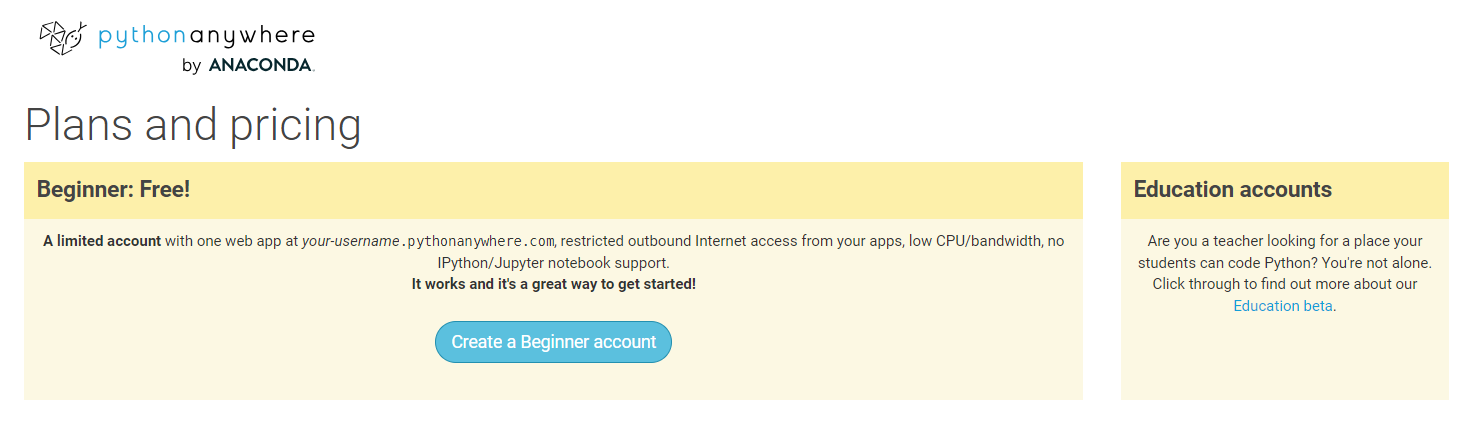
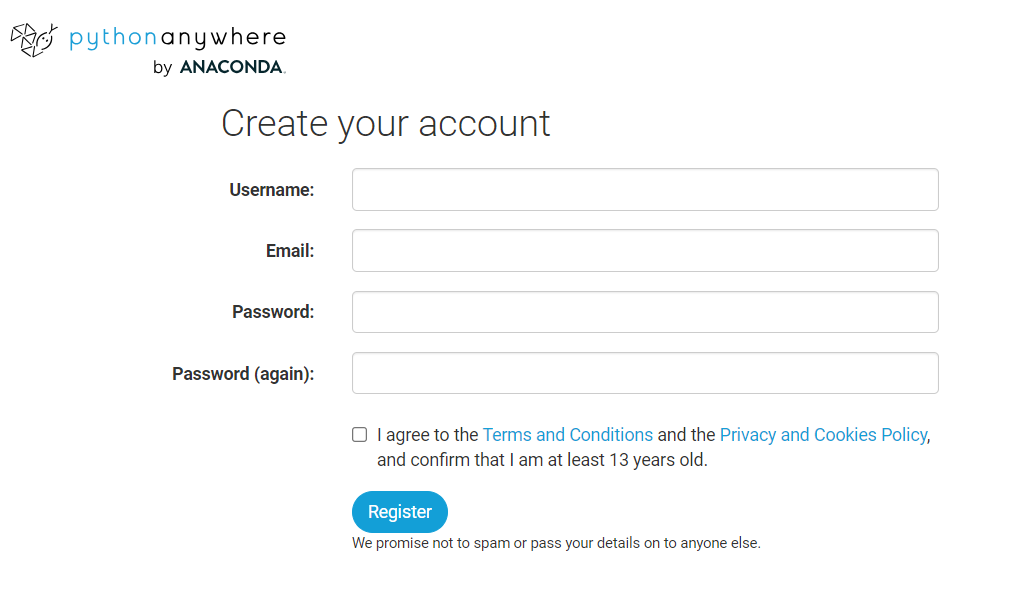
1. Регистрируем почту. В нашем случае регистрируем на proton.me
2. Регистрируем аккаунт на pythonanywhere.com. PythonAnywhere — это сервис по запуску кода на Python в облаке. Мы будем использовать его, чтобы разместить наш сайт «вживую» в интернете.
3. Регистрируем бесплатный план. Для наших задач этого вполне достаточно!



