

개발이 완료된 장고프로젝트는 현재 로컬에서만 테스트 가능하다.

우리가 팀프로젝트로 개발한 장고프로젝트를 언제든 그리고 어디서든 접속하여 페이지를 사용할 수 있도록 실제 웹 사이트에 배포하는 것이 여러모로 좋을 것이다.

이렇게 개발이 완료된 웹 프로젝트의 결과물 즉 웹 애플리케이션을 웹상에서 서비스해 주는 기술을 웹 호스팅 서비스라고 하며 파이썬으로 구현한 웹 애플리케이션은 pythonanywhere 사이트에 올려서(배포해서) 서비스하는 것을 추천한다.

PythonAnywhere

무료 백과 사전, 위키피디아에서

PythonAnywhere 는 Python 프로그래밍 언어를 기반으로 하는 온라인 IDE ( 통합 개발 환경 ) 및 웹 호스팅 서비스 ( Platform as a Service ) 입니다. [1] 2012 년 Giles Thomas와 Robert Smithson이 설립 한이 제품은 구문 강조 기능 이있는 코드 편집기 와 함께 서버 기반 Python 및 Bash 명령 줄 인터페이스에 대한 브라우저 내 액세스를 제공합니다 . 프로그램 파일은 사용자의 브라우저를 사용하여 서비스와주고받을 수 있습니다. 서비스에서 호스팅하는 웹 응용 프로그램은 WSGI 기반 응용 프로그램 프레임 워크를 사용하여 작성할 수 있습니다 .

PythonAnywhere는 Python 기반 스크립트 프로그램 인 Resolver One을 제작 한 Resolver Systems, [2] 에서 만들었습니다 . 2012 년 10 월 16 일에 제품은 새로운 회사 인 PythonAnywhere LLP에 인수되었습니다.이 회사는 향후 제품을 개발 및 유지 관리하고 기존 개발 팀을 인수했습니다. [3]

개발 팀은 PythonAnywhere를 사용하여 PythonAnywhere를 개발하고, 협업 기능이 극단적 인 프로그래밍 방법 을 사용하기 때문에 도움이된다고 말합니다 . [4]

**내용 [hide]**

- 1 품모
- 2 용도
- 3 삼 또한보십시오
- 4 참고 문헌
- 5 외부 링크

**기능 [ 편집 ]**

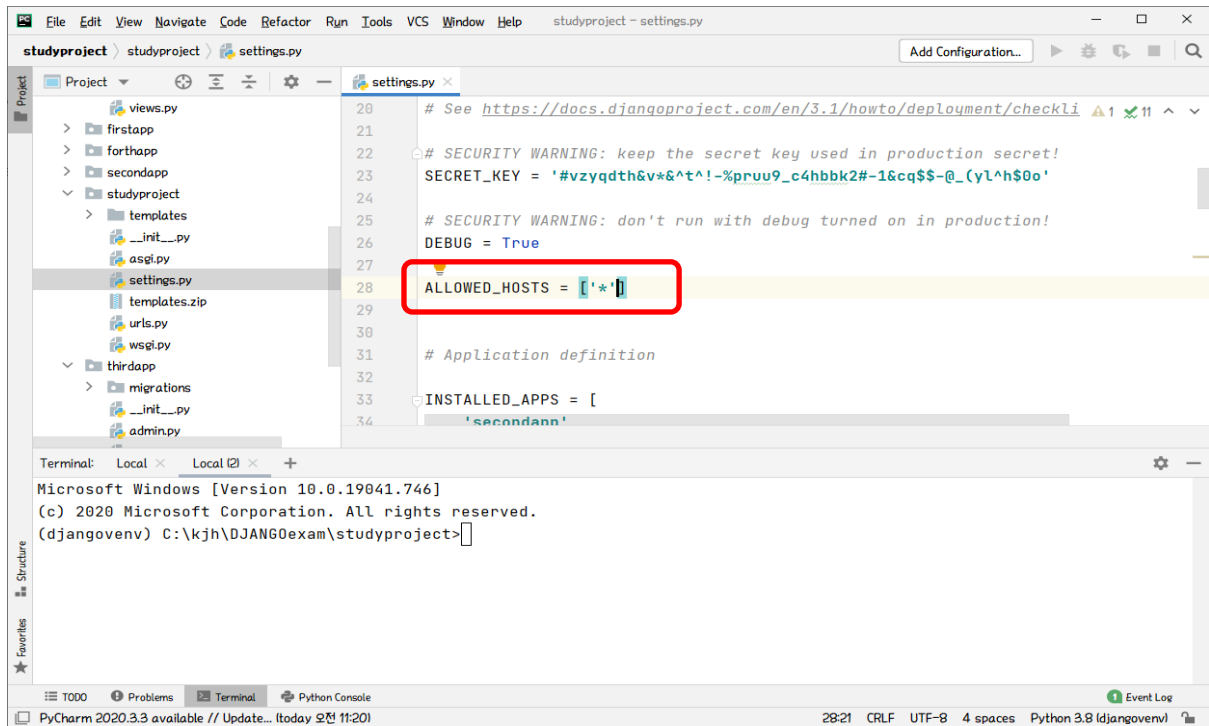
- Python 버전 2.7, 3.3, 3.4, 3.5, 3.6, 3.7 및 3.8을 포함한 CPython , PyPy 및 IPython 지원. [5]
- 호스팅 된 서버에서 실행되는 코드가있는 브라우저 내 대화 형 콘솔로 여러 사용자가 공유 할 수 있습니다.
- WSGI 기반 웹 호스팅, 예 : Django , Flask , web2py
- iPad 및 기타 모바일 장치 에서 코딩을 지원 합니다.
- 구문 강조 표시 브라우저 내 편집기.
- 많은 인기있는 Python 모듈이 사전 설치되어 있습니다. [6]
- 주어진 시간에 스크립트를 실행하기위한 크론 과 유사한 예약 작업.
- 스크립트를 실행하고 실패시 자동으로 다시 시작하는 상시 작동 작업. [7] [8]

**용도 [ 편집 ]**

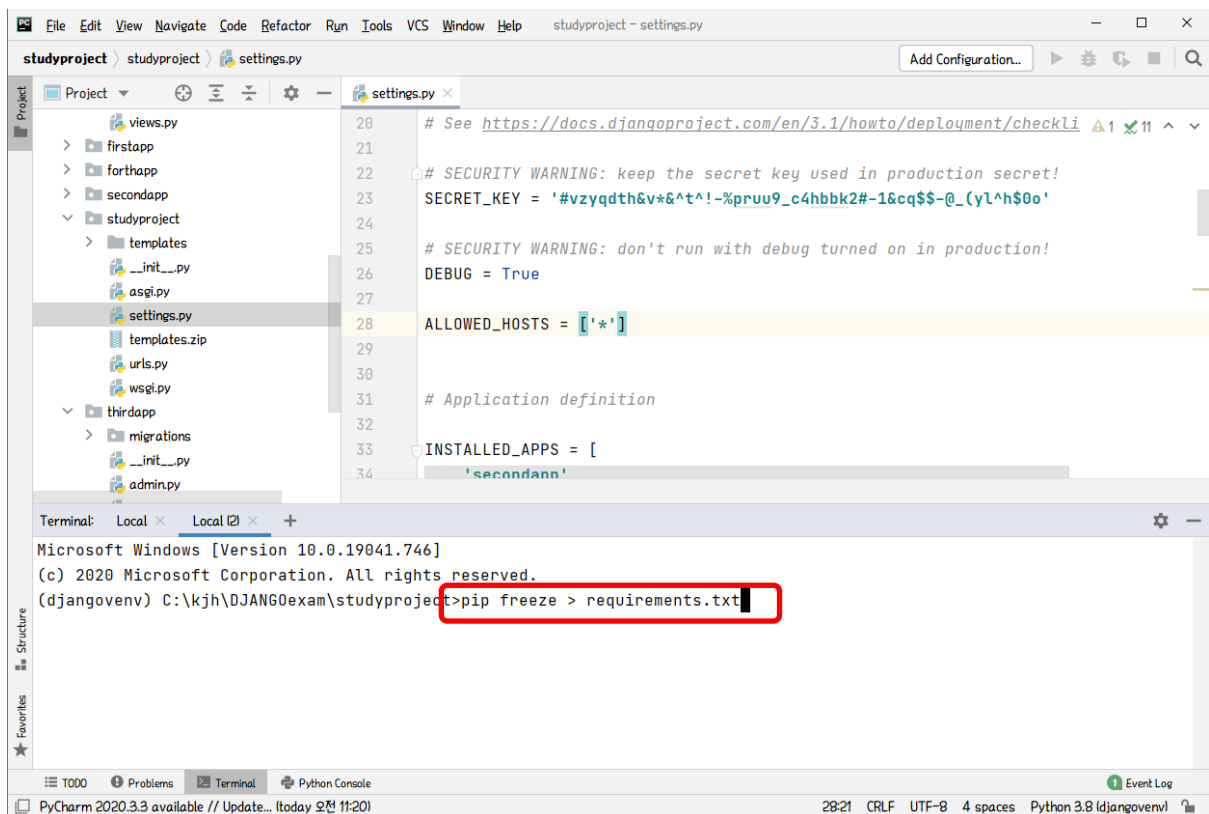
PythonAnywhere는 웹 프레임 워크에 대한 공식 책에서 " web2py 애플리케이션 을 배포하는 가장 간단한 방법"으로 설명되어 있

pythonanywhere 사이트에 올리기(배포) 위해서는 다음에 제시된 과정을 진행한다.

