



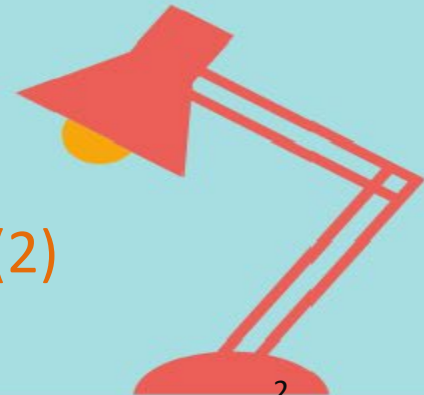
111-2進階程式設計課程(17)

Advanced Computer Programming

亞大資工系

課程大綱

- W1-課程介紹/Introduction
- W2-Python libraries
- W3-BeautifulSoup(1)
- W4-BeautifulSoup(2)
- W5-
- W6-Scrapy(1)
- W7-Scrapy(2)
- W8-Storing Data
- W9-Midterm project
- W10-Web & HTTP
- W11-Flask
- W12-Flask Routes
- W13-Jinja template
- W14-Flask-form
- W15-Flask-mail
- W16-REST API
- W17-Project development(2)
- W18-Final presentation



Front-End 前端設計

- 前端涵蓋了設計和實現使用者介面、互動、動畫效果等方面，包括：
 1. 網頁前端開發（ Front-end Development ）：使用 HTML、CSS、JavaScript 等技術實現前端設計，包括將設計轉化為可瀏覽的網頁、網頁動畫、互動效果等。
 2. 跨平台設計（ Cross-platform Design ）：設計師需要考慮不同的瀏覽器、操作系統、設備，使網頁能夠在各種平台上正常運行。
 3. 使用者體驗設計（ User Experience Design, UX Design ）：設計師透過使用者研究和測試，設計使用者界面的結構、流程、內容，以提升使用者體驗。
 4. 網頁設計（ Web Design ）：將設計師提供的視覺設計稿轉換成實際的網頁，設計網頁佈局、配色、字體等，並考慮不同設備（電腦、手機、平板）的適應性。
 5. 樣式設計（ Styling ）：設計網頁元素的樣式，例如色彩、字體、大小、邊框、背景等，以及使用 CSS 技術實現這些樣式。
 6. 互動設計（ Interactive Design ）：設計師設計使用者與網頁互動的方式，例如按鈕、表單、滑動、動畫效果等。



Back-end 後端設計

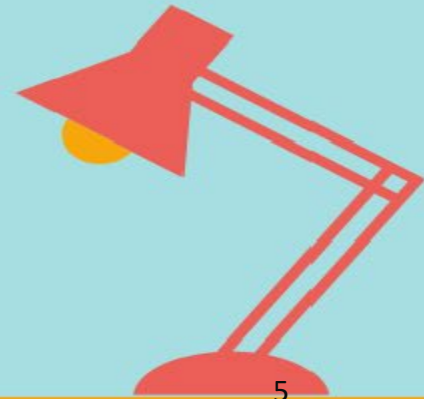
- 後端設計是指設計和開發網絡應用程序或軟件的服務器端的過程：
 - 服務器端編程語言：用於開發後端邏輯的編程語言的選擇，例如 Java、Python、PHP、Ruby 或 Node.js。
 - 資料庫：資料庫技術的選擇和資料庫模式的設計來管理應用程序的資料。
 - API：允許應用程序前端與後端通信的 API 的設計和實現。
 - 身份驗證和授權：設計後端來處理用戶身份驗證和授權，例如密碼管理、權限和訪問控制。
 - 服務器架構：服務器架構的設計，包括負載均衡、緩存和其他確保可擴展性和高性能的技術。



免費的線上 Python 執行環境

PythonAnywhere

- 免費帳號的功能受到如下限制：
 - 只能建立一個 App (應用程式)
 - 網外存取 Internet 有限制
 - CPU 與儲存有限制 (一天 100 秒 CPU 時間, 512MB 儲存)
 - 不提供 Jupyter (但有 IPython)
 - 只能有兩個 Console (Bash 與 Python)




System Image

System Image

The system image for your account determines the versions of Python that you can use and the packages that are pre-installed. [This page](#) shows the packages that are installed in each system image and in each version of Python.


Changing your system image may mean that you need to change your code. It may result in changing of your default python3 and/or editor Run button python versions. Please do not change it without reading [this help page](#) first.


Current system image: fishnchips 

Default Python Versions

You need to upgrade your system image to be able to change your default python command.


python should run: 2.7

python3 should run: 3.8 

The editor Run button should use: 3.8 



Dashboard儀表板

 pythonanywhere

Dashboard Consoles Files Web Tasks Databases

Up here you will see instructions walking you through the key features of our Education beta.

(If after closing this helper, you want to go through it again – or try another one – go to the [Help page](#))

→

Close this tutorial

Dashboard

Welcome, [htchu](#)

CPU Usage: 0% used – 0.00s of 100s. Resets in 10 hours, 4 minutes [More Info](#)

File storage: 16% full – 80.1 MB of your 512.0 MB quota [More Info](#)

[Upgrade Account](#)

Recent Consoles

[+ 5 -](#)

You have no recent consoles.

[View all](#)

New console:

[\\$ Bash](#) [>>> Python](#) [More...](#)

Recent Files

[+ 5 -](#)

[/home/htchu/mysite/flask_app.py](#)

[/home/htchu/mysite3/urls.py](#)

[/home/htchu/mysite3/polls/urls.py](#)

[/home/htchu/mysite3/polls/views.py](#)

[/home/htchu/mysite3/manage.py](#)

[+ Open another file](#) [Browse files](#)

Recent Notebooks

[+ 5 -](#)

Your account does not support Jupyter Notebooks. [Upgrade your account](#) to get access!