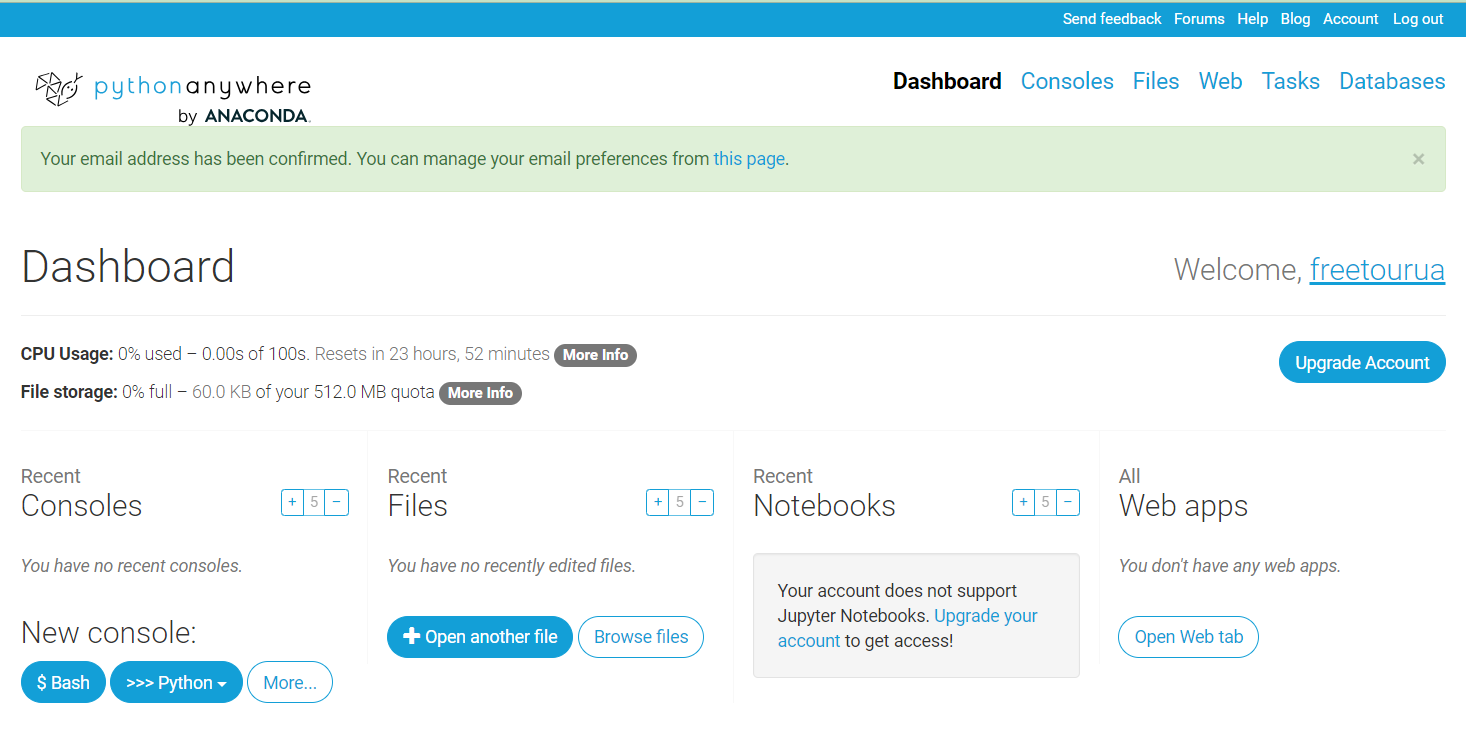
1. Заполняем регистрационную форму.
2. После подтверждения аккаунта посредством отправки электронного письма Вы окажетесь в Dashboard