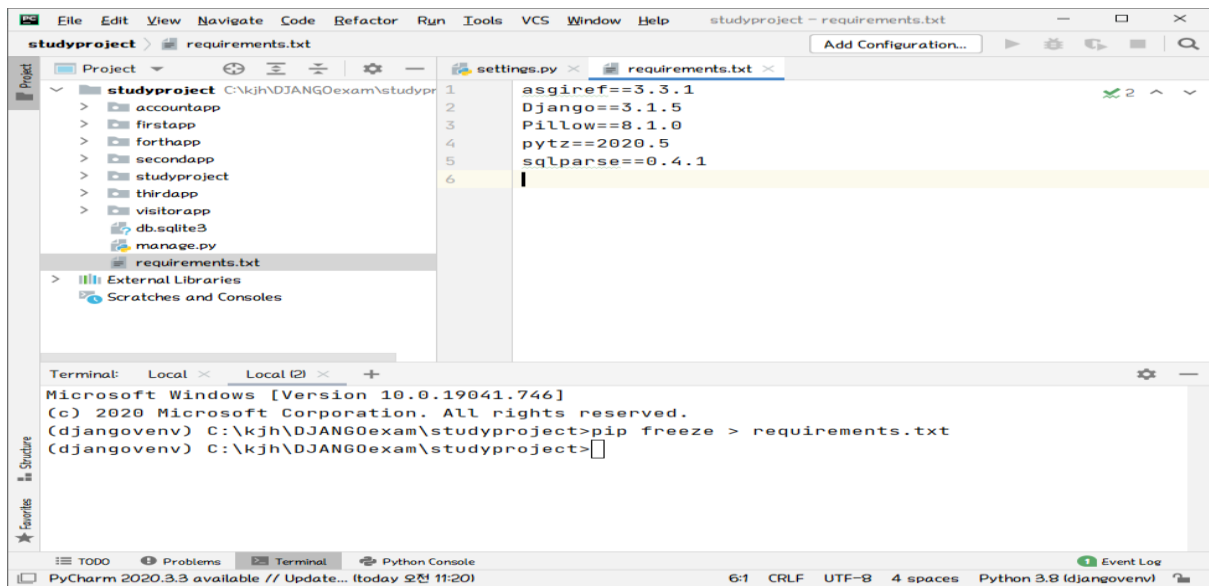
- (1) 먼저 프로젝트폴더명/settings.py 파일에서 ALLOWED\_HOSTS 변수의 값을 [] 에서 ['\*'] 로 변경한다.  
([]은 localhost 로만 접근 가능한 웹 애플리케이션을 생성하게 된다.)



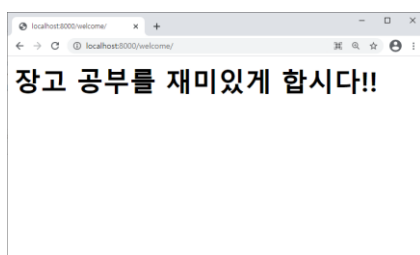
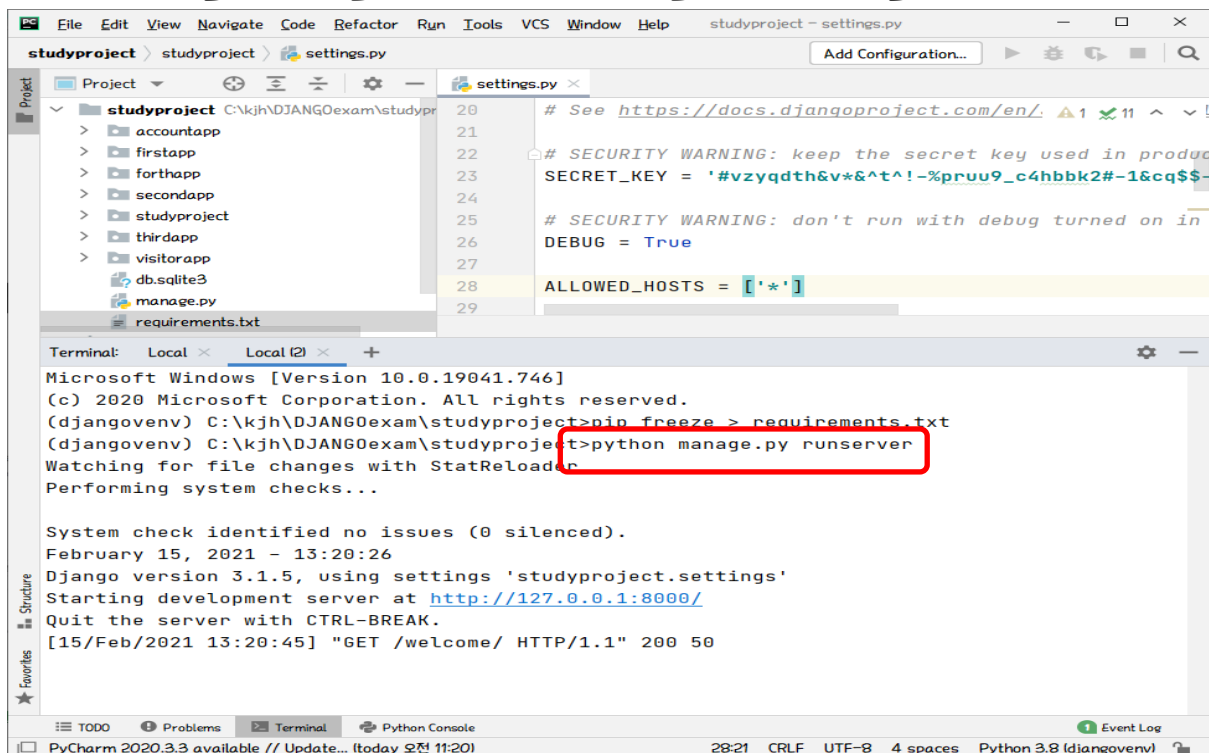
- (2) 터미널 창에서 다음 명령을 수행시켜서 설치된 패키지 정보를 저장하는 requirements.txt 를 생성한다.



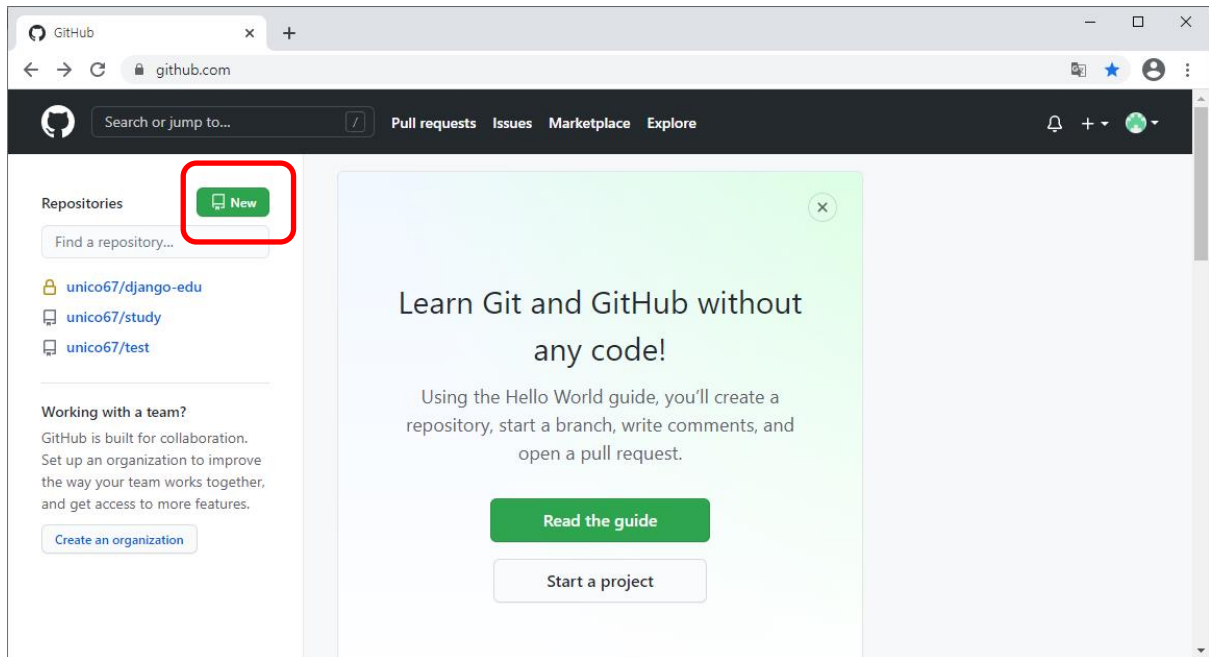
그러면 다음과 같은 내용으로 구성되는 requirements.txt 파일이 생성된다.



