



엠비션 유튜브에 영상이 올라올 때마다, 매번 거의 동일한 형식의 댓글을 남기는 사람이 있다.




최용민 4일 전

오늘의 엠비션 점수  
[강자석 - 9618위] (전적없음)  
D1 2P - 238승 241패  
[강찬밥 - 679위] (5승 3패)  
GM 510P - 243승 195패  
오랜만의 저녁밥이네오 잘먹겠습니다~  
간략히


👍 138 🔄  답글

▼ 답글 10개 보기




최용민 3일 전

오늘의 엠비션 점수  
[강자석 - 9595위] (전적없음)  
D1 2P - 238승 241패  
[강찬밥 - 681위] (전적없음)  
GM 510P - 243승 195패  
전적이 없어서 고수달 전적을 들고왔습니다  
[고수달] D2 - 52P 790승 793패


👍 315 🔄  답글

▼ 답글 25개 보기




최용민 1일 전(수정됨)

오늘의 엠비션 점수  
[강자석 - 9595위] (전적없음)  
D1 2P - 238승 241패  
[강찬밥 - 820위] (4승 4패)  
GM 455P - 251승 206패  
이 남자.. 챌린저에선 어떨까?  
간략히

👍 124 🔄  답글

▼ 답글 7개 보기




최용민 2시간 전

오늘의 엠비션 점수  
[강자석] (전적없음)  
D1 2P - 238승 241패  
[강찬밥 - 828위] (전적없음)  
GM 455P - 251승 206패  
식당 휴무일입니다 오늘은 안늦었습니다  
간략히


👍 167 🔄 답글

▼ 답글 14개 보기



가람

연가람 1일 전(수정됨)  
최용민수가 아직 안와서 찾아본  
오늘의 엠비션 점수  
[강찬밥 - 821위] (4승 4패)  
GM 455P - 251승 206패

👍 276 🔄  답글

▼ 답글 10개 보기

대신 올려주는 사람까지 있다 ㅋㅋㅋ

보자마자 번뜩 아이디어가 떠올랐다. 최용민수를 위해 오늘의 앰비션 점수를 받을 수 있는 서비스를 만들어보기로 했다.

```
$ python main.py
오늘의 앰비션 점수
[강자석 - 9,546위] (전적없음)
D1 2P - 238승 241패
[강찬밥 - 827위] (전적없음)
GM 455P - 251승 206패
이 남자.. 헬린저에선 어떨까?
```

### 프로토타입을 터미널에서 실행해본 결과

웹 사이트에서 버튼 한 번 눌러서 이 문자열을 받을 수 있다면 최용민수는 더 이상 댓글 작성을 위해 전적 검색 사이트를 이용할 필요가 없다!

