111-2進階程式設計課程(16)
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-Databases
- W16-SQLAlchemy
- W17-Project development(2)
- W18-Final presentation

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)

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

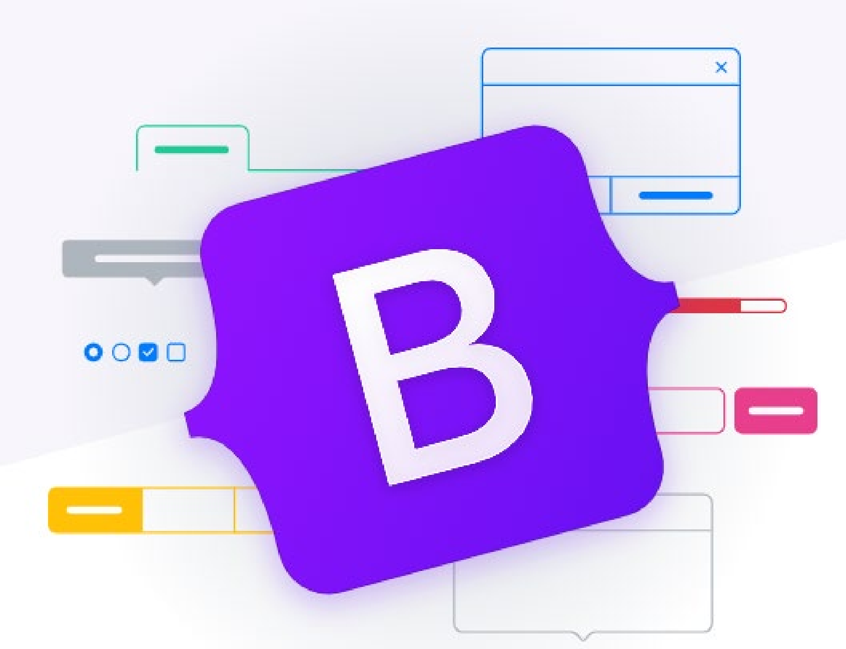
- Flask is a command
  - flask run
  - flask routes
  - flask shell