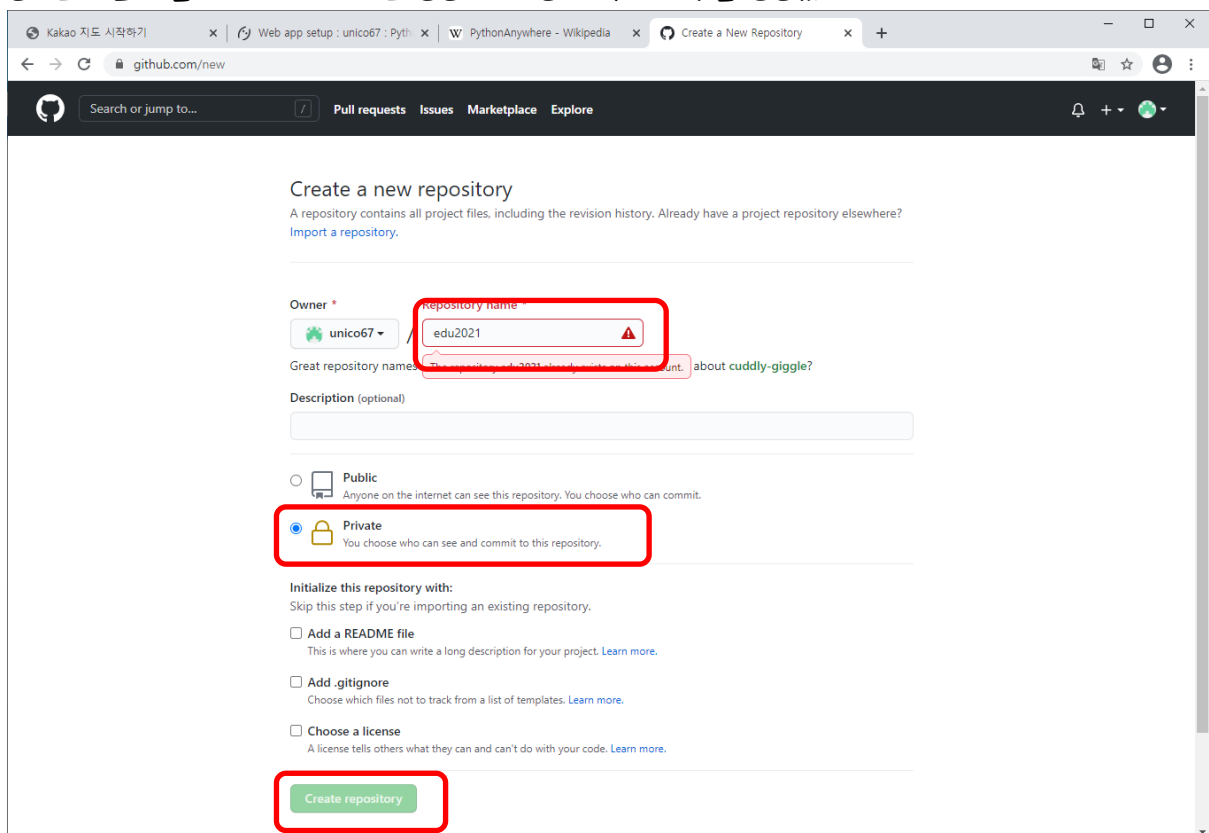
그런 다음 다음과 같이 장고서버를 기동시키고 웹 페이지 하나를 요청해서 제대로 결과가 나오는지 확인한다.



(3) 깃허브에 로그인 하고 새로운 저장소를 한 개 만든다.

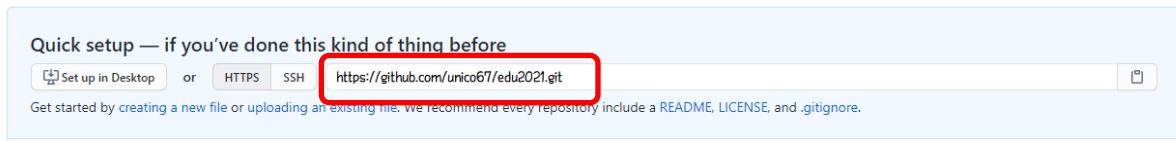


강사는 다음과 같이 edu2021 이라는 명칭으로 저장소(repository)를 생성했다.

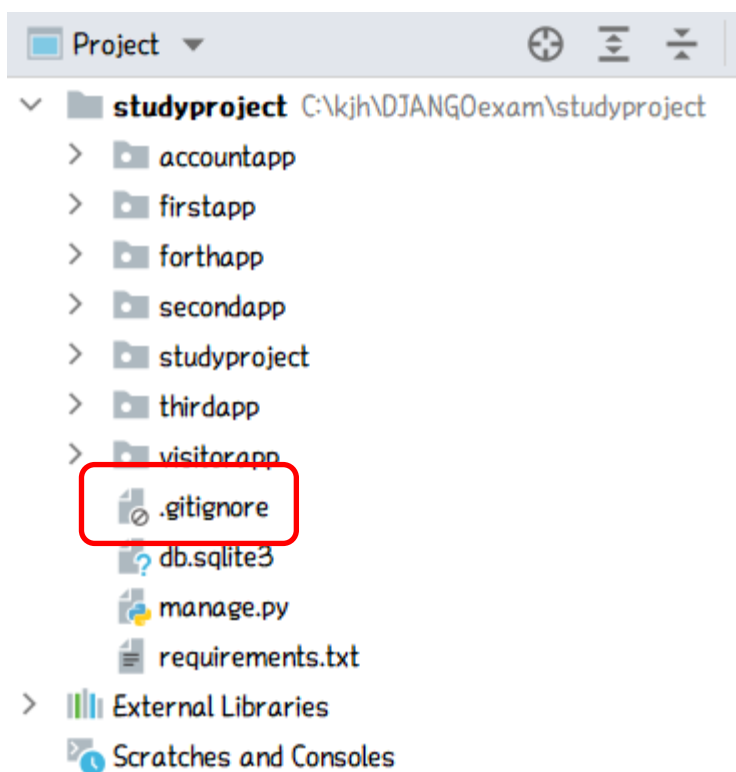
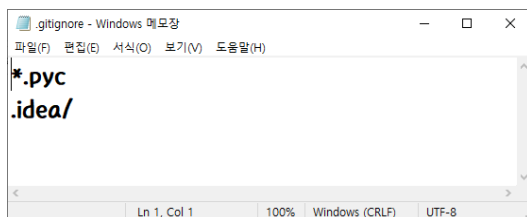


다음 화면에 출력된 URL 을 메모장에 보관한다.

<https://github.com/unico67/edu2021.git>



(4) 깃허브에 새로이 생성한 저장소(강사는 edu2021)에 프로젝트 소스를 업로드하기 위해서 다음과 같은 내용으로 .gitignore 라는 파일을 프로젝트 폴더에 생성한다. 이 파일의 용도는 업로드하지 않으려는 파일들에 대한 정보를 작성하는 것이다.



(5) 명령 프롬프트(cmd) 창을 기동시키고 다음 명령들을 수행시킨다.

다음 명령을 실행시켜서 프로젝트 폴더로 이동한다.

(cd c:\kjh\DJANGOexam\studypoint)

```
C:\kjh\DJANGOexam\studypoint>git init
Initialized empty Git repository in C:\kjh\DJANGOexam\studypoint\.git/

C:\kjh\DJANGOexam\studypoint>git add .

C:\kjh\DJANGOexam\studypoint>git commit -m "project commit"
[master (root-commit) ef37534] project commit
135 files changed, 3012 insertions(+)
create mode 100644 .gitignore
create mode 100644 accountapp/__init__.py
create mode 100644 accountapp/admin.py
create mode 100644 accountapp/apps.py
create mode 100644 accountapp/migrations/__init__.py
create mode 100644 accountapp/models.py
create mode 100644 accountapp/templates/index.html
create mode 100644 accountapp/templates/login.html
create mode 100644 accountapp/templates/logout.html
create mode 100644 accountapp/templates/member.html
create mode 100644 accountapp/templates/register.html
create mode 100644 accountapp/tests.py
create mode 100644 accountapp/urls.py
create mode 100644 accountapp/views.py
create mode 100644 db.sqlite3
create mode 100644 firstapp/__init__.py
create mode 100644 firstapp/admin.py
create mode 100644 firstapp/apps.py
create mode 100644 firstapp/migrations/__init__.py
create mode 100644 firstapp/models.py
create mode 100644 firstapp/tests.py
create mode 100644 firstapp/views.py
create mode 100644 forthapp/__init__.py
create mode 100644 forthapp/admin.py
create mode 100644 forthapp/apps.py
create mode 100644 forthapp/migrations/0001_initial.py
create mode 100644 forthapp/migrations/__init__.py
create mode 100644 forthapp/models.py
create mode 100644 forthapp/templates/c.html
```

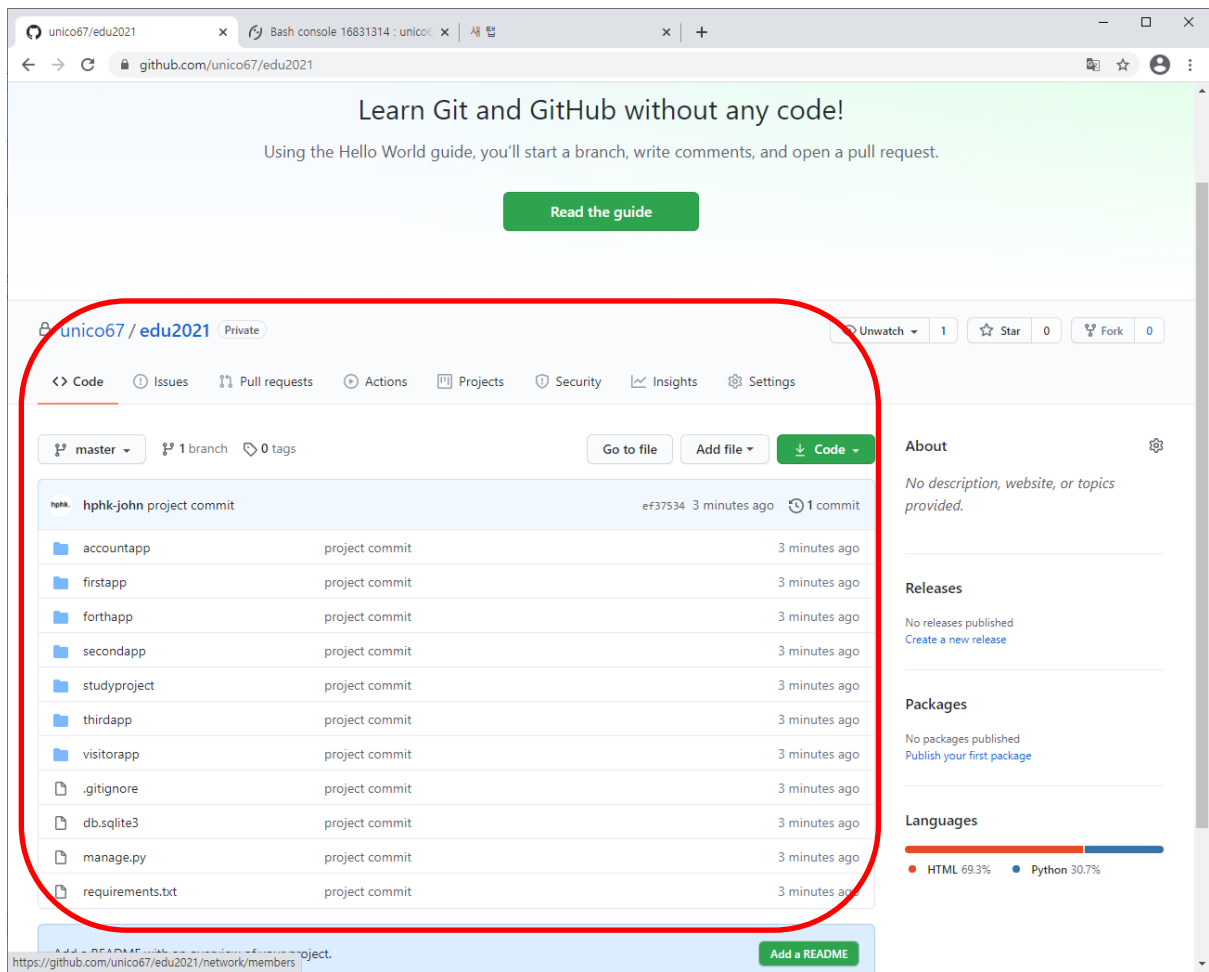
```
C:\kjh\DJANGOexam\studypoint>git remote add origin https://github.com/unico67/edu2021.git

C:\kjh\DJANGOexam\studypoint>git push -u origin master
Enumerating objects: 133, done.
Counting objects: 100% (133/133), done.
Delta compression using up to 8 threads
Compressing objects: 100% (127/127), done.
Writing objects: 100% (133/133), 607.24 KiB | 12.65 MiB/s, done.
Total 133 (delta 21), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (21/21), done.
To https://github.com/unico67/edu2021.git
 * [new branch]      master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.

C:\kjh\DJANGOexam\studypoint>
```