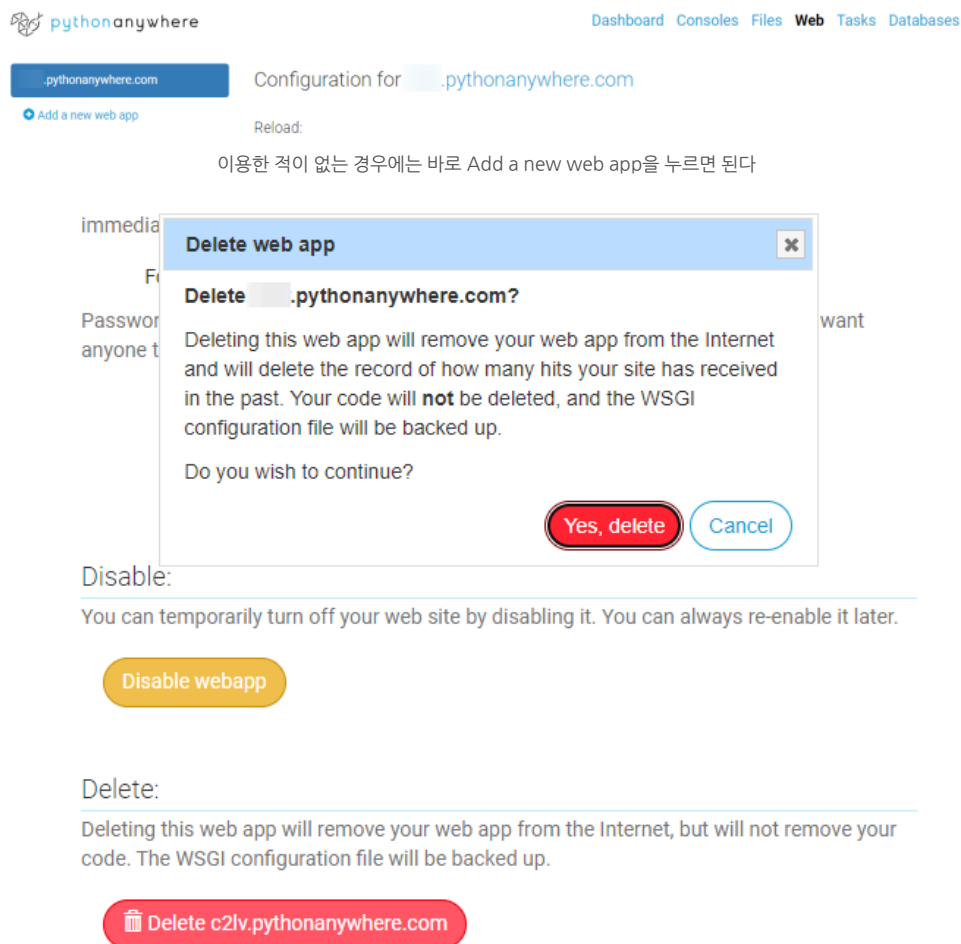
어쩌다보니 토이 프로젝트로 1년 전에 생각했던 아이디어를 이제서야 구현하게 되었다. 지금부터는 프로젝트의 백엔드를 Flask로 구현하여 pythonanywhere로 배포하는 과정과 그 과정에서 겪은 에러, 에러의 해결 방법을 적어보

려고 한다. 프로젝트의 깃허브 링크는 다음과 같다.  
<https://github.com/c2lv/Today-s-Ambition-score.git>

# pythonanywhere를 이용한 Flask 프로젝트 배포 과정

## 1. 회원가입 및 웹 앱 생성

파이썬어니웨어(<https://www.pythonanywhere.com>)에서 무료계정을 이용하면 무료로 웹 호스팅을 진행할 수 있다.  
물론 무료이다보니 하나의 웹 앱만 배포 가능하고, 콘솔 창을 동시에 두 개까지만 이용할 수 있는 등 다소 제약이 존재하는 한다.  
나는 이전에 가입해 이용했던 적이 있어, 기존 프로젝트를 삭제하고 새로 만들어서 진행했다. 홈 화면에서 'Web'을 클릭해보자.



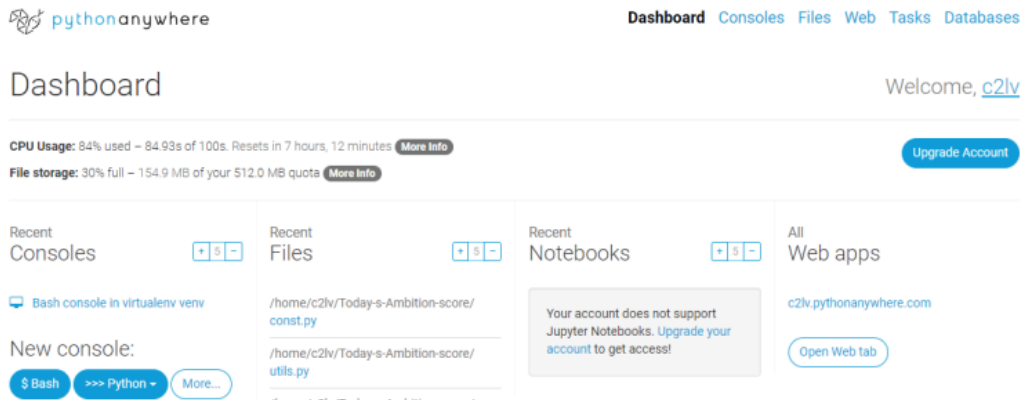
이용한 적이 있는 경우 기존 웹 앱을 지우고 Add a new web app을 누르면 된다  
원래 URL 가리려고 했는데, 이 사진 밑에 안 가린 거 보고 귀찮아서 안 가리려고 한다

'Add a new web app'을 클릭한 이후 생성 과정은 캡처해두지 못했는데, 본인 프로젝트에 해당하는 파이썬 웹 프레임워크와 파이썬 버전을 선택해주면 된다. 나는 최대한 최신 버전을 선택해 Flask는 1.1.1 버전, Python은 3.6 버전을 선택한 것으로 기억한다.