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

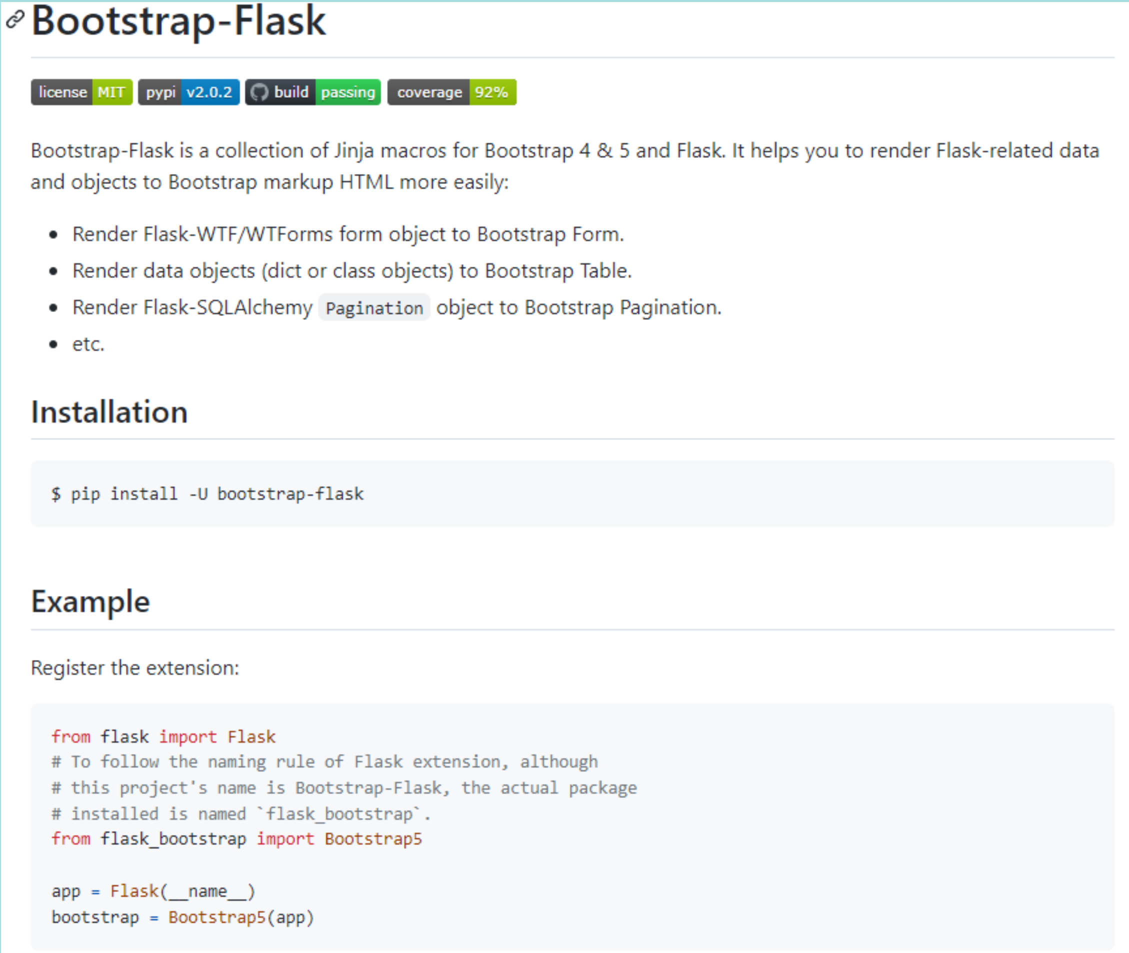
Bootstrap

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



https://getbootstrap.com/

Bootstrap-Flask



Bootstrap-Flask

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
```

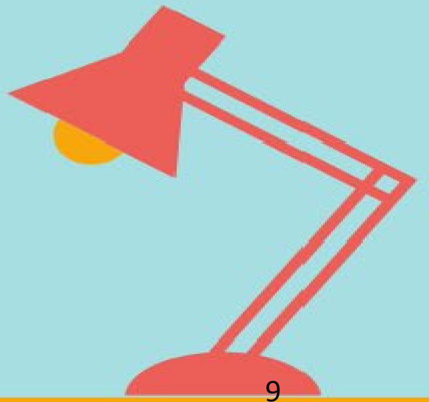
Example

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)
```