All Web apps

[htchu.pythonanywhere.com](#)

[Open Web tab](#)




Command Console 命令控制台

 pythonanywhere

Dashboard **Consoles** Files Web Tasks Databases

Up here you will see instructions walking you through the key features of our Education beta.


(If after closing this helper, you want to go through it again – or try another one – go to the [Help page](#))




Close this tutorial

CPU Usage: 0% used – 0.00s of 100s. Resets in 16 hours, 2 minutes [More Info](#)

Start a new console:

Python: [3.8](#) / [3.7](#) / [3.6](#) / [3.5](#) / [2.7](#) IPython: [3.8](#) / [3.7](#) / [3.6](#) / [3.5](#) / [2.7](#) PyPy: [2](#) / [3](#)
Other: [Bash](#) | [MySQL](#)
Custom: 


Your consoles:

[flask_app.py \(Python3.8\)](#) 

Consoles shared with you

No-one has shared any consoles with you :(

Running processes

 [Fetch process list](#)



Files

Up here you will see instructions walking you through the key features of our Education beta.
(If after closing this helper, you want to go through it again – or try another one – go to the [Help page](#))



Close this tutorial

/home/  htchu

 [Open Bash console here](#)

16% full – 80.1 MB of your 512.0 MB quota [More Info](#)

Directories

Enter new directory name

New directory





























[.cache/](#)
[.config/](#)
[.local/](#)
[.virtualenvs/](#)
[mysite/](#)
[mysite2/](#)
[mysite3/](#)



Files

Enter new file name, eg hello.py

New file


 .bashrc	  	2021-05-04 07:38	559 bytes
 .gitconfig	  	2021-05-04 07:38	266 bytes
 .profile	  	2021-05-04 07:38	79 bytes
 .python_history	  	2021-05-04 08:18	7 bytes
 .pythonstartup.py	  	2021-05-04 07:38	77 bytes
 .vimrc	  	2021-05-04 07:38	4.6 KB
 README.txt	  	2021-05-04 07:38	232 bytes

 Upload a file

100MiB maximum size




Web

 pythonanywhere

Dashboard Consoles Files **Web** Tasks Databases


Up here you will see instructions walking you through the key features of our Education beta.

(If after closing this helper, you want to go through it again – or try another one – go to the [Help page](#))




Close this tutorial

htchu.pythonanywhere.com



Configuration for htchu.pythonanywhere.com

Reload:



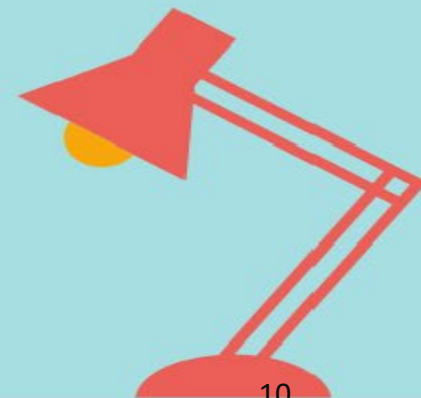
Best before date:

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 has expired, click the button below to reactivate it

Run until 3 months from today

Paying users' sites stay up forever without any need to log in to keep them running.



Tasks任務

 pythonanywhere

Dashboard Consoles Files Web **Tasks** Databases

Up here you will see instructions walking you through the key features of our Education beta.

(If after closing this helper, you want to go through it again – or try another one – go to the [Help page](#))

→

Close this tutorial

CPU Usage: 0% used – 0.00s of 100s. Resets in 15 hours, 58 minutes [More Info](#)

Scheduled tasks

Server time: 02:15 UTC

Daily, at : UTC [Create](#)

Frequency	Time	Command	Description	Expiry	Actions
You have no tasks yet.					

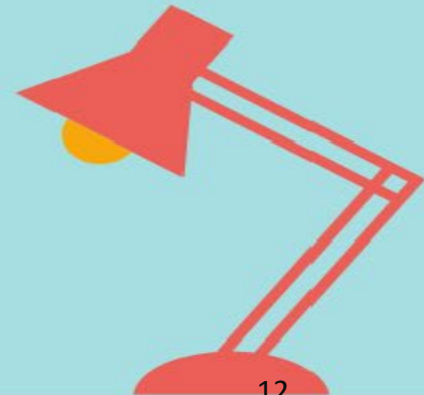
Your **Scheduled task** is a script that will run every day at a time of your choosing – you can use it to do stuff like scraping websites, or checking that your server is running.

[Paying users](#) can schedule several tasks, can run them both hourly and daily, and they never expire. Just sayin'...



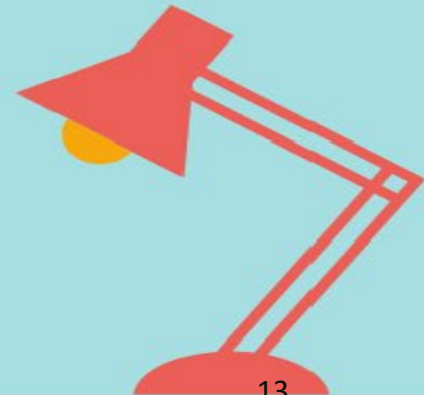
Web application frameworks

- JavaScript
 - Express.js, React.js, Angular.js
- PHP
 - Laravel, CodeIgniter
- Ruby
 - Rails
- Python
 - Django, Flask , FastAPI
- Java
 - Spring Boot



Flask framework

- Required
 - **Jinja**-template engine
 - **Werkzeug**-WSGI toolkit
- Optional
 - **sqlalchemy**-SQL toolkit
 - marshmallow: simplified object serialization
 - Celery-task queue