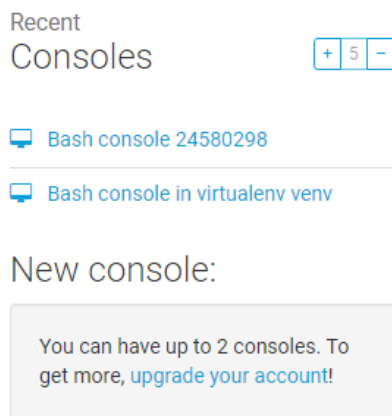
그런데 어차피 가상 환경을 이용할 것이므로 여기서의 선택지가 중요한 지는 모르겠다.

## 2. 프로젝트 업로드 및 가상 환경 설정

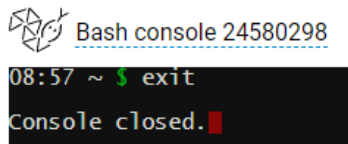
프로젝트를 업로드하기 위해 'Dashboard'에서 '\$ Bash'를 클릭한다.



'\$ Bash'는 좌측 하단에 있다



기존에 2개의 콘솔을 이용중이었다면 이렇게 나온다  
실행중인 콘솔을 종료시키고 새로 만들거나  
실행중인 콘솔이 Bash라면 거기서 진행해도 된다



종료는 exit을 입력하거나 Ctrl + D 입력

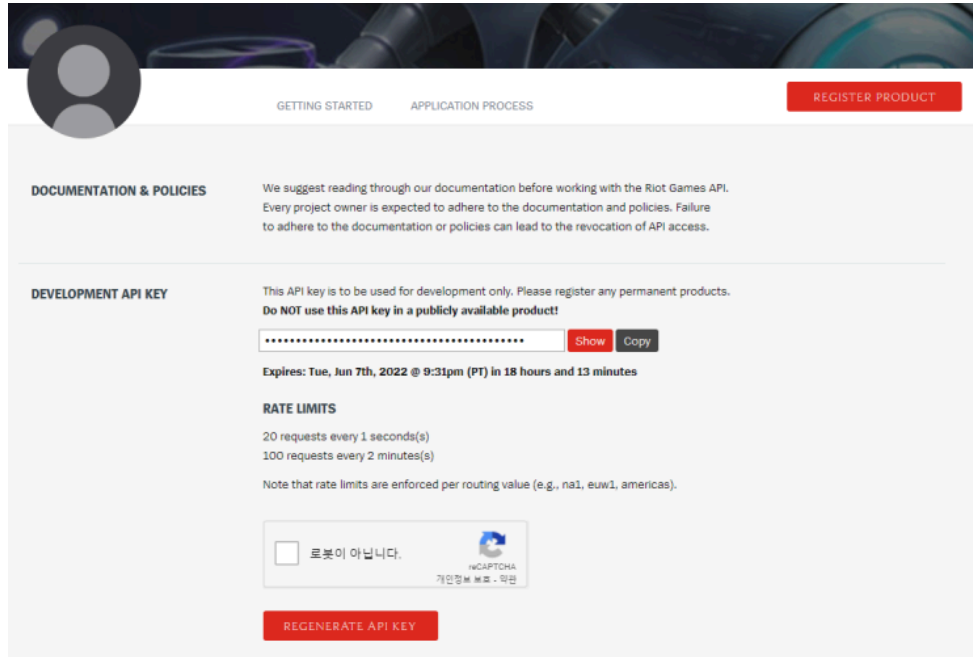
Bash console이 실행되면 다음과 같이 입력한다.

```
git clone https://github.com/c2lv/Today-s-Ambition-score.git
cd Today-s-Ambition-score
pip install virtualenv
virtualenv --python=python3.6 venv
source venv/bin/activate
pip install -r requirements.txt
echo export "RIOTGAMES_API_KEY=Enter Your DEVELOPMENT_API_KEY" >> .env
```