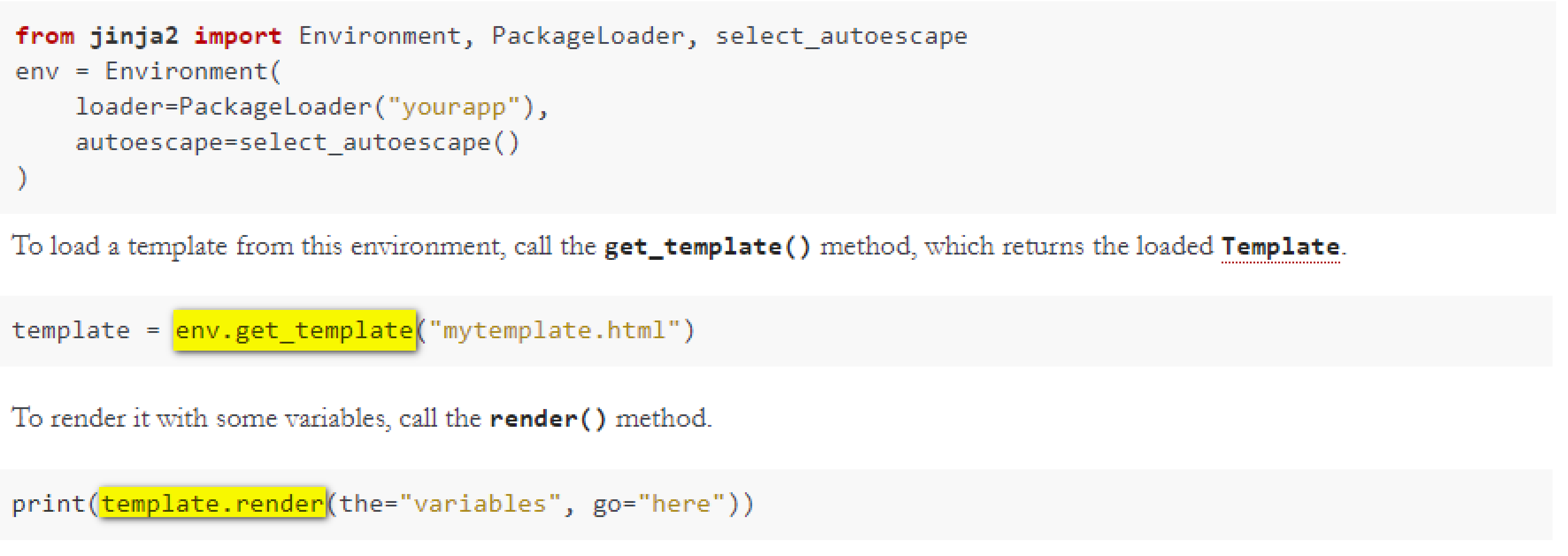
Installation
$ pip install -U bootstrap-flask
Example
Register the extension:

```
from flask import Flask
from flask_bootstrap import Bootstrap5

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



https://github.com/greyli/bootstrap-flask

Jinja-the template language



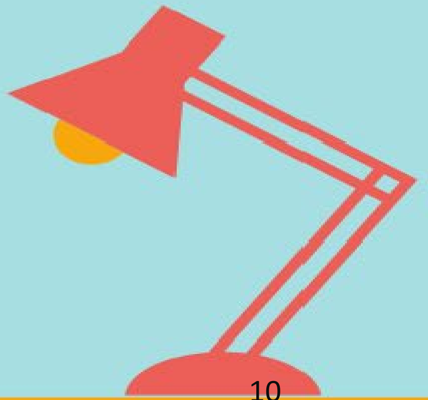
```python
from jinja2 import Environment, PackageLoader, select_autoescape