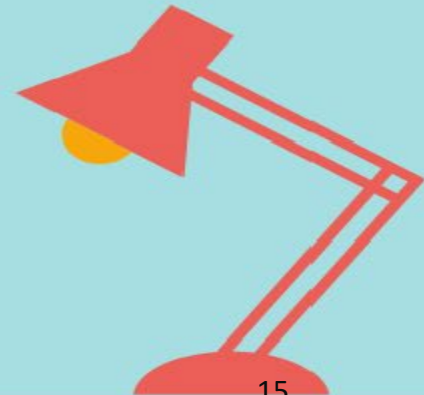
Flask extensions

- **Flask-Bootstrap:** Bootstrap
- **Flask-WTF:** WTForms including CSRF, file upload, and reCAPTCHA
- **Flask-Moment:** Localization of Dates and Times
- **Flask-Babel:** Internationalization and localization support
- **Flask-DebugToolbar:** In-browser debugging tools
- **Flask-Assets:** Integration of CSS and JavaScript assets
- **Flask-Session:** implementation of user sessions with server-side storage
- **Flask-SocketIO:** Socket.IO server implementation with support for WebSocket and long-polling



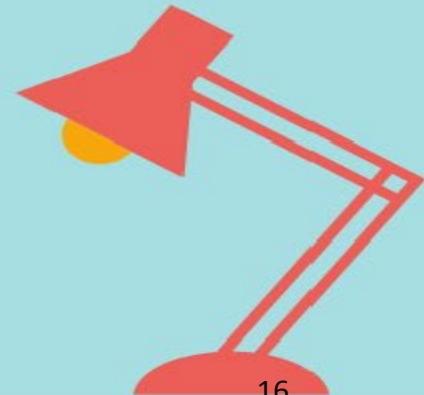
Flask

- Flask is a class with
 - run() function
 - route() functions
- Flask is a command
 - flask run
 - flask routes
 - flask shell



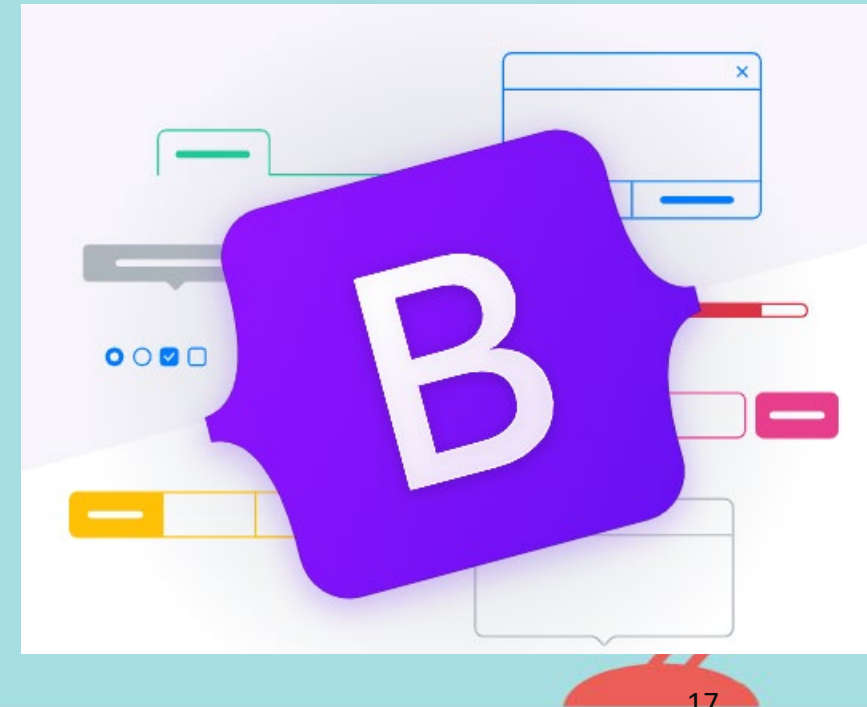
Flask environment

- Bash (Linux/Mac)
 - export FLASK_APP= appname
 - export FLASK_ENV=development
 - flask run
- Windows command
 - set FLASK_APP=appname
 - set FLASK_ENV=development
 - flask run



Bootstrap

- The most popular front-end toolkit in the world.
- Quickly design and customize responsive mobile-first sites
- Currently v5.1.3



Bootstrap-Flask

Bootstrap-Flask

license MIT pypi v2.0.2 build passing coverage 92%

Bootstrap-Flask is a collection of Jinja macros for Bootstrap 4 & 5 and Flask. It helps you to render Flask-related data and objects to Bootstrap markup HTML more easily:

- Render Flask-WTF/WTForms form object to Bootstrap Form.
- Render data objects (dict or class objects) to Bootstrap Table.
- Render Flask-SQLAlchemy `Pagination` object to Bootstrap Pagination.
- etc.