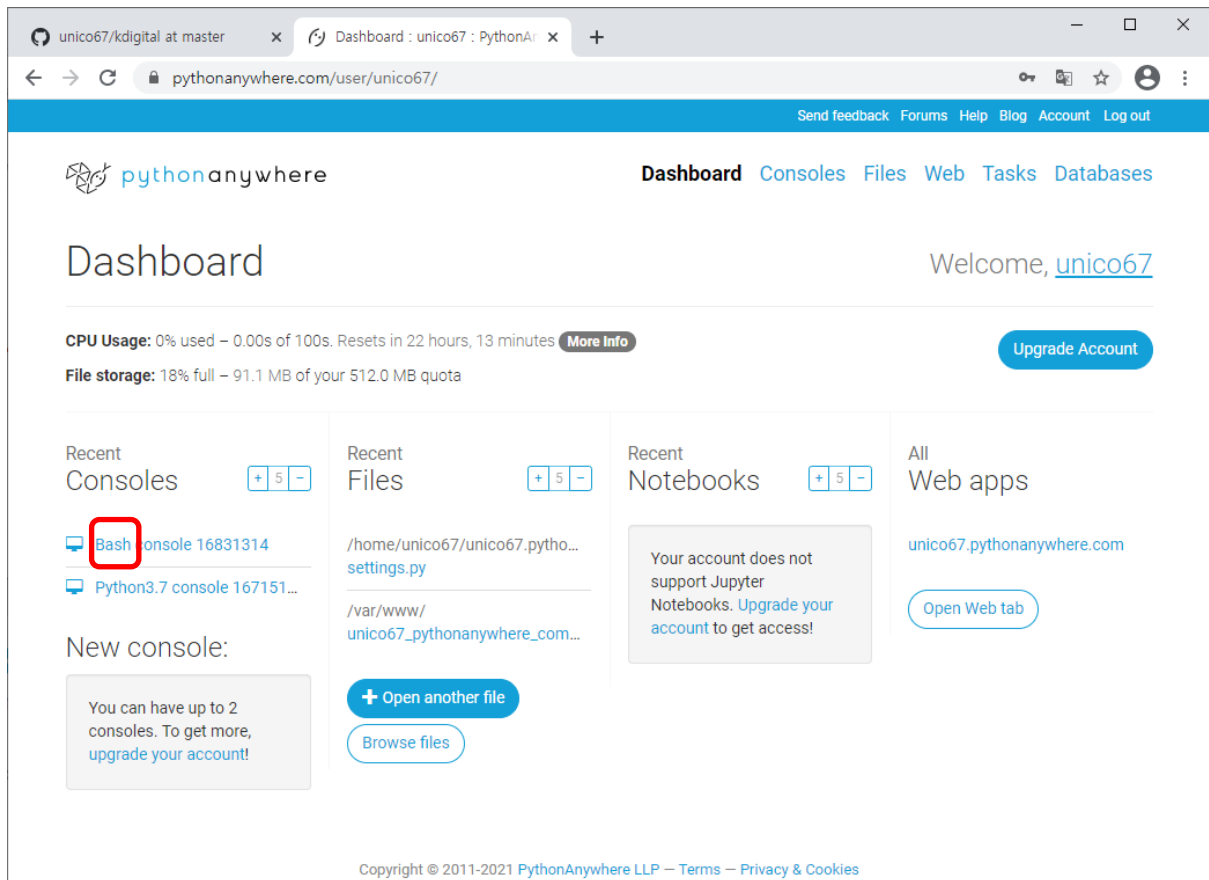
본인꺼로

다음과 같이 edu2021 저장소에 파일들이 저장된 것을 확인할 수 있다.

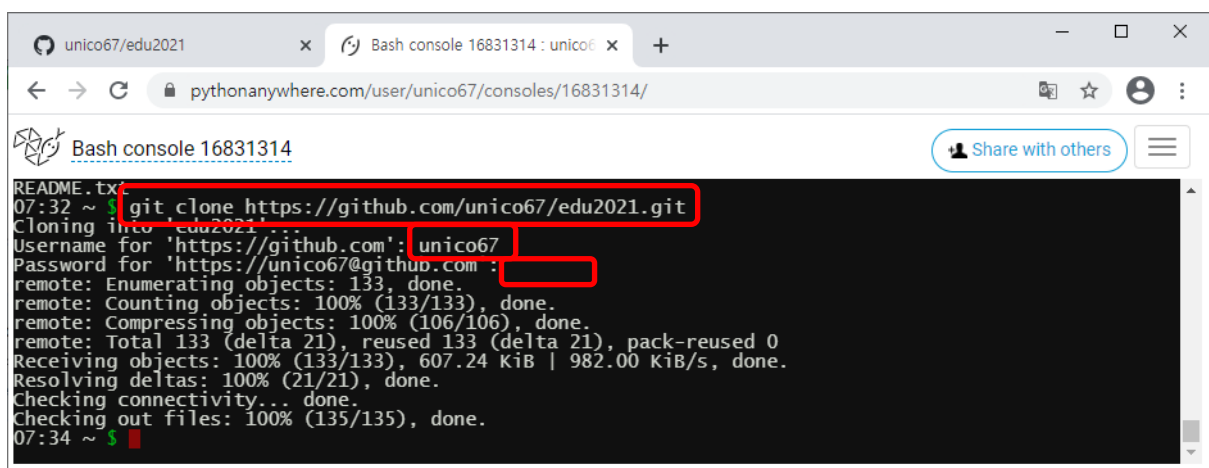


여러분들은 이미 깃허브를 사용하고 있었으므로 (3),(4),(5)은 스킵해도 될거 같네용.....

(6) pythonanywhere 사이트에 회원 가입해서 계정을 하나 만든다. Dashboard 화면에서 Bash를 클릭하여 명령프롬프트 창과 비슷한 화면을 기동시킨다.



다음 명령을 실행시켜서 깃허브에 있는 edu2021 저장소의 파일들을 복제해 온다.





pythonanywhere.com/user/unico67/consoles/16831314/

Bash console 16831314

Checking out files: 100% (135/135), done.

```
07:34 ~ $ ls
README.txt edu2021
07:35 ~ $ tree edu2021
edu2021
├── accountapp
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── templates
│   │   ├── index.html
│   │   ├── login.html
│   │   ├── logout.html
│   │   ├── member.html
│   │   └── register.html
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── db.sqlite3
├── firstapp
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations
│   │   └── __init__.py
│   ├── models.py
│   ├── tests.py
│   └── views.py
├── forthapp
│   ├── __init__.py
│   ├── admin.py
│   ├── apps.py
│   ├── migrations
│   │   ├── 0001_initial.py
│   │   └── __init__.py
│   ├── models.py
│   ├── templates
│   │   ├── c.html
│   │   ├── d.html
│   │   ├── r.html
│   │   └── u.html
│   ├── tests.py
│   ├── urls.py
│   └── views.py
├── manage.py
├── requirements.txt
├── secondapp
│   └── __init__.py
```