env = Environment(
    loader=PackageLoader("yourapp"),
    autoescape=select_autoescape()
)
```

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

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

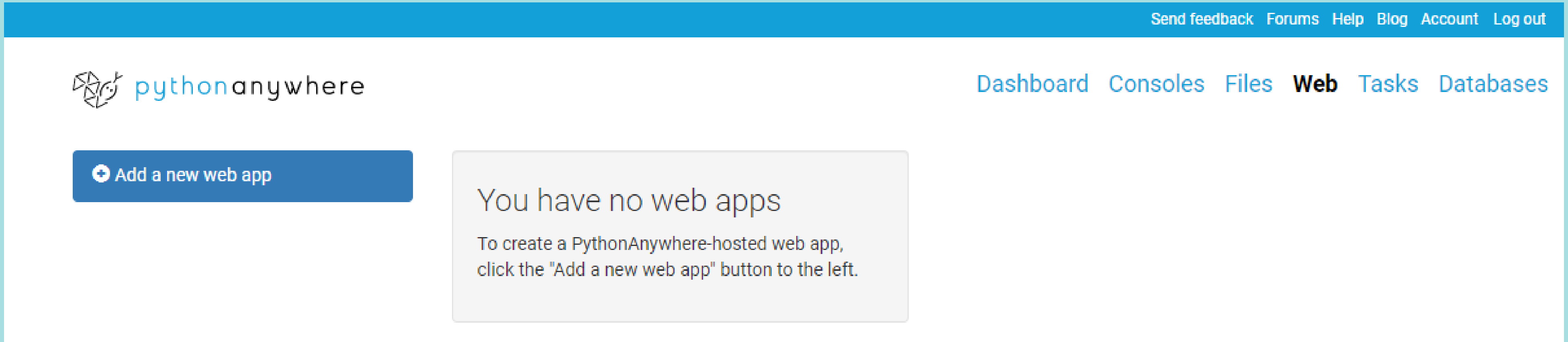
To render it with some variables, call the **`render()`** method.

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



https://jinja.palletsprojects.com/en/3.1.x/api/#basics

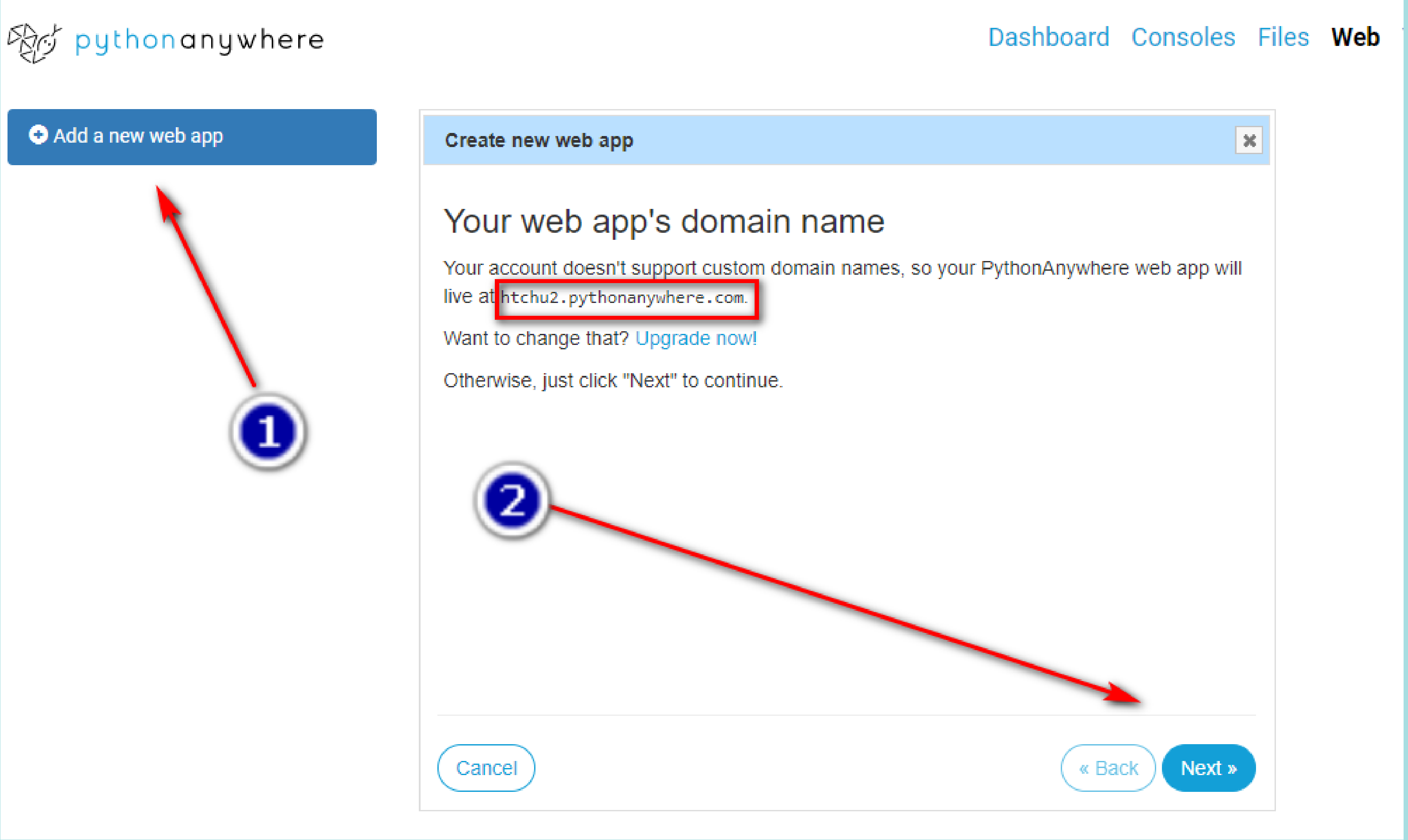