До настоящего момента твой сайт был доступен только для локального просмотра, теперь же ты узнаешь, как развернуть его на удалённом сервере! Развертывание (deploy) — это процесс публикации приложения в интернете, чтобы люди могли наконец увидеть твое творение :)

Как ты уже знаешь, веб-сайт должен располагаться на сервере. Есть много компаний, предоставляющих сервера в интернете. Мы воспользуемся услугами одной из них, с довольно простым процессом публикации: PythonAnywhere. PythonAnywhere бесплатен для маленьких приложений с небольшим числом посетителей, и этого будет для нас более чем достаточно.

Другим внешним сервисом, которым мы воспользуемся, будет GitHub — сервис хостинга кода. Существуют и другие похожие сервисы, но практически у каждого программиста есть GitHub аккаунт, теперь будет и у тебя!

В итоге твой код будет в трёх местах. На локальном компьютере ты будешь заниматься разработкой и тестированием. Когда результат тебя устроит, ты загрузишь свою программу на GitHub. А твой сайт будет на PythonAnywhere, и ты сможешь обновлять его, просто загружая новую версию кода с GitHub.

Git — это «система управления версиями», используемая множеством программистов. Эта программа отслеживает изменения, происходящие с файлами, чтобы впоследствии можно было восстановить состояние кода на нужный момент времени. Это немного похоже на функцию отслеживания изменений в Microsoft Word, но куда мощнее.

Ты можешь загрузить Git с официального сайта git-scm.com

1. **Создаём Git-репозиторий**

Git отслеживает изменения определенного набора файлов, который называется репозиторием (сокращенно "репо"). Давайте создадим такой для нашего проекта. Открой консоль и запусти эти команды в папке freetour:

*git init*

*git config --global user.name "freetour"*

*git config --global user.email* [*freetour\_ua@proton.me*](mailto:freetour_ua@proton.me)

Инициализировать git-репозиторий придется только один раз за проект (и тебе больше не придется вводить имя пользователя и адрес электронной почты).

Git будет отслеживать изменения всех файлов и каталогов в заданной директории, однако некоторые из них мы предпочли бы игнорировать. Для этого нам нужно создать файл .gitignore в корневом каталоге репозитория. Открой редактор и создай новый файл со следующим содержанием:

*.gitignore*

*\*.pyc*

*\*~*

*\_\_pycache\_\_*

*myvenv*

*db.sqlite3*

*/static*

*.DS\_Store*

И сохрани его как .gitignore в корневом каталоге " *freetour* ".

Точка в начале имени файла имеет важное значение! Если у тебя есть проблемы с созданием таких файлов (Mac не позволит создать файл с названием, начинающимся с точки, через Finder, например), тогда используй кнопку «Сохранить как» в меню своего редактора кода, это точно поможет.

Среди файлов, которые мы перечислили в .gitignore, есть db.sqlite3. Этот файл содержит твою локальную базу данных, где будут храниться твои посты. Мы не добавляем его в репозиторий, поскольку твой сайт на PythonAnywhere будет использовать другую базу данных. Эта база данных тоже может быть SQLite, как на твоём рабочем компьютере, но обычно используется MySQL, которая может справиться с большим количеством посетителей, чем SQLite. В любом случае, поскольку мы не копируем базу данных SQLite на GitHub, все посты, которые уже созданы, будут доступны только на локальном компьютере, и нам придётся заново создать их на опубликованном сайте.

Используй команду git status перед git add.

Команда git status возвращает информацию обо всех ранее неотслеживаемых/изменённых/добавленных в git файлах, а также статус ветки и многое другое.