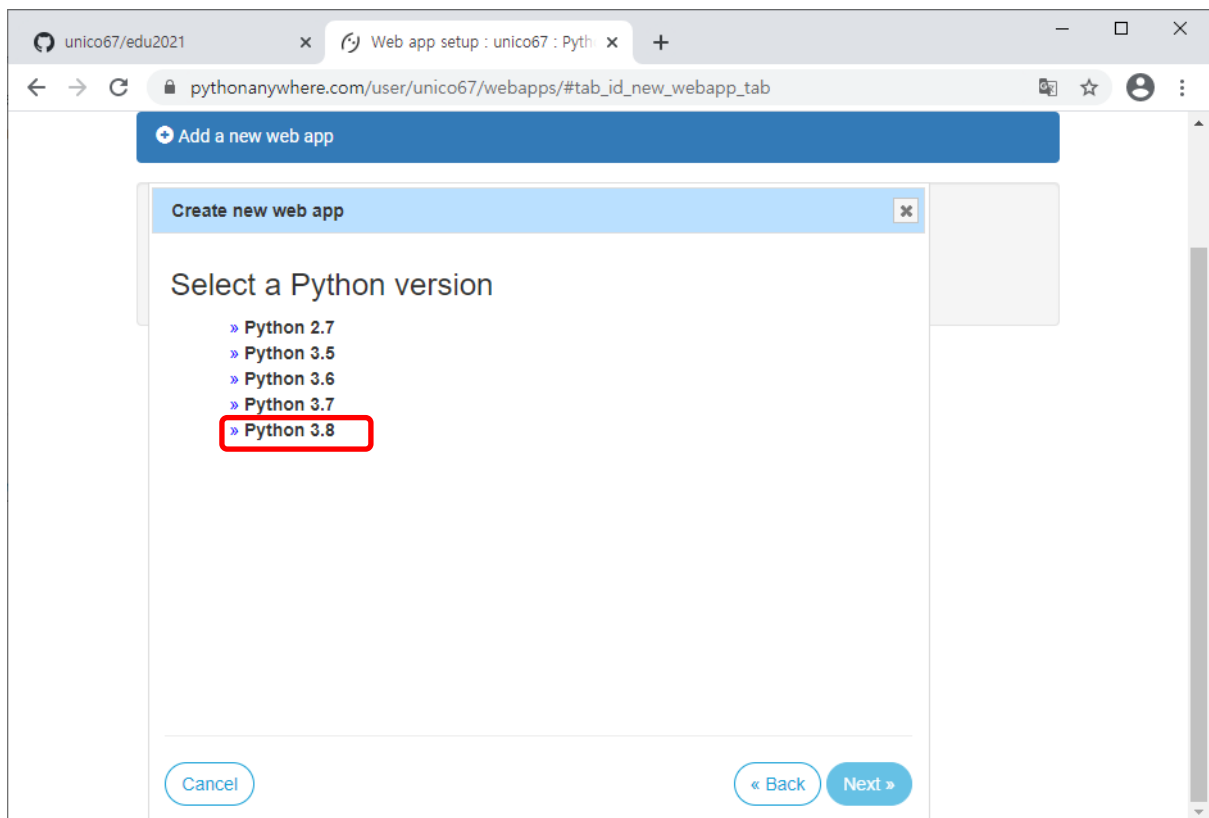
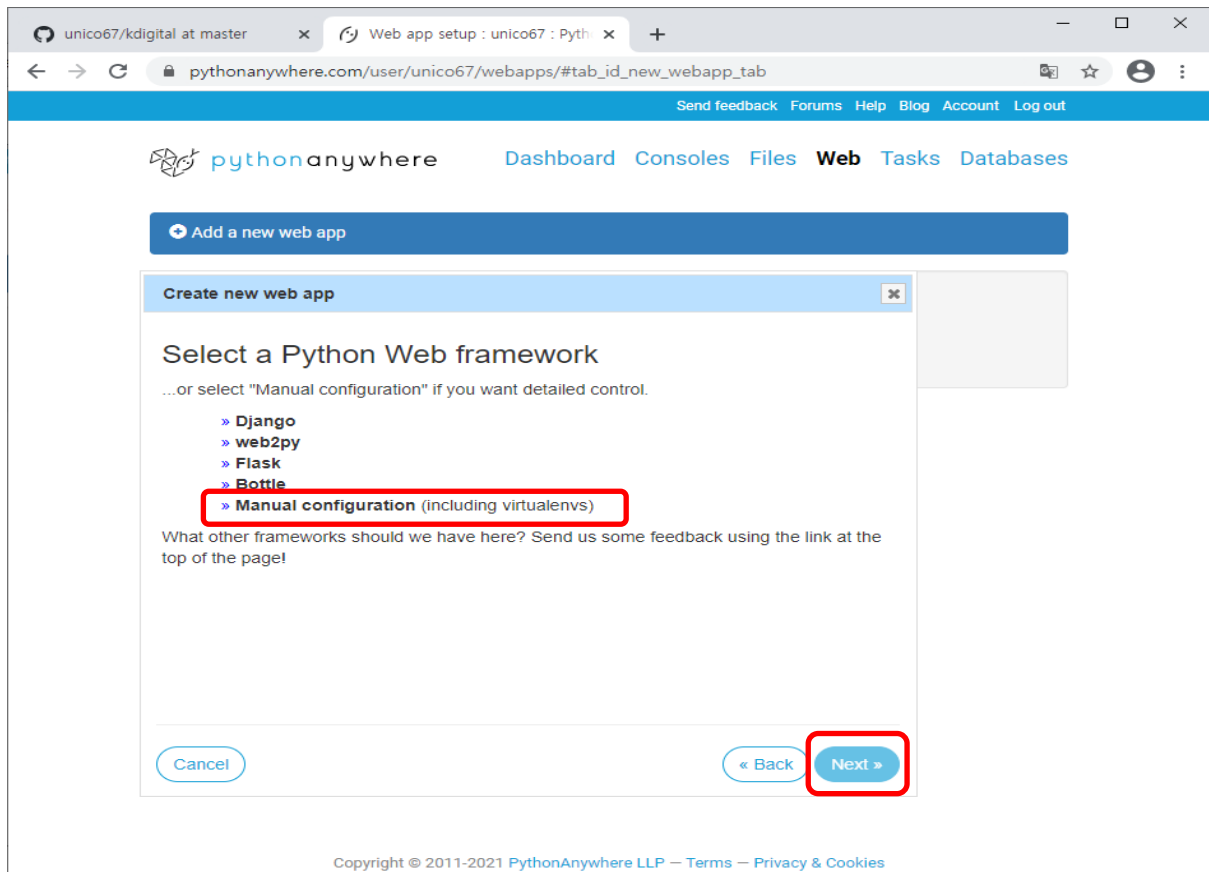
(7) pythonanywhere 사이트에 새로운 웹앱을 생성한다.

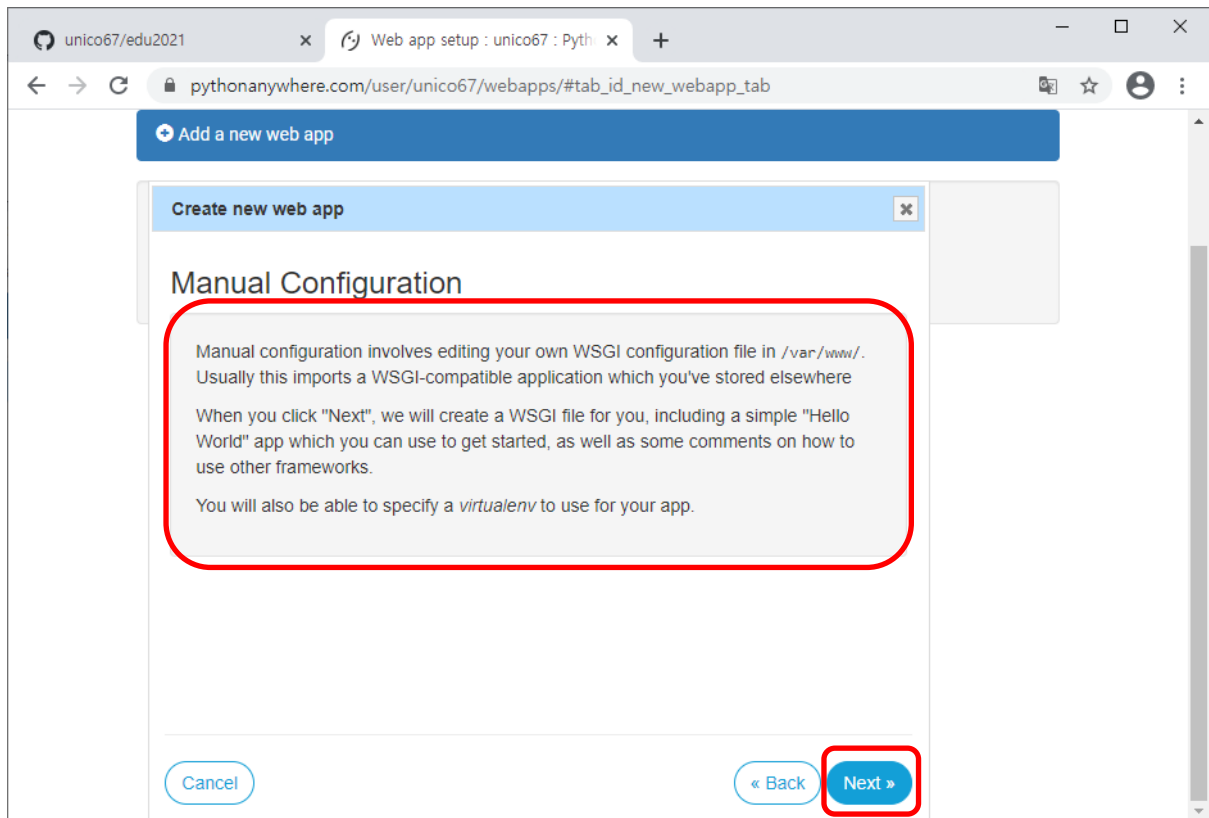
The image consists of two screenshots of the PythonAnywhere website, illustrating the steps to create a new web application.