Step 1: Go to Web tab



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



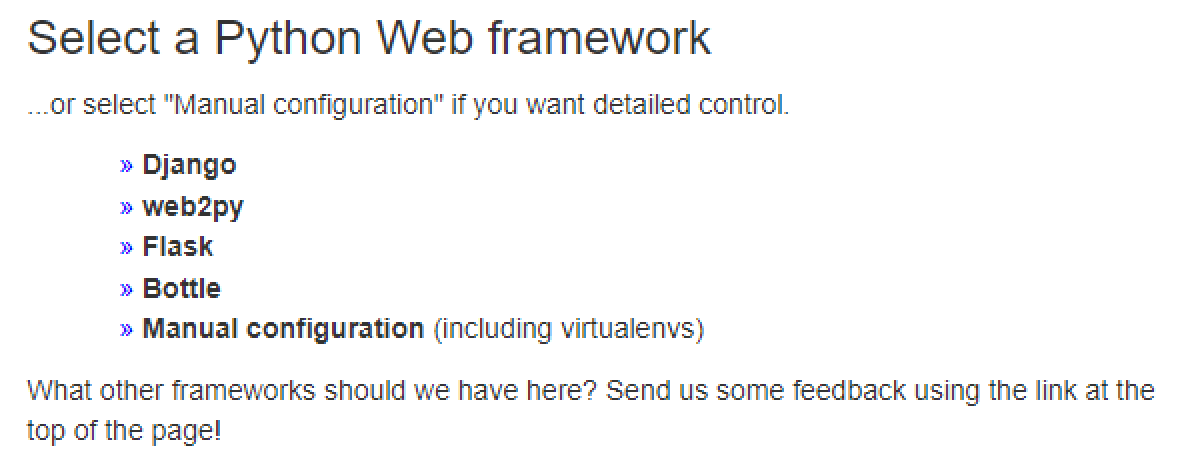
Your web app's domain name

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!

Otherwise, just click "Next" to continue.

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

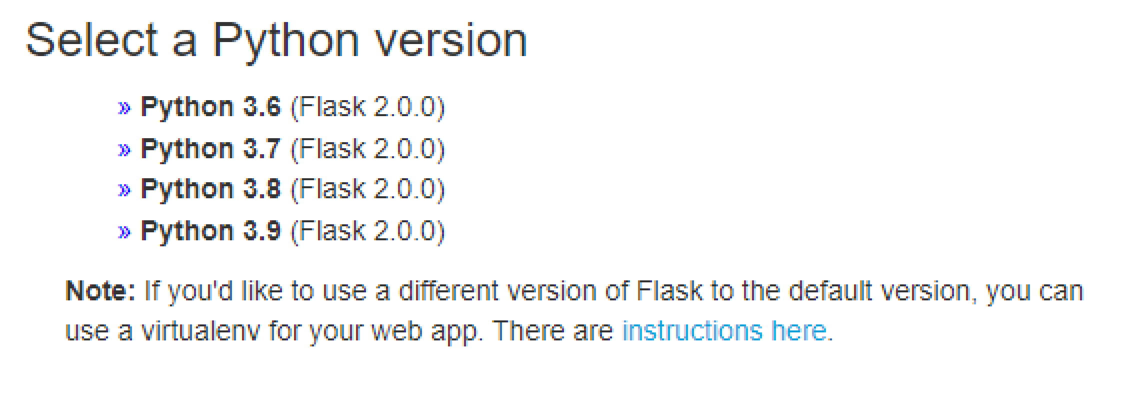


Select a Python Web framework

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