Сохраним наши изменения

*git add .*

1. **Загружаем код в репозиторий GitHub**

Зайди на GitHub.com и создай новую бесплатную учётную запись

После чего нужно связать свой локальный репозиторий с репозиторием на GitHub и загрузить файлы проекта.

Мы не будем здесь рассматривать этот аспект.

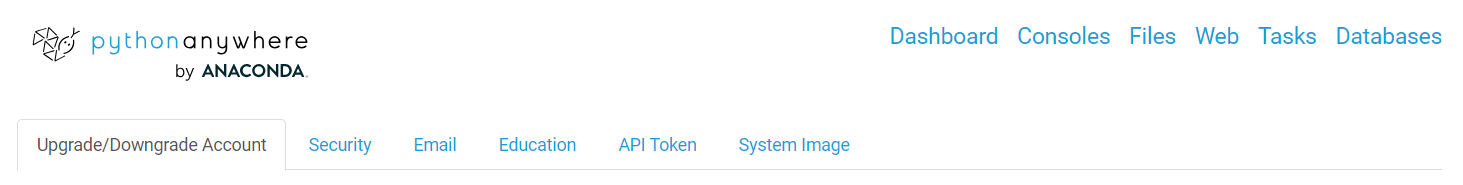
Если всё сделано правильно, то код из локального репозитория должен скопироваться на GitHub.

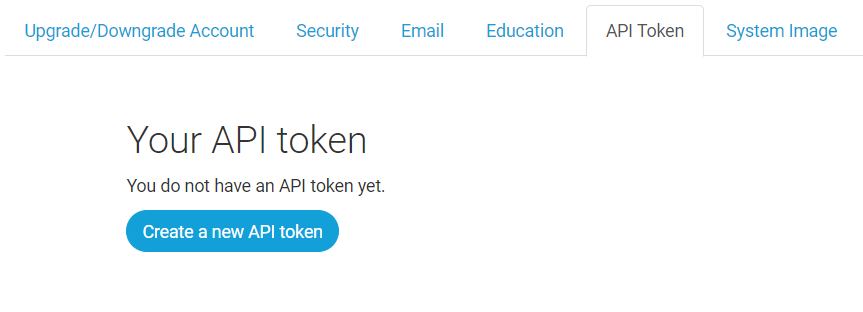
1. Настройка блога на PythonAnywhere

Создадим API токен для PythonAnywhere.

Это нужно будет сделать только один раз. Когда ты зарегистрируешься на PythonAnywhere, откроется панель управления (dashboard). На ней в правом верхнем углу будет ссылка на страницу «Account»:

Там выбери вкладку «API token» и нажми кнопку, на которой написано «Create new API token» (создать новый API token).





Вернись на главную страницу PythonAnywhere, кликнув логотип. Затем запусти Bash-консоль. Нажав на кнопку bash ты запускаешь командную строку, которая находится на серверах PythonAnywhere.

Чтобы опубликовать сайт на PythonAnywhere, нужно загрузить на PythonAnywhere код с Github и затем настроить PythonAnywhere так, чтобы он распознал код и запустил веб-приложение. Существуют способы сделать это «вручную», но для PythonAnywhere есть программа-помощник, которая сделает это сама. Установим её.

*pip3.6 install --user pythonanywhere*

Увидим такие последние строчки:

Successfully installed certifi-2024.8.30 charset-normalizer-2.0.12 click-8.0.4 docopt-0.6.2 idna-3.10 packaging-21.3 pyparsing-3.1.4 python-dateutil-2.9.0.post0 pythonanywhere-0.10.3 requests-2.27.1 schema-0.7.7 tabulate-0.8.10 typer-0.10.0 urllib3-1.26.2010:56 ~ **$**

Теперь запустим эту вспомогательную утилиту, которую мы установили. Она настроит приложение, скачав его код с GitHub. Напечатаем следующее в консоли PythonAnywhere (нельзя забывать использовать свой ник на GitHub , URL в консольной команде должен совпадать с URL, используемым в команде clone):

