```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Document</title>
    <style>
      body {
        text-align: center;
      }
    </style>
  </head>

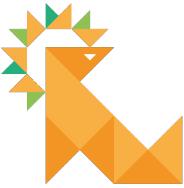
  <body>
    <h1>
      워드 카운터
    </h1>
    <hr/>
    <form action="{% url 'result' %}" method="POST">
      {% csrf_token %}
      <textarea name="sentence" rows="30" cols="60"></textarea><br>
      <input type="submit"></input>
    </form>
  </body>
</html>
```
- Bottom status bar: Python 3.9.2 64-bit ('venv': venv) | 0 ▲ 0 ⚡ | Ln 27, Col 1 | Spaces: 4 | UTF-8 | LF | Django Template | Go Live | ⚡ | ⚡

csrf_token은 뭘까요?
→ CSRF 공격을 방지하기 위해
존재하는 토큰입니다.
(웹 취약점 공격 중 하나)



index.html 결과 확인!





POST로 넘어온 데이터 확인해보기!

Document

127.0.0.1:8000

워드 카운터

이건 test입니다!

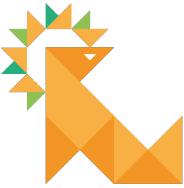
제출

```
python manage.py runserver
/Users/shinkeonkim/django_word_counter/templates/index.html changed, reloading.
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 24, 2021 - 10:53:25
Django version 3.2, using settings 'django_word_counter.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
/Users/shinkeonkim/django_word_counter/templates/index.html changed, reloading.
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 24, 2021 - 10:53:27
Django version 3.2, using settings 'django_word_counter.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[24/Apr/2021 10:54:15] "GET / HTTP/1.1" 200 680
/Users/shinkeonkim/django_word_counter/word_counter/views.py changed, reloading.
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
April 24, 2021 - 10:55:57
Django version 3.2, using settings 'django_word_counter.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CONTROL-C.
[24/Apr/2021 10:56:02] "GET / HTTP/1.1" 200 680
이건 test입니다!
[24/Apr/2021 10:56:36] "POST /result HTTP/1.1" 200 17
```



POST로 넘어온 데이터 확인해보기!

The screenshot shows a code editor interface with a dark theme. On the left is the Explorer sidebar, which lists the project structure:

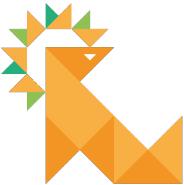
- OPEN EDITORS: views.py (word_counter)
- DJANGO_WORD_COUNTER:
 - django_word_counter
 - templates
 - venv
 - word_counter
 - __pycache__
 - migrations
 - __init__.py
 - admin.py
 - apps.py
 - models.py
 - tests.py
 - urls.py
 - views.py
- db.sqlite3
- manage.py

The main editor area displays the contents of `views.py`:

```
views.py — django_word_counter
word_counter > views.py > result
1 from django.shortcuts import render
2
3 # Create your views here.
4 def index(request):
5     return render(request, 'index.html')
6
7 def result(request):
8     print(request.POST.get('sentence'))
9     return render(request, 'result.html')
10
```

The status bar at the bottom shows the following information:

- Python 3.9.2 64-bit ('venv': venv)
- Ln 7, Col 21
- Spaces: 4
- UTF-8 LF
- Python
- Go Live



views.py 작성하기

The screenshot shows a dark-themed instance of Visual Studio Code (VS Code) with the following details:

- File Explorer:** On the left, it shows the project structure under "OPEN EDITORS". The "views.py" file in the "word_counter" directory is currently selected.
- Code Editor:** The main area displays the "views.py" code for a Django application named "django_word_counter". The code defines two functions: "index" and "result". The "result" function retrieves a sentence from the POST request, splits it into words, and counts the occurrences of each word. The results are then sorted and rendered to the user.
- Annotations:** A red annotation box highlights the "if" condition in the "result" function's loop, with the text "와! if 구문하여 +1을 할까요?" overlaid.
- Status Bar:** At the bottom, it shows the Python version ("Python 3.9.2 64-bit ('venv': venv)"), line and column information ("Ln 22, Col 1"), and file encoding ("UTF-8").



result.html 작성하기

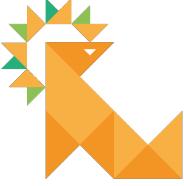
The screenshot shows the VS Code interface with the following details:

- EXPLORER:** Shows the project structure:
 - templates > result.html
 - DJANGO... > django_word_counter
 - templates > index.html
 - result.html (selected)
 - > venv
 - > word_counter
 - = db.sqlite3
 - manage.py
- RESULT:** Shows the content of result.html:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>결과</title>
    <style>
        body {
            text-align: center;
        }
    </style>
</head>

<body>
    <h1>
        결과
    </h1>
    <hr/>
    {% for word, count in word_counts %}
        {{word}} : {{count}} <br>
    {% endfor %}
</body>
</html>
```

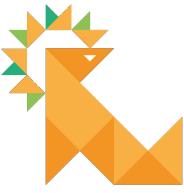
- Annotations:** A red box highlights the Django template loop from line 20 to 24. To the right of the code, the Korean text "이건 뭘까요?" (What is this?) is displayed in red.
- STATUS BAR:** Python 3.9.2 64-bit ('venv': venv) | Spaces: 2 | UTF-8 | LF | Django Template | Go Live



django template 작성하기!

django template tag들을 이용해서 python 문법으로 html을 작성할 수 있어요!

이걸 왜 쓸까요?



확인해보기!

Document

127.0.0.1:8000

워드 카운터

우리 우리 이제 행복하자
제발 제발 제발

제출

결과

제발 : 3
우리 : 2
이제 : 1
행복하자 : 1