Installation

```
$ pip install -U bootstrap-flask
```

Example

Register the extension:

```
from flask import Flask
# To follow the naming rule of Flask extension, although
# this project's name is Bootstrap-Flask, the actual package
# installed is named `flask_bootstrap`.
from flask_bootstrap import Bootstrap5

app = Flask(__name__)
bootstrap = Bootstrap5(app)
```

Installation

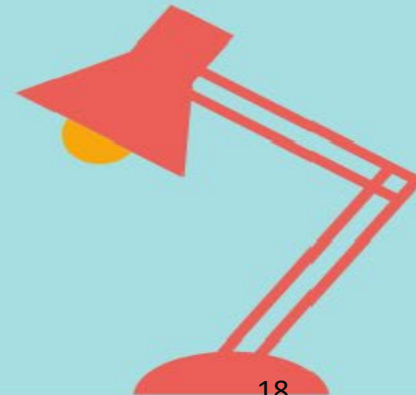
```
$ pip install -U bootstrap-flask
```

Example

Register the extension:

```
from flask import Flask
from flask_bootstrap import Bootstrap5
```

```
app = Flask(__name__)
bootstrap = Bootstrap5(app)
```



Jinja-the template language

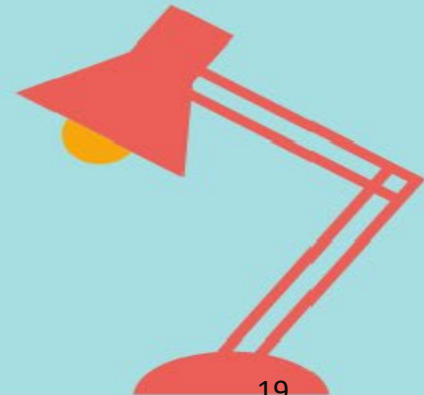
```
from jinja2 import Environment, PackageLoader, select_autoescape
env = Environment(
    loader=PackageLoader("yourapp"),
    autoescape=select_autoescape()
)
```

To load a template from this environment, call the `get_template()` method, which returns the loaded Template.

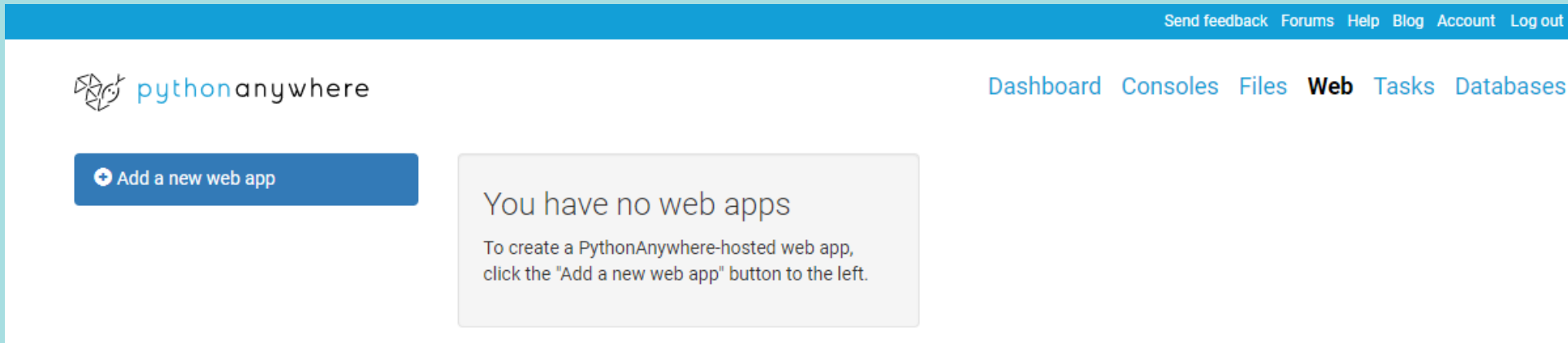
```
template = env.get_template("mytemplate.html")
```

To render it with some variables, call the `render()` method.

```
print(template.render(the="variables", go="here"))
```



Step 1: Go to Web tab



The screenshot shows the PythonAnywhere dashboard interface. At the top, a blue navigation bar contains links for "Send feedback", "Forums", "Help", "Blog", "Account", and "Log out". Below this, the "pythonanywhere" logo is on the left, and a horizontal menu on the right includes "Dashboard", "Consoles", "Files", "Web" (which is bolded and underlined), "Tasks", and "Databases". On the left side of the main content area, there is a blue button with a plus icon and the text "Add a new web app". To the right of this button is a light gray box with the heading "You have no web apps" and the text "To create a PythonAnywhere-hosted web app, click the 'Add a new web app' button to the left."