*pa\_autoconfigure\_django.py --nuke --branch=master --python=3.10* [*https://github.com/vanguar/freetour.git*](https://github.com/vanguar/freetour.git)

Утилита будет печатать в консоль, что она делает:

Скачивает твой код с GitHub

Создаёт виртуальное окружение на PythonAnywhere, такое же, как на твоём компьютере

Обновляет твой файл настроек с настройками деплоя

Создаёт базу данных на PythonAnywhere, используя команду manage.py migrate

Разбирается с твоими статическими файлами (о них будет дальше)

Настраивает PythonAnywhere так, чтобы твоё приложение было доступно в интернете

Все эти шаги автоматизированы на PythonAnywhere, но они совершенно такие же, какие надо было бы совершить с любым другим хостинговым сервисом.

Главное, на что нужно обратить внимание сейчас, — это то, что наша база данных на PythonAnywhere никак не связана с базой данных на компьютере. Поэтому там будут разные посты и разные аккаунты администраторов. Как следствие, для базы на PythonAnywhere необходимо создать аккаунт администратора так же, как это делали локально с помощью команды createsuperuser. На PythonAnywhere заранее активировано виртуальное окружение, так что всё, что нужно сделать — это запустить в консоли PythonAnywhere команду:

(freetourua.pythonanywhere.com) $ python manage.py createsuperuser

Введи параметры для своего пользователя-админа. Лучше всего использовать те же самые данные, что и на локальном компьютере, чтобы избежать путаницы.

Сейчас, если хочешь, посмотри на файлы на PythonAnywhere с помощью команды ls:

*(freetourua.pythonanywhere.com) 12:18 ~/freetourua.pythonanywhere.com (master)$ ls*

*blog blogtour db.sqlite3 manage.py python readmy.docx requirements.txt runtime.txt static*

Ты также можешь заглянуть на страницу «Files» и посмотреть, что лежит на сервере, используя встроенный в PythonAnywhere файловый менеджер.

твой сайт теперь доступен всем в интернете! Ты можешь найти ссылку на него на странице «Web» на PythonAnywhere.

Что нужно сделать дальше?

Нужно перейти во вкладку Files, найти файл settings.py, в нём найти строку:

*# SECURITY WARNING: don't run with debug turned on in production!*

*DEBUG = True*

И поменять True на False

Должно выглядеть так

*DEBUG = False*

Если оставить DEBUG = True, то при ошибках пользователи смогут увидеть подробную информацию о твоём приложении, что небезопасно.

Далее нажмите на кнопку Save, нажмите на логотип, чтобы выйти на главное окно, перейдите на вкладку Web и нажмите Reload, чтобы изменения вступили в силу.

После того как ты загрузил изменения на GitHub, нужно обновить код на сервере и перезапустить приложение на PythonAnywhere. Вот что нужно сделать:

Войти в консоль на PythonAnywhere:

Открой Bash-консоль на PythonAnywhere.

Перейти в директорию твоего проекта:

Обычно это папка с названием твоего проекта. В твоём случае:

*cd ~/freetourua.pythonanywhere.com*

Обновить код из репозитория.

Выполни команду:

*git pull*

Это подтянет последние изменения из GitHub.

Активировать виртуальное окружение:

*workon freetourua.pythonanywhere.com*

Установить новые зависимости (если изменял requirements.txt):

*pip install -r requirements.txt*

Применить миграции (если есть новые миграции):

*python manage.py migrate*

Собрать статические файлы (если нужно)

*python manage.py collectstatic –noinput*

Перезагрузить веб-приложение:

Перейди в панель управления PythonAnywhere, выбери вкладку "Web" и нажми кнопку "Reload" для твоего приложения.

После этих шагов твои изменения должны отобразиться на сайте.

Также нужно учесть, что каждые три месяца нужно подтверждать актуальность размещения в сети Вашего сайта нажатием кнопки **Run until 3 months from today**