**Top Screenshot:** The user is logged in as 'unico67/kdigital at master'. The page title is 'Web app setup : unico67 : PythonAnywhere'. The URL is 'pythonanywhere.com/user/unico67/webapps/#tab\_id\_new\_webapp\_tab'. The navigation bar includes links for 'Send feedback', 'Forums', 'Help', 'Blog', 'Account', and 'Log out'. The main content area shows the 'pythonanywhere' logo and a navigation menu with 'Dashboard', 'Consoles', 'Files', 'Web', 'Tasks', and 'Databases'. A blue button labeled 'Add a new web app' is highlighted with a red rectangle. Below this button, a message states: 'You have no web apps. To create a PythonAnywhere-hosted web app, click the "Add a new web app" button to the left.'

**Bottom Screenshot:** The user is still logged in. The 'Add a new web app' button is now a modal window titled 'Create new web app'. The modal contains the text: 'Your web app's domain name. Your account doesn't support custom domain names, so your PythonAnywhere web app will live at unico67.pythonanywhere.com. Want to change that? [Upgrade now!](#) Otherwise, just click "Next" to continue.' The 'Next' button at the bottom right of the modal is highlighted with a red rectangle. The 'Cancel' button is at the bottom left, and the 'Back' button is in the middle.

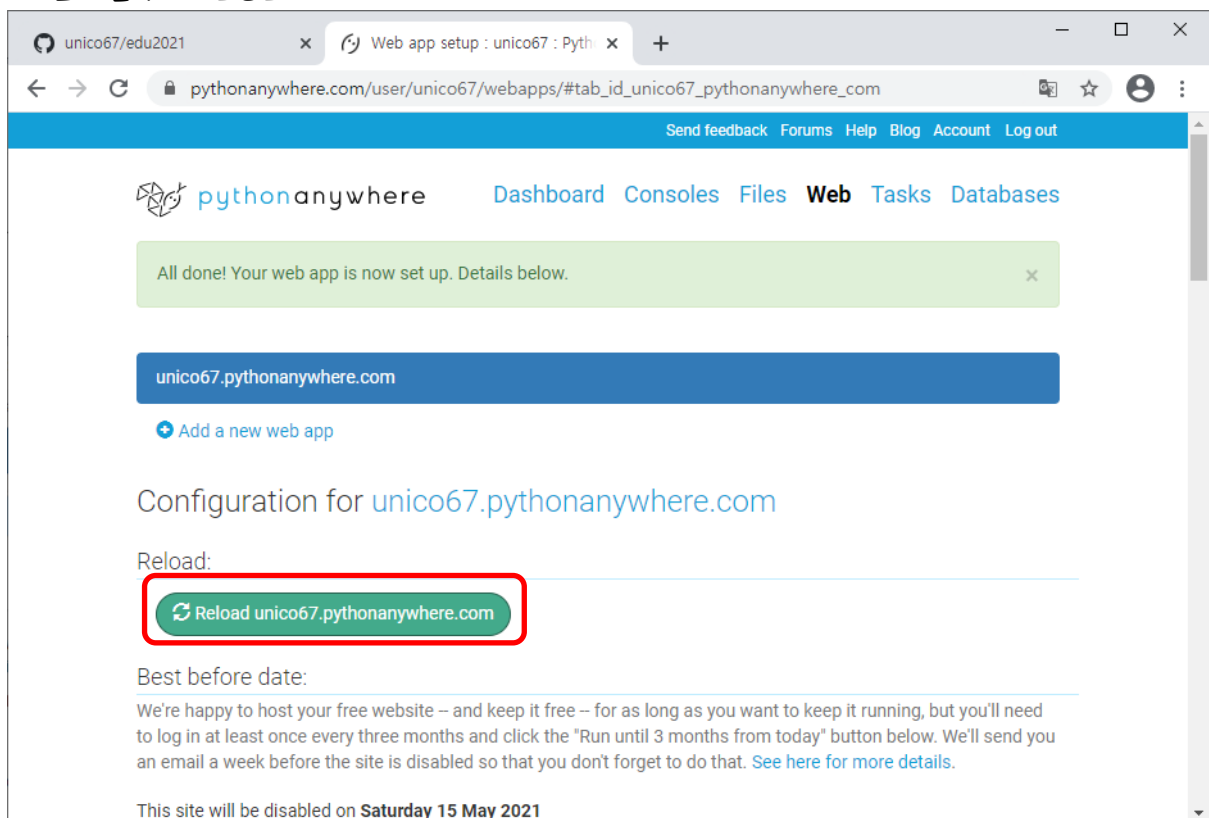
Copyright © 2011-2021 PythonAnywhere LLP — Terms — Privacy & Cookies





다음과 같은 웹앱 설정화면이 출력된다.

앞으로의 작업은 대부분 이 화면에서 환경 설정을 처리하게 된다. 수정을 모두 끝낸 후에는 Reload..... 버튼을 시스템에 클릭하여 반영한다.



(8) 복제해온 장고프로젝트를 위한 가상환경을 생성하고 여기에 django 캐피지를 설치한다.

```
08:10 ~ $ mkvirtualenv --python=python3.8 unico67.pythonanywhere.com
Running virtualenv with interpreter /usr/bin/python3.8
Already using interpreter /usr/bin/python3.8
Using base prefix '/usr'
New python executable in /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/python3.8
Also creating executable in /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/python
Installing setuptools, pip, wheel...
done.
virtualenvwrapper.user_scripts creating /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/pr
edeactivate
virtualenvwrapper.user_scripts creating /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/po
stdeactivate
virtualenvwrapper.user_scripts creating /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/pr
eactivate
virtualenvwrapper.user_scripts creating /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/po
stactivate
virtualenvwrapper.user_scripts creating /home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/ge
t_env_details
(unico67.pythonanywhere.com) 08:13 ~ $ deactivate
08:13 ~ $ which python
/usr/bin/python
08:13 ~ $ workon unico67.pythonanywhere.com
(unico67.pythonanywhere.com) 08:13 ~ $ which python
/home/unico67/.virtualenvs/unico67.pythonanywhere.com/bin/python
(unico67.pythonanywhere.com) 08:14 ~ $ pip install django
Looking in links: /usr/share/pip-wheels
Collecting django
  Using cached Django-3.1.6-py3-none-any.whl (7.8 MB)
Collecting sqlparse>=0.2.2
  Using cached sqlparse-0.4.1-py3-none-any.whl (42 kB)
Collecting pytz
  Using cached pytz-2021.1-py2.py3-none-any.whl (510 kB)
Collecting asgiref<4,>=3.2.10
  Using cached asgiref-3.3.1-py3-none-any.whl (19 kB)
Installing collected packages: sqlparse, pytz, asgiref, django
Successfully installed asgiref-3.3.1 django-3.1.6 pytz-2021.1 sqlparse-0.4.1
(unico67.pythonanywhere.com) 08:14 ~ $
```

웹앱 화면에서 Virtualenv 항목에 가서 다음과 같이 작성하여 가상환경 패스를 등록한다.

pythonanywhere.com/user/unico67/webapps/#tab\_id\_unico67\_pythonanywhere\_com

Enter the path to your web app source code

Working directory:

[/home/unico67/](#)  
Go to directory

WSGI configuration file:

[/var/www/unico67.pythonanywhere.com/wsgi.py](#)

Python version:

3.8

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/unico67/.virtualenvs/unico67.pythonanywhere.com](#)

[Start a console in this virtualenv](#)

Log files:

The first place to look if something goes wrong.

[unico67.pythonanywhere.com/access.log](#)

[unico67.pythonanywhere.com/error.log](#)

[unico67.pythonanywhere.com/server.log](#)

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

Static files:

Files that aren't dynamically generated by your code, like CSS, JavaScript or uploaded files, can be served much

(9) 소스들이 존재하는 폴더를 설정하기 위해서 Code 항목에 있는 Enter the..... 링크를 클릭한다.

pythonanywhere.com/user/unico67/webapps/#tab\_id\_unico67\_pythonanywhere\_com

Today (yesterday) 2 (0)  
Hour (previous hour) 0 (2)

Want some more data? [Paying accounts](#) get pretty charts ;-)

Code:  
What your site is running.

[Enter the path to your web app source code](#)

[/home/unico67/](#)  
[Go to directory](#)

[/var/www/unico67\\_pythonanywhere\\_com\\_wsgi.py](#)

3.8

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/unico67/.virtualenvs/unico67.pythonanywhere.com](#)  
[Start a console in this virtualenv](#)

Log files:  
The first place to look if something goes wrong.

Access log:

다음과 같이 소스 폴더의 패스를 작성한다.

pythonanywhere.com/user/unico67/webapps/#tab\_id\_unico67\_pythonanywhere\_com

Today (yesterday) 2 (0)  
Hour (previous hour) 0 (2)

Want some more data? [Paying accounts](#) get pretty charts ;-)

Code:  
What your site is running.

[/home/unico67/edu2021](#)  
[Go to directory](#)

[/home/unico67/](#)  
[Go to directory](#)

[/var/www/unico67\\_pythonanywhere\\_com\\_wsgi.py](#)

3.8

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/unico67/.virtualenvs/unico67.pythonanywhere.com](#)  
[Start a console in this virtualenv](#)

Log files:  
The first place to look if something goes wrong.