- line 1: 깃허브(github)에서 프로젝트 폴더를 복사해온다
- line 2: 복사한 프로젝트 폴더로 이동한다
- line 3: 가상환경 패키지(virtualenv)를 설치한다
- line 4: python 3.6 버전의 가상환경 venv를 생성한다
- line 5: 가상환경을 활성화한다
- line 6: 가상환경에 requirements.txt에 적힌 패키지들을 설치한다

line 7: 프로젝트에서 사용되는 환경변수인 RIOTGAMES\_API\_KEY를 설정하기 위해 환경변수 값이 담긴 .env 파일을 생성한다

(Enter Your DEVELOPMENT\_API\_KEY 자리에 자신의 DEVELOPMENT\_API\_KEY 값을 넣어주어야 한다)



DEVELOPMENT\_API\_KEY는 라이엇게임즈 개발자 페이지에서 'Show' 버튼을 누르면 확인할 수 있다  
해당 페이지에서 확인할 수 있는 개발용 키의 경우 유효기간이 하루이다

virtualenv로 가상환경 만들 때 파이썬 버전까지 선택하는 것은 이번이 처음이다.

requirements.txt는 현재 환경의 패키지과 해당 패키지 버전을 requirements.txt에 저장하는 `pip freeze > requirements.txt` 명령어로 만들어진다. requirements.txt가 있으면 해당 프로젝트에 필요한 패키지를 `pip install -r requirements.txt` 명령어로 한꺼번에 설치할 수 있기 때문에 편리하다. 명령어만 이용해서 만들다보니 잘 몰랐는데, 이번 프로젝트를 진행하며 패키지 버전을 다양하게 표현할 수 있다는 것을 알게 되었다. 파이썬처럼 '#'으로 주석 사용도 가능했다.


A==3.0	A package 3.0 버전 설치
A>=3.0	A package 3.0 버전 이상 중 하나 설치
A<=3.0	A package 3.0 버전 이하 중 하나 설치
A!=3.0	3.0 버전을 제외한 A package 중 하나 설치
A	A package 최신 버전 설치
A>=3.0,<=4.0	A package 3.0 이상 4.0 이하 버전 중 하나 설치
A~>=3.0	A package 3.0 이상인 3 버전대 중 하나 설치 (A>=3.0, ==3.*와 동일) <a href="https://peps.python.org/pep-0440/#compatible-release">https://peps.python.org/pep-0440/#compatible-release</a>
A==3.*	A package 3 버전대 중 하나 설치

환경변수의 경우 파이썬애니웨어 도움말 페이지(<https://help.pythonanywhere.com/pages/environment-variables-for-web-apps/>)를 참고해서 설정했다. '.env' 파일을 생성한 후 한 번 더 설정이 필요한데, 나는 WSGI file을 수정하는 방법을 택했다.

### 3. 웹 앱 환경 설정

다시 'Web'을 클릭해서 웹 앱 환경 설정 페이지로 이동한다. 위에서부터 하나씩 살펴보자.

Reload:

 Reload c2lv.pythonanywhere.com

Best before date:

We're happy to host your free website – and keep it free – for as long as you want to keep it running, but you'll need to log in at least once every three months and click the "Run until 3 months from today" button below. We'll send you an email a week before the site is disabled so that you don't forget to do that. [See here for more details.](#)

This site will be disabled on **Wednesday 07 September 2022**

[Run until 3 months from today](#)

[Paying users'](#) sites stay up forever without any need to log in to keep them running.

Reload 버튼을 누르면 사이트가 갱신된다. Flask는 Django와 다르게 수정사항이 생기면 저장 후 다시 실행시켜주어야 한다. 따라서 웹 앱 코드를 수정하고 나면 사이트 새로고침과는 별개로 Reload 버튼을 눌러주어야 했다. Django로 배포하면 수정 후 따로 Reload를 안 해줘도 되는지 궁금해진다. 아래 'Run until 3 months from today' 버튼은 무료 호스팅의 제약 중 하나다. 버튼을 누른 후 3개월이 지나면 호스팅이 비활성화된다. 비활성화 일주일 전에 메일로 알려준다고 하니 당장은 신경쓸 사항이 아니다.