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

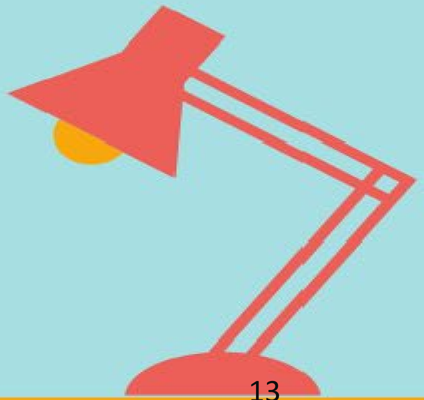
What other frameworks should we have here? Send us some feedback using the link at the top of the page!



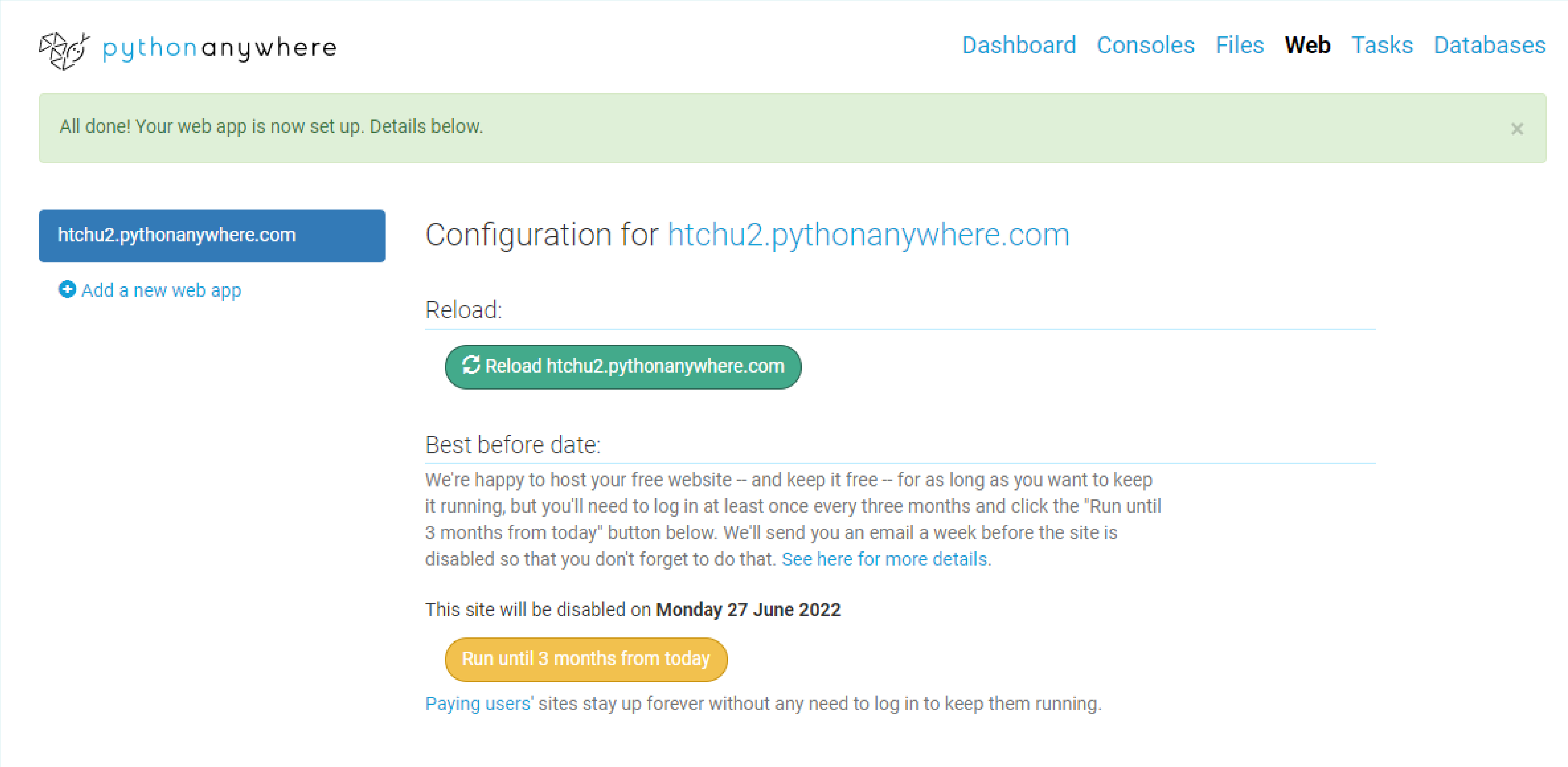
Select a Python version

- » **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



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

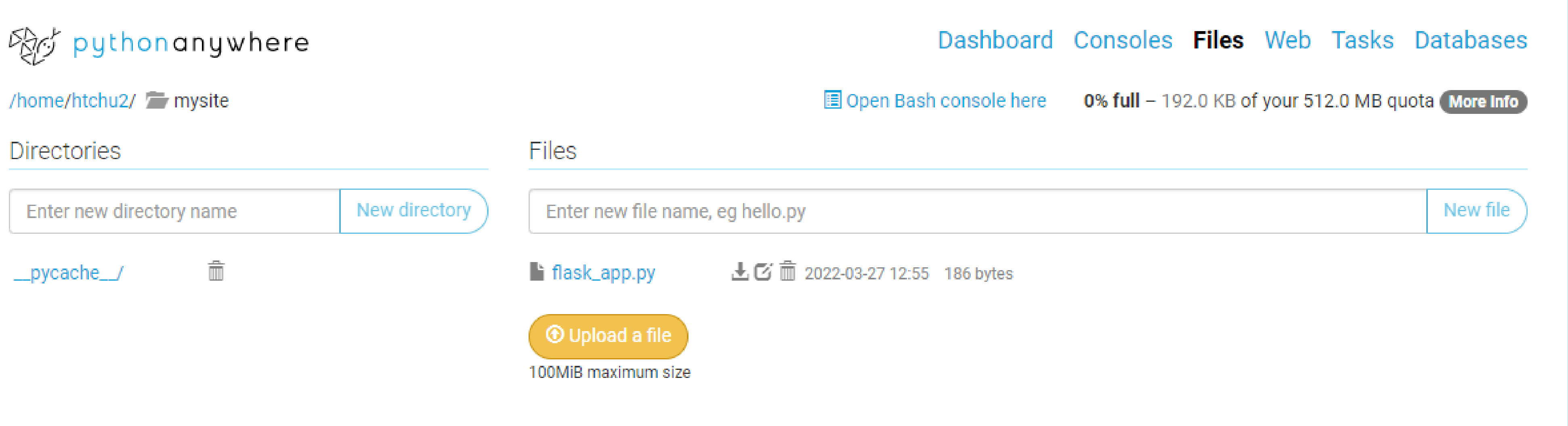