Send feedback Forums Help Blog Account Log out

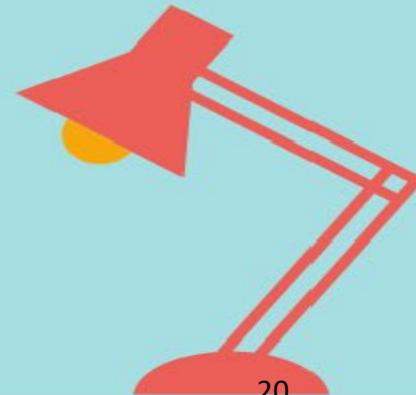
pythonanywhere

Dashboard Consoles Files **Web** Tasks Databases

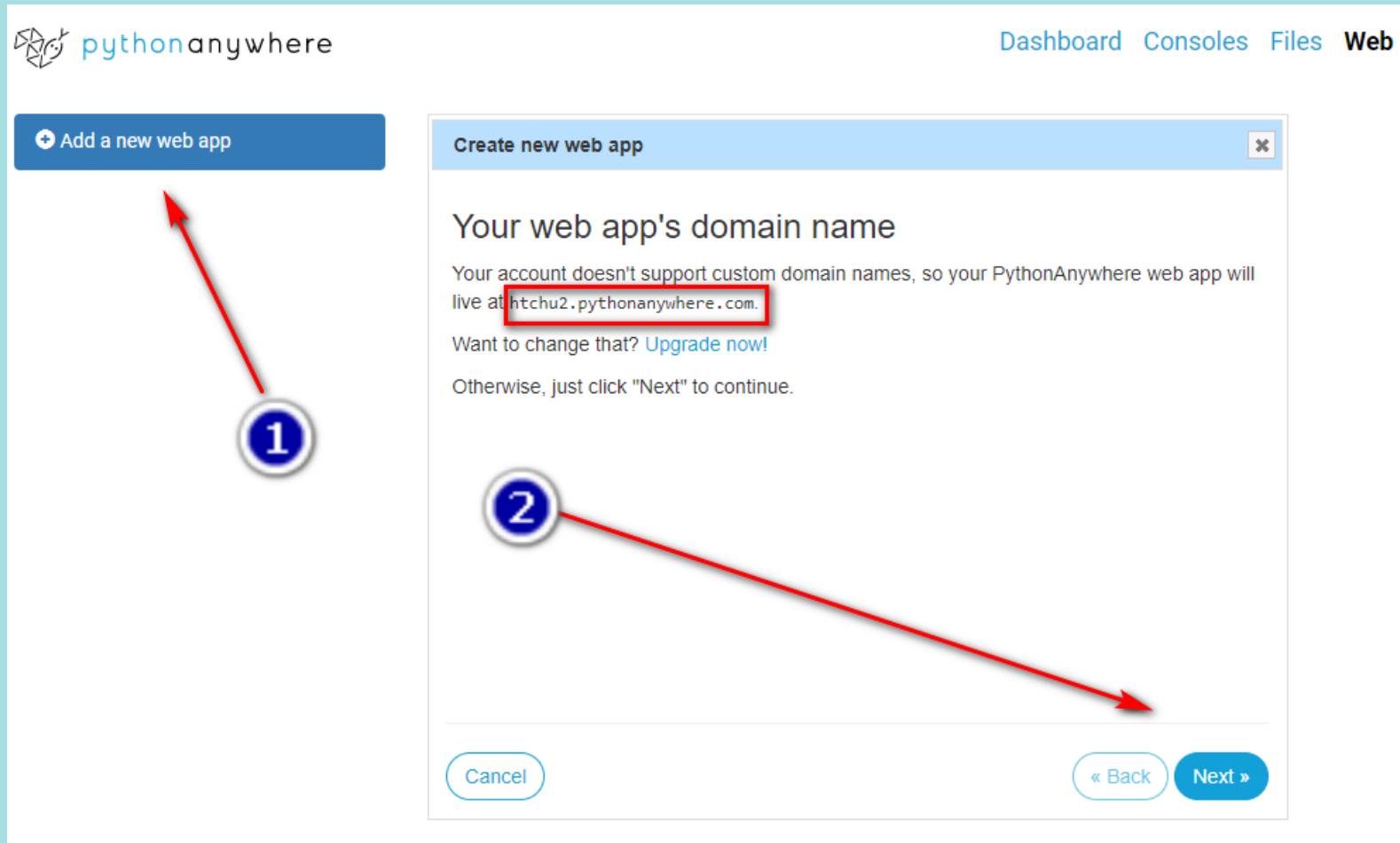
+ Add a new web app

You have no web apps

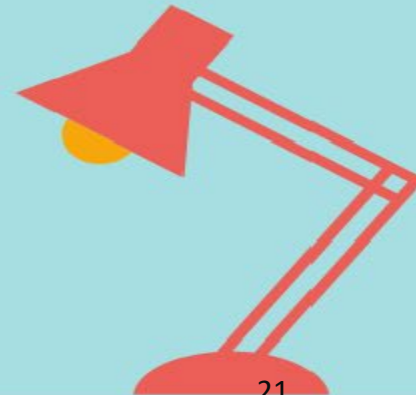
To create a PythonAnywhere-hosted web app, click the "Add a new web app" button to the left.



Step 2: Add a new web app



The screenshot shows the PythonAnywhere dashboard. At the top left is the PythonAnywhere logo. At the top right are navigation links: Dashboard, Consoles, Files, and Web. On the left side, there is a blue button labeled '+ Add a new web app'. A red arrow points from a blue circle with the number '1' to this button. On the right side, a modal window titled 'Create new web app' is open. Inside the modal, the text reads: 'Your web app's domain name', 'Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `htchu2.pythonanywhere.com`', 'Want to change that? [Upgrade now!](#)', and 'Otherwise, just click "Next" to continue.' A red arrow points from a blue circle with the number '2' to the 'Next »' button at the bottom right of the modal. The modal also has a 'Cancel' button on the bottom left and a close button (X) in the top right corner.



Step 3: Select a Python Web framework and a Python version

Select a Python Web framework

...or select "Manual configuration" if you want detailed control.

- » Django
- » web2py
- » Flask
- » Bottle
- » **Manual configuration** (including virtualenvs)

What other frameworks should we have here? Send us some feedback using the link at the top of the page!