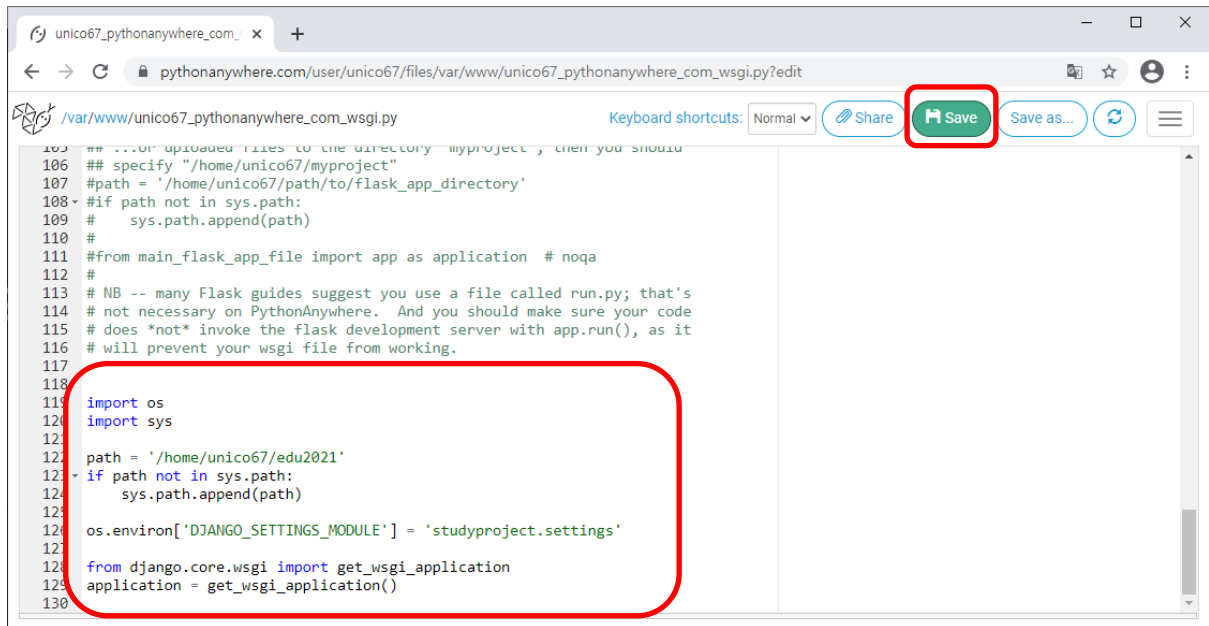
(10) WSGI configuration file 을 수정하기 위해서 다음 링크를 클릭한다.

The screenshot shows the 'Web app setup' page for a user named 'unico67'. The page displays various statistics and configuration options. A red rectangle highlights the 'WSGI configuration file' path: `/var/www/unico67_pythonanywhere_com_wsgi.py`. Other visible elements include the 'Code' section with a 'What your site is running.' message, 'Source code' and 'Working directory' links, 'Python version' (3.8), 'Virtualenv' instructions, and 'Log files' information.

다음과 같이 에디터 창에 파일 내용이 출력되는데 19행부터 47행까지를 주석으로 묶기 위해 `'''` 를 입력한다.

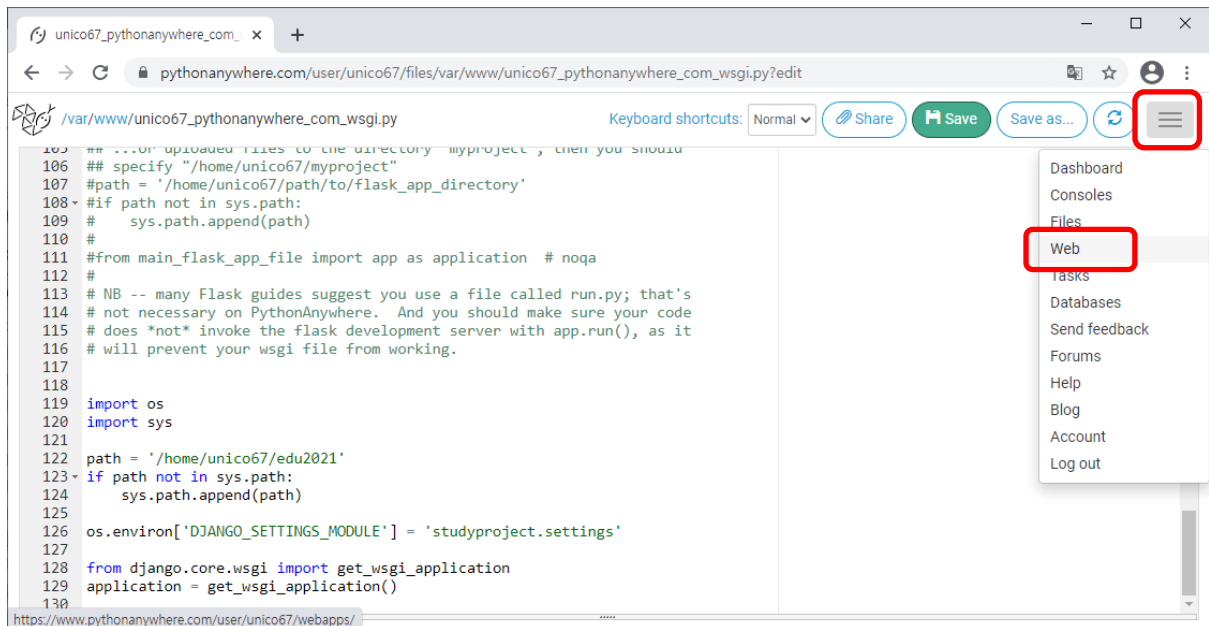
The screenshot shows the code editor for the file `/var/www/unico67_pythonanywhere_com_wsgi.py`. The code is a Python WSGI application. A red rectangle highlights the opening triple quote `'''` at line 17, and another red rectangle highlights the closing triple quote `'''` at line 47. The code between these quotes is a simple HTTP response template. The rest of the code defines the `application` function and includes comments about Django and Flask templates.

마지막 부분에 다음에 제시된 내용을 작성하고 Save 버튼을 클릭한다.



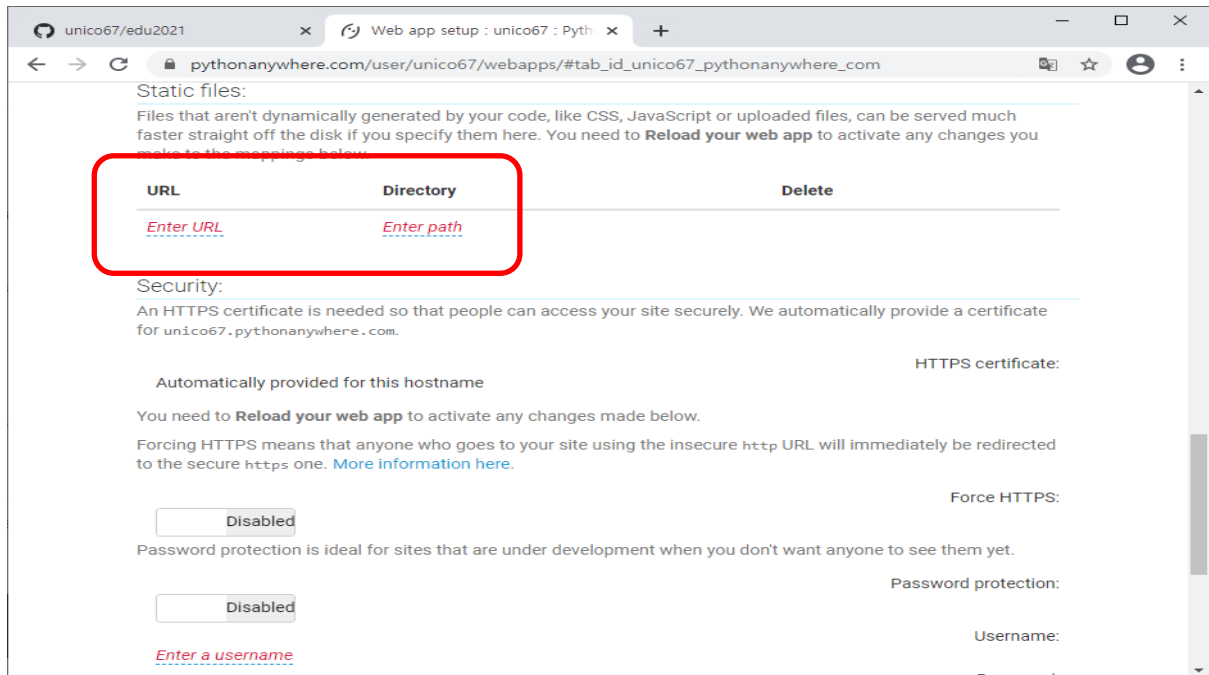
```
105 ## ...or uploaded files to the directory myproject, then you should
106 ## specify "/home/unico67/myproject"
107 #path = '/home/unico67/path/to/flask_app_directory'
108 #if path not in sys.path:
109 #    sys.path.append(path)
110 #
111 #from main_flask_app_file import app as application # noqa
112 #
113 # NB -- many Flask guides suggest you use a file called run.py; that's
114 # not necessary on PythonAnywhere. And you should make sure your code
115 # does *not* invoke the flask development server with app.run(), as it
116 # will prevent your wsgi file from working.
117
118
119 import os
120 import sys
121
122 path = '/home/unico67/edu2021'
123 if path not in sys.path:
124     sys.path.append(path)
125
126 os.environ['DJANGO_SETTINGS_MODULE'] = 'studyproject.settings'
127
128 from django.core.wsgi import get_wsgi_application
129 application = get_wsgi_application()
130
```

Save 버튼을 클릭한 다음에는 다음과 같이 버튼과 메뉴를 클릭하여 웹 화면으로 나간다.