Best before date:

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**

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

Step 5: check the files



/home/htchu2/ mysite

0% full – 192.0 KB of your 512.0 MB quota

Directories

Enter new directory name

__pycache__/

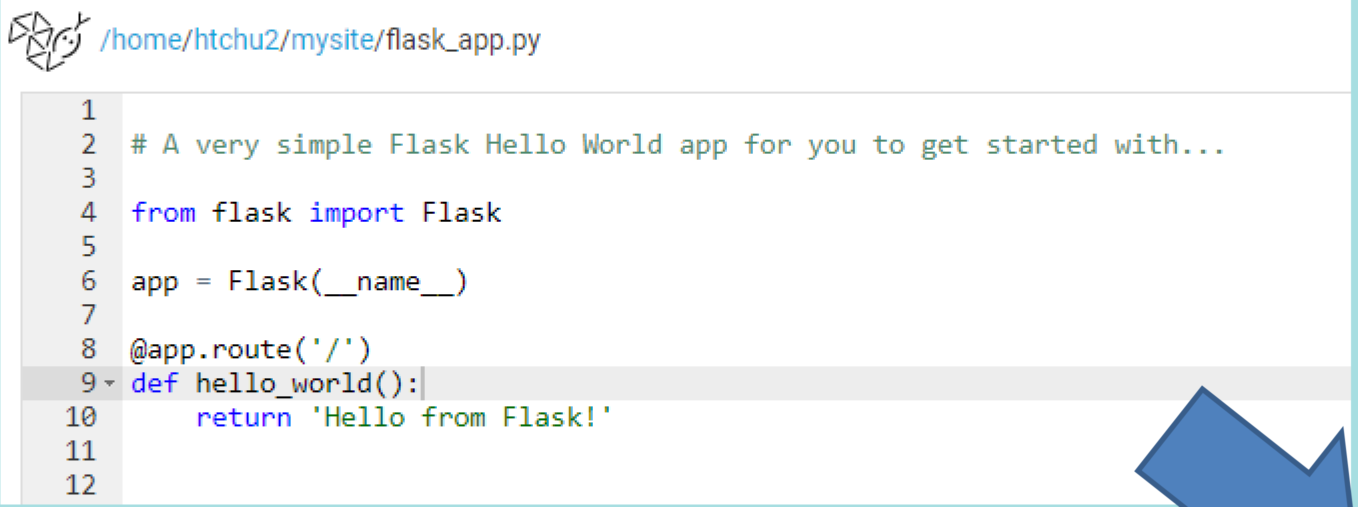
Files

Enter new file name, eg hello.py

flask_app.py 2022-03-27 12:55 186 bytes

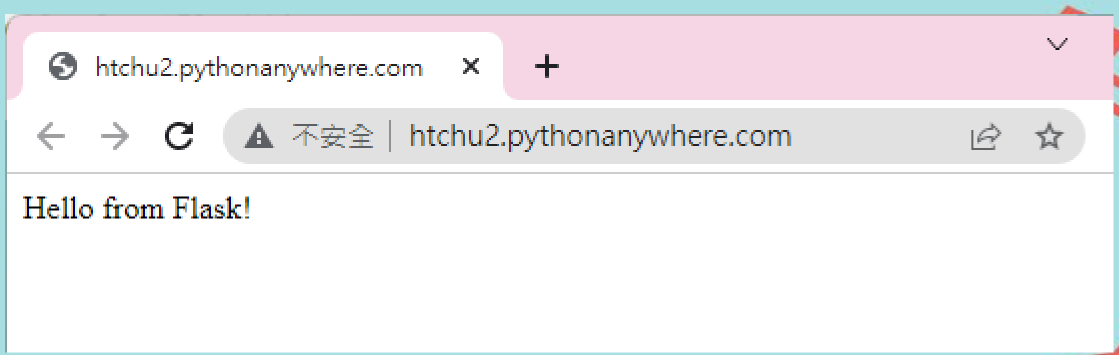
100MiB maximum size