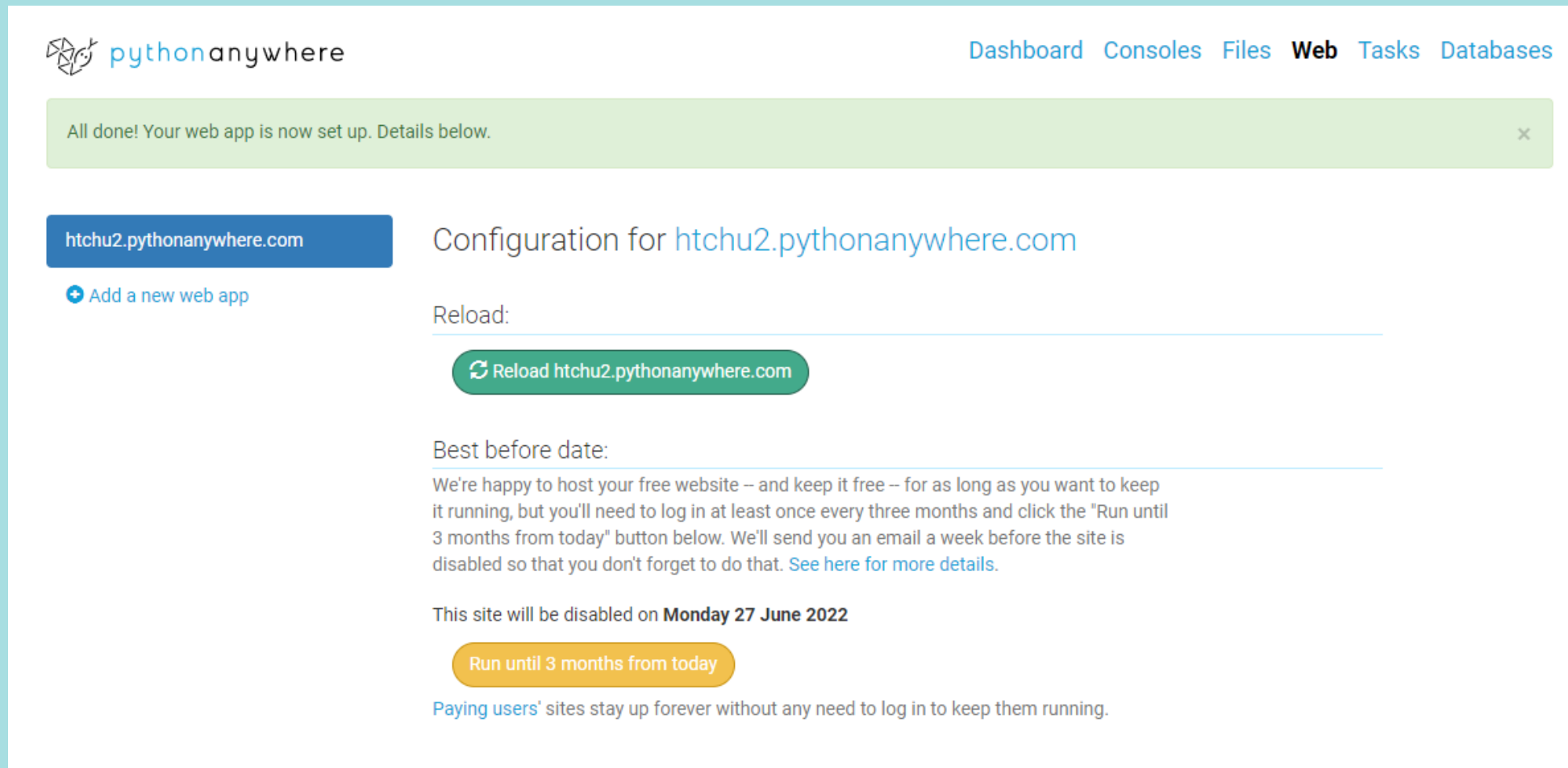
Select a Python version

- » Python 3.6 (Flask 2.0.0)
- » Python 3.7 (Flask 2.0.0)
- » Python 3.8 (Flask 2.0.0)
- » Python 3.9 (Flask 2.0.0)

Note: If you'd like to use a different version of Flask to the default version, you can use a virtualenv for your web app. There are [instructions here](#).



Step 4: Quick start new Flask project



The screenshot shows the PythonAnywhere web interface. At the top left is the PythonAnywhere logo. To the right are navigation links: Dashboard, Consoles, Files, Web (highlighted), Tasks, and Databases. A green notification bar at the top states: "All done! Your web app is now set up. Details below." Below this, on the left, is a blue button labeled "htchu2.pythonanywhere.com" and a link "+ Add a new web app". The main content area is titled "Configuration for htchu2.pythonanywhere.com". It contains a "Reload:" section with a green button "Reload htchu2.pythonanywhere.com". Below that is a "Best before date:" section with a paragraph explaining the free hosting policy and a yellow button "Run until 3 months from today". At the bottom, it states "This site will be disabled on Monday 27 June 2022" and includes a note for paying users.

pythonanywhere

Dashboard Consoles Files **Web** Tasks Databases

All done! Your web app is now set up. Details below. ×

htchu2.pythonanywhere.com

+ Add a new web app

Configuration for htchu2.pythonanywhere.com

Reload:

Reload htchu2.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.](#)

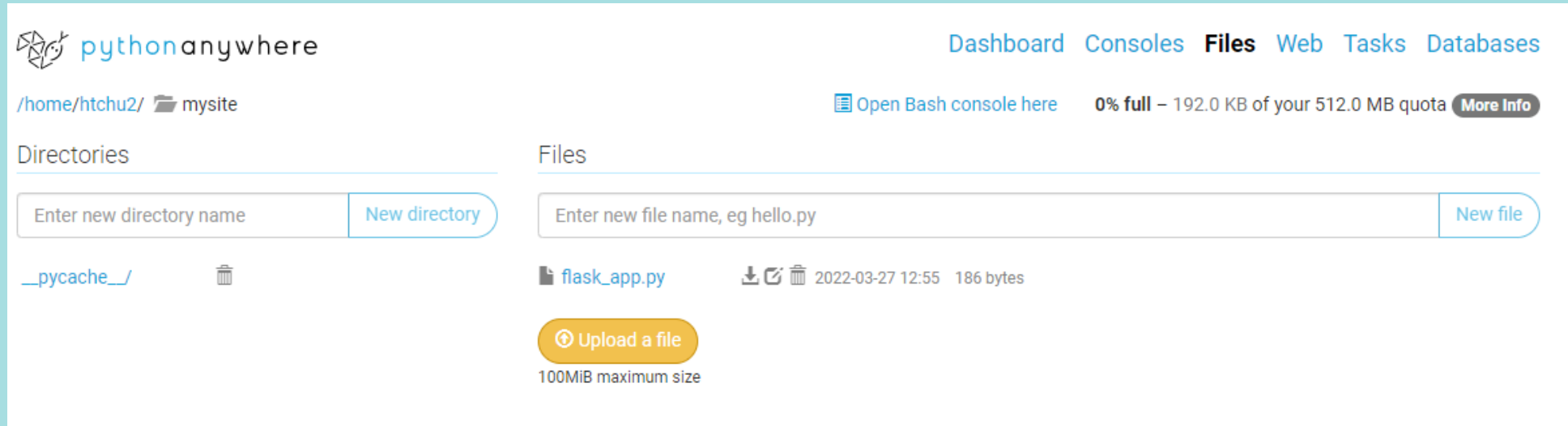
This site will be disabled on **Monday 27 June 2022**