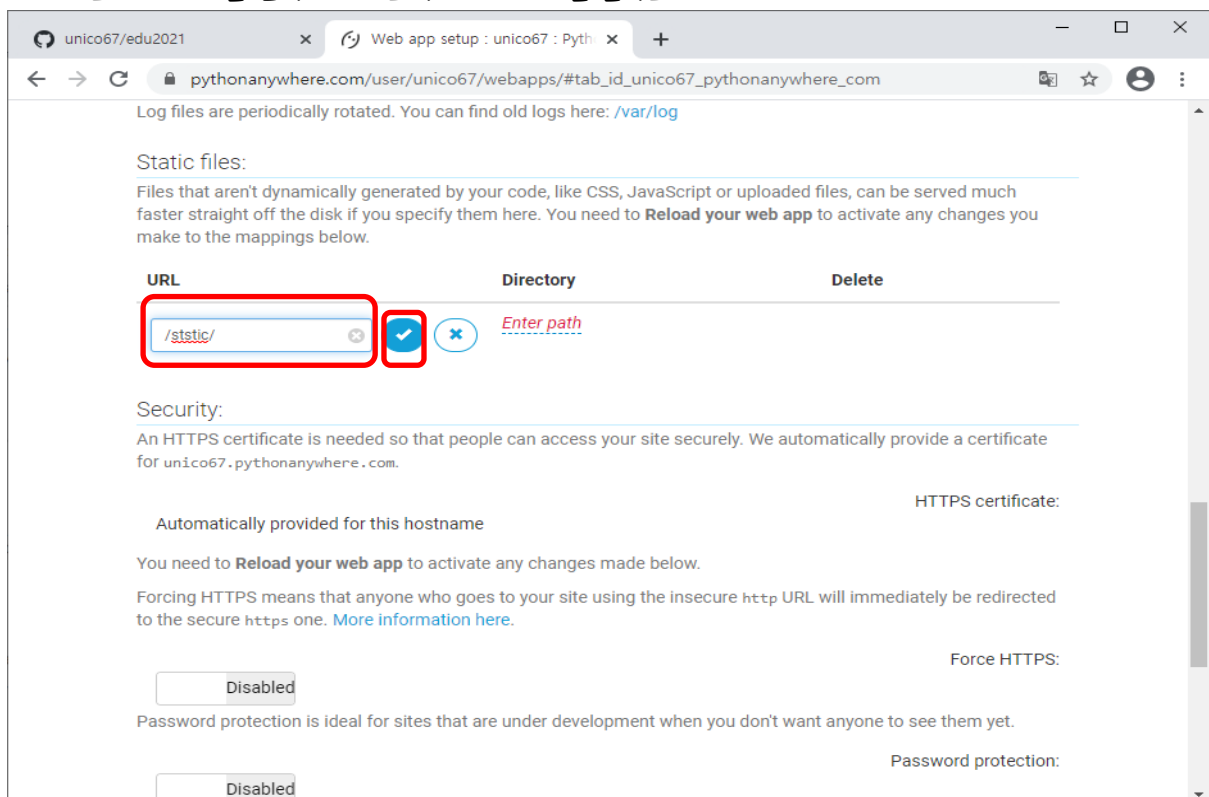




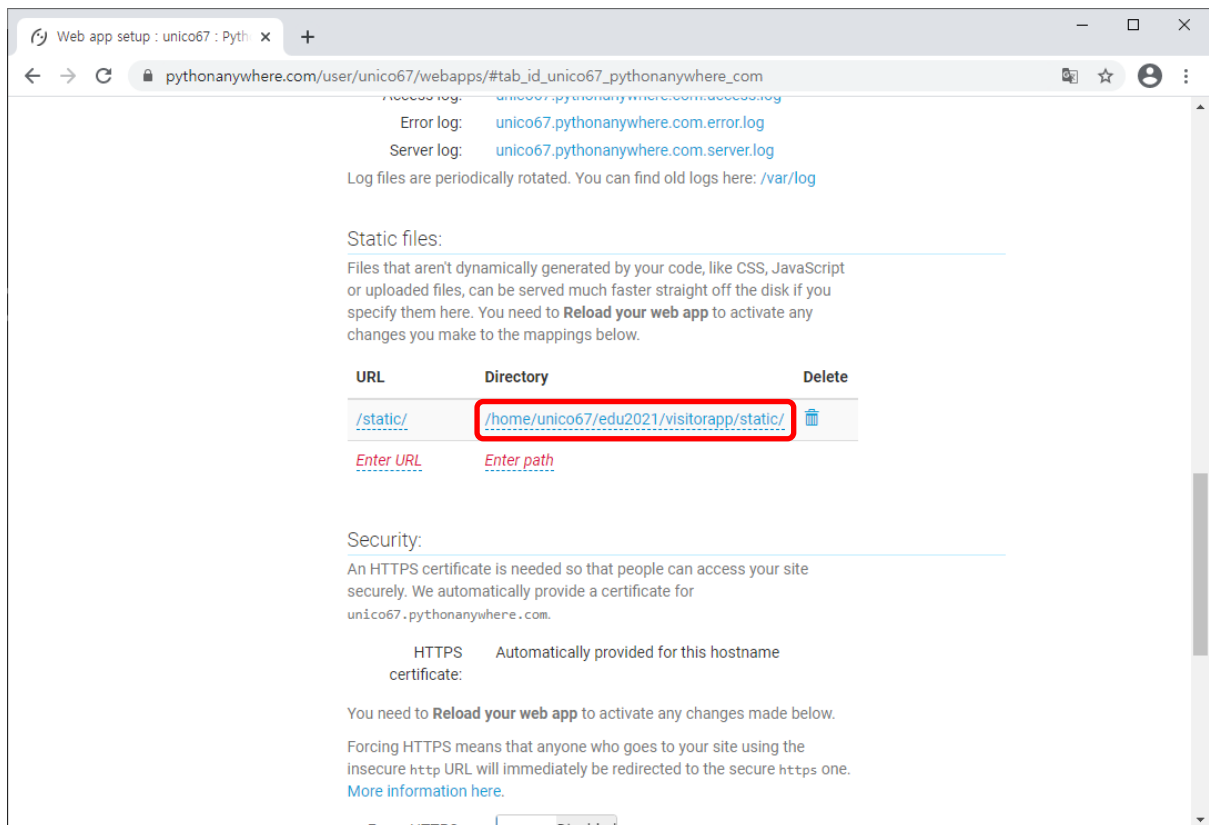
(11) Static files 를 설정하여 ststic 자원들이 존재하는 디렉토리 정보를 조정한다.



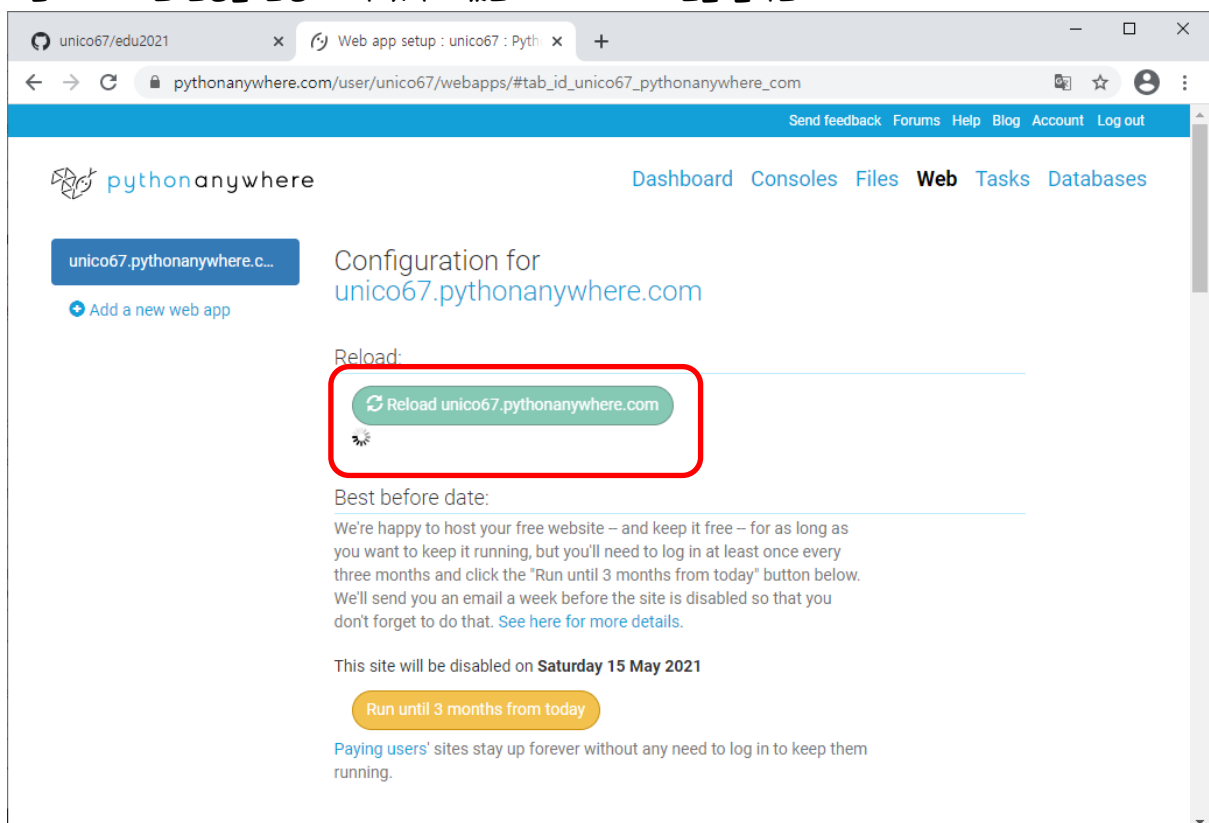
URL 에는 /static/ 을 입력하고 파란색 체크 이미지를 클릭한다.



Directory 에는 static 폴더에 대한 패스를 입력하고 파란색 채크 이미지를 클릭한다.



지금까지의 모든 변경을 반영하도록 왼쪽에 있는 Reload..... 버튼을 클릭한다.



부족한 설명은 다음 사이트 참고

[https://tutorial-extensions.djangogirls.org/ko/manual\\_pythonanywhere\\_deploy/](https://tutorial-extensions.djangogirls.org/ko/manual_pythonanywhere_deploy/)

static 자원 collect 하는 것과 DB migrate 하는 것은 생략함....