Code:

What your site is running.

Source code:	<a href="#">/home/c2lv/Today-s-Ambition-score</a>	<a href="#">Go to directory</a>
Working directory:	<a href="#">/home/c2lv/</a>	<a href="#">Go to directory</a>
WSGI configuration file:	<a href="#">/var/www/c2lv_pythonanywhere_com_wsgi.py</a>	
Python version:	3.6	

클릭해서 경로를 맞춰준다. 보통 Working directory는 '/home/{아이디}/', Source code는 '/home/{아이디}/{레포지토리 이름}'으로 맞추는 듯하다. 파이썬 버전도 다르길래 맞춰주었다. 버전이 가상 환경 기준인지 글로벌 환경 기준인지 정확히는 잘 모르겠다.

WSGI configuration file 경로를 클릭하면 해당 파일 수정 페이지로 이동하는데, [도움말 페이지](#)에 나온 대로 아래와 같이 수정해주었다.



[/var/www/c2lv\\_pythonanywhere\\_com\\_wsgi.py](#)

```
1  # This file contains the WSGI configuration required to serve up your
2  # web application at http://<your-username>.pythonanywhere.com/
3  # It works by setting the variable 'application' to a WSGI handler of some
4  # description.
5  #
6  # The below has been auto-generated for your Flask project
7
8  import sys
9  import os
10 from dotenv import load_dotenv
11
12 # add your project directory to the sys.path
13 project_home = '/home/c2lv/Today-s-Ambition-score'
14 if project_home not in sys.path:
15     sys.path = [project_home] + sys.path
16
17 load_dotenv(os.path.join(project_home, '.env'))
18
19 # import flask app but need to call it "application" for WSGI to work
20 from flask_app import app as application # noqa
21
```

## Virtualenv:

Use a virtualenv to get different versions of flask, django etc from our default system ones. [More info here](#). You need to **Reload your web app** to activate it; NB - will do nothing if the virtualenv does not exist.

</home/c2lv/Today-s-Ambition-score/venv>

[Start a console in this virtualenv](#)

가상환경 경로도 Code에서 한 것처럼 맞춰준다.

## Log files:

The first place to look if something goes wrong.

Access log: [c2lv.pythonanywhere.com.access.log](#)

Error log: [c2lv.pythonanywhere.com.error.log](#)

Server log: [c2lv.pythonanywhere.com.server.log](#)

Log files are periodically rotated. You can find old logs here: [/var/log](#)

나 같은 경우 배포 과정에서 에러를 잡는 데 Error log와 Server log가 도움이 되었다.

```
2022-06-07 07:41:23 ##### http://fow.kr/find/강자석
2022-06-07 07:41:23 ##### 403
2022-06-07 07:41:23 Error from get_today_wins_losses_and_ranking
print()로 URL, 에러 코드, 에러가 발생한 함수 등을 출력해보고 Server log 파일에서 확인했다
```

Error log에서 에러를 발견하고, Server log에서 이것저것 출력해본 결과를 보며 문제가 무엇인지 구체적으로 찾아 나갈 수 있다.

## Security:

An HTTPS certificate is needed so that people can access your site securely. We automatically provide a certificate for [c2lv.pythonanywhere.com](#).

HTTPS certificate: Automatically provided for this hostname

You need to **Reload your web app** to activate any changes made below.

Forcing HTTPS means that anyone who goes to your site using the insecure http URL will immediately be redirected to the secure https one. [More information here](#).

Force HTTPS: ☒ Enabled

Password protection is ideal for sites that are under development when you don't want anyone to see them yet.

Password protection: ☐ Disabled

Username: [Enter a username](#)

Password: [Enter a password](#)

## Disable:

You can temporarily turn off your web site by disabling it. You can always re-enable it later.

[Disable webapp](#)

## Delete:

Deleting this web app will remove your web app from the Internet, but will not remove your code. The WSGI configuration file will be backed up.

[Delete c2lv.pythonanywhere.com](#)

마지막은 한꺼번에. Security에서 Force HTTPS를 활성화 해준다. HTTP를 사용하는 등 안전하지 않은 URL로 해당 웹 앱에 접근했을 때 안전한 URL로 리디렉션 시켜주는데, 딱히 안 쓸 이유가 없는 것 같아서 활성화해주었다.