Run until 3 months from today

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

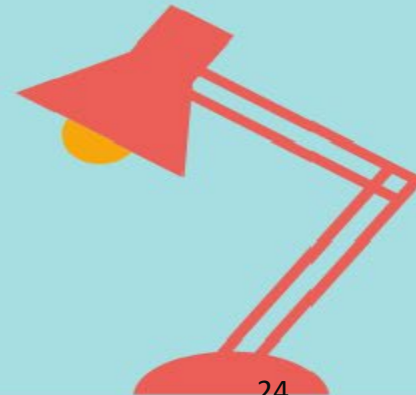


Step 5: check the files




The screenshot displays the PythonAnywhere web interface. At the top left is the PythonAnywhere logo. To its right are navigation links: Dashboard, Consoles, **Files**, Web, Tasks, and Databases. Below the logo, the current path is shown as `/home/htchu2/` with a folder icon and the name `mysite`. On the right side of the header, there is a link to "Open Bash console here" and a storage status indicator: "0% full – 192.0 KB of your 512.0 MB quota", followed by a "More Info" button.

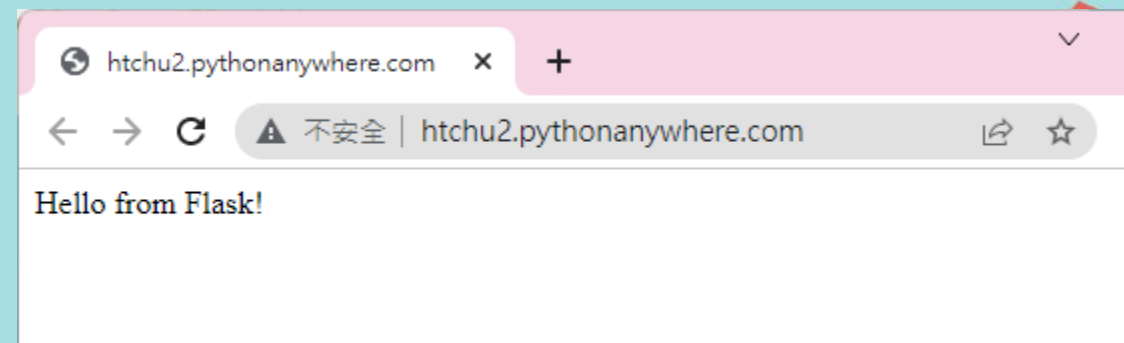
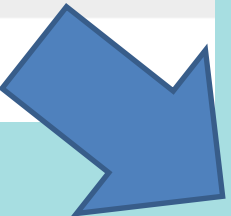
The main content area is divided into two panels. The left panel, titled "Directories", contains a text input field "Enter new directory name" and a "New directory" button. Below this, a directory named `__pycache__` is listed with a trash icon. The right panel, titled "Files", contains a text input field "Enter new file name, eg hello.py" and a "New file" button. Below this, a file named `flask_app.py` is listed with download, edit, and delete icons, along with its timestamp "2022-03-27 12:55" and size "186 bytes". At the bottom of the Files panel is an "Upload a file" button and the text "100MiB maximum size".