Step 6: check the program and the web app



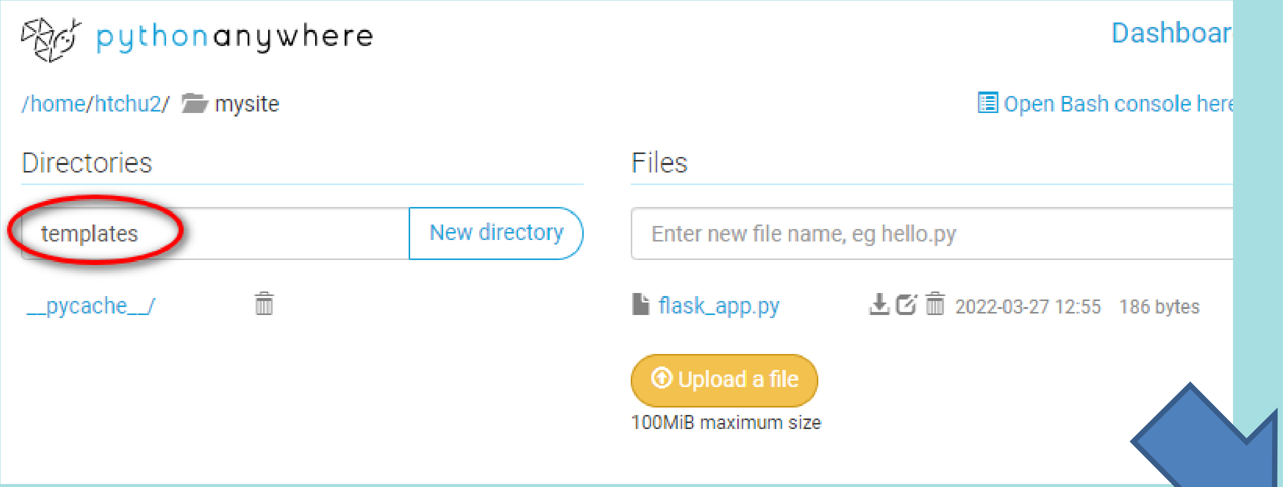
/home/htchu2/mysite/flask_app.py

```python

# A very simple Flask Hello World app for you to get started with...

from flask import Flask

app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello from Flask!'


```



```
Hello from Flask!
```

Step 7: add html templates



/home/htchu2/ mysite

Open Bash console here

Directories

templates