Disable 버튼은 호스팅 비활성화시킬 때, Delete 버튼은 웹 앱 지울 때 누르면 된다.

해당 프로젝트는 DB와 Static files 연동 과정이 필요 없으니 생략했다. 이제 [c2lv.pythonanywhere.com](#)에 들어가면 Flask 프로젝트가 잘 배포되어 있어야 하는데, 역시나 여러 에러들이 발목을 잡았다. 지금부터는 배포 과정에서 발

견한 에러와 해결 방법을 적어보려고 한다.

## 에러 1: SyntaxError: Non-ASCII character '\Wxec' in file {file\_name}.py on line {line\_number}, but no encoding declared; see <http://python.org/dev/peps/pep-0263/> for details.

처음 가상환경을 설정할 때 버전 설정을 안 해줬다가 2버전대 파이썬 가상환경이 만들어졌다. 여기서 플라스크 실행 테스트를 해보다가 발생한 에러이다. 해당 버전의 파이썬은 기본적으로 ASCII를 표준 인코딩으로 사용하기 때문에 한글 문자가 에러를 발생시킨 것이다. 해당 에러가 발생한 파일 첫 번째 또는 두 번째 줄에 인코딩을 명시해주면 해결된다. 아래와 같은 형식으로!

```
# -*- coding: utf-8 -*-
```

다른 형식도 가능한데, 보다 자세한 사항은 에러 링크를 참고하면 된다.  
나 같은 경우 애초에 의도한 파이썬 버전이 아니었기 때문에, 그냥 다시 파이썬 3.6 버전의 가상환경을 만들어서 해결했다.

## 에러 2: TypeError: must be str, not NoneType

로컬에서 하던 것처럼 RIOTGAMES\_API\_KEY를 콘솔에서 export 명령어로 추가해주면 될 줄 알았는데, 아니었다. 환경 변수가 추가되지 않아 값이 들어있어야 할 키가 비게 되어 발생한 에러이다. 앞서 했던 것처럼 .env 파일을 추가하고 WSGI file을 수정해주어야 한다.

## 에러 3: OSError: Tunnel connection failed: 403 Forbidden

```
urllib3.exceptions.MaxRetryError: HTTPSConnectionPool(host='fow.kr', port=443): Max retries
exceeded with url: /find/%EA%B0%95%EC%9E%90%EC%84%9D (Caused by
ProxyError('Cannot connect to proxy.', OSError('Tunnel connection failed: 403 Forbidden',)))
requests.exceptions.ProxyError: HTTPSConnectionPool(host='fow.kr', port=443): Max retries
exceeded with url:
/find/%EA%B0%95%EC%9E%90%EC%84%9D (Caused by ProxyError('Cannot connect to
proxy.', OSError('Tunnel connection failed: 403 Forbidden',)))
```

위 두 에러가 뒤따라왔다. 마지막이 되어서야 밝히지만, 이것 때문에 pythonanywhere 배포를 포기했다. 도움말 페이지에도 나와 있던 에러인데, 간단히 말해 무료 계정에는 안전 문제로 웹 사이트 액세스에 제한이 있다. 화이트리스트에 포함된 사이트만 허용되는데, fow.kr은 포함되어있지 않다. 추가 요청을 할 수는 있는데 공식, 공개, 문서화된 API가 있는 사이트, 즉 사람이 사용하는 것이 아니라 기계가 사용하도록 설계되고 의도된 사이트만 화이트리스트에 추가한다고 도움말 페이지에 명시되어 있다. fow.kr은 이에 부합하지 않아 추가 신청이 의미가 없다.

참고로 '%EA%B0%95%EC%9E%90%EC%84%9D'는 앰비션의 룰 부계정 닉네임인 강자석이다.  
관련 도움말 페이지: [403 Forbidden error](#), [Requesting Whitelist additions](#), [Whitelisted sites for free users](#)

결국 pythonanywhere에서 열심히 삽질해놓고 heroku로 배포하게 됐다. 기왕 삽질한 거 아까우니 기록이라도 잘 남겨야지.