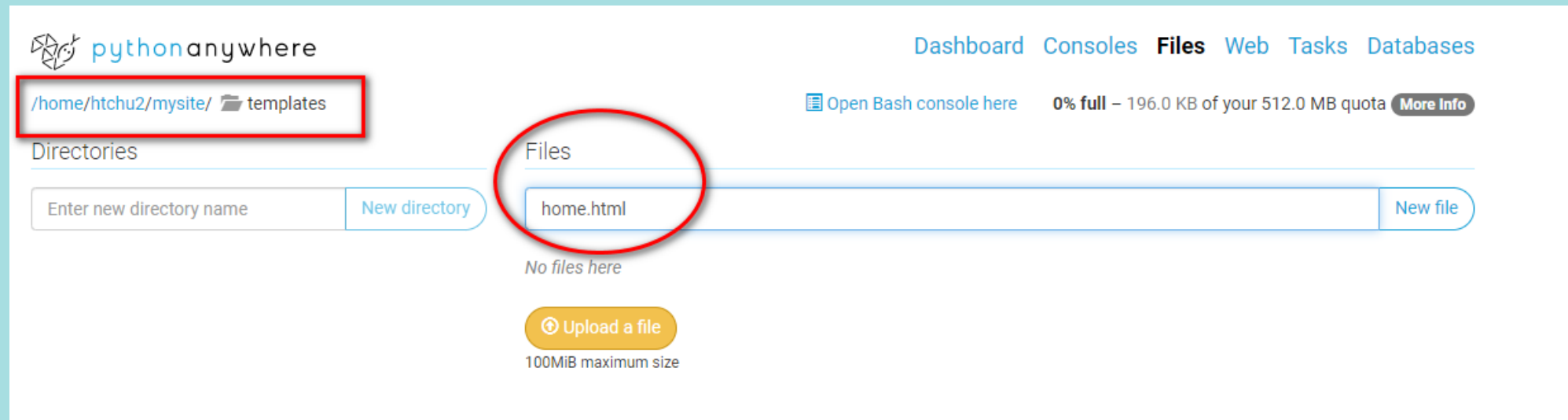
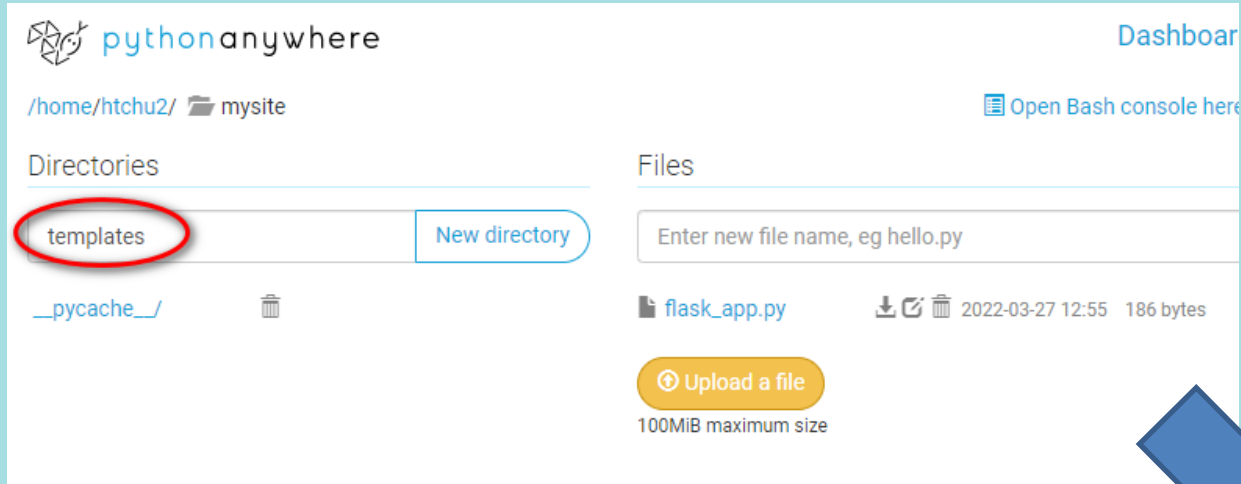
Step 6: check the program and the web app

 /home/htchu2/mysite/flask_app.py

```
1
2 # A very simple Flask Hello World app for you to get started with...
3
4 from flask import Flask
5
6 app = Flask(__name__)
7
8 @app.route('/')
9 def hello_world():
10     return 'Hello from Flask!'
11
12
```



Step 7: add html templates



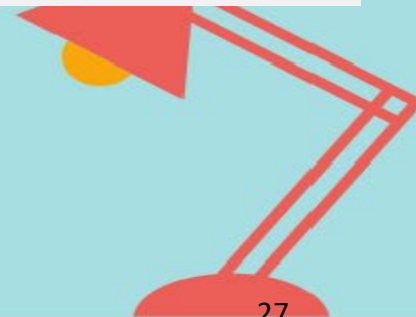
Step 8: edit home.html




- `<h1>Top 10 HTML tags</h1>`
`

` In this article we will explain our list of top 10 HTML tags.`

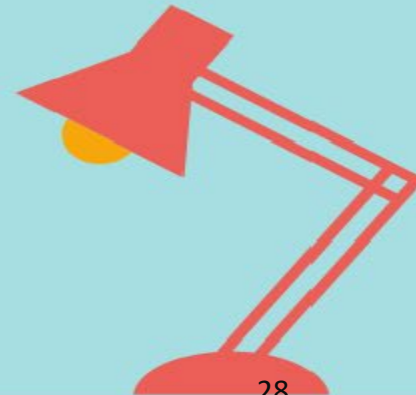
` `<h2>First tag is <A></h2>` Bold is all about making words more important.




Step 9: edit flask_app.py

 /home/htchu2/mysite/flask_app.py

```
1
2 # A very simple Flask Hello World app for you to get started with...
3
4 from flask import Flask, render_template
5
6 app = Flask(__name__)
7
8 @app.route('/')
9 def home():
10     return render_template("home.html")
11
```



Step 10: Reload web

 pythonanywhere

Dashboard Consoles Files Web

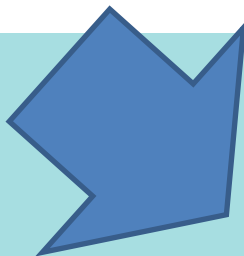
htchu2.pythonanywhere.com

+ Add a new web app

Configuration for htchu2.pythonanywhere.com

Reload:

Reload htchu2.pythonanywhere.com



htchu2.pythonanywhere.com x +

← → ↻ ⚠ 不安全 | htchu2.pythonanywhere.com

My First Web Page

Headings Are Great Fun

This is my first **paragraph** in my new *webpage*. This is going to be great. I am so excited I can hardly contain **myself**. Don't you just love paragraphs? I find them very useful.

Web Pages Are Exciting Too

Yes, that's right - web pages can be a lot of fun. Learning how to create web pages is easy and **entertaining**. This is my second *paragraph*. I hope you like it.

[Link to Google](#)

- Apples
- Bananas
- Pears
- Oranges
- Grapes

"A designer knows he has achieved perfection not when there is nothing left to add, but when there is nothing left to take away." - Saint Exupéry



Thanks!

Q&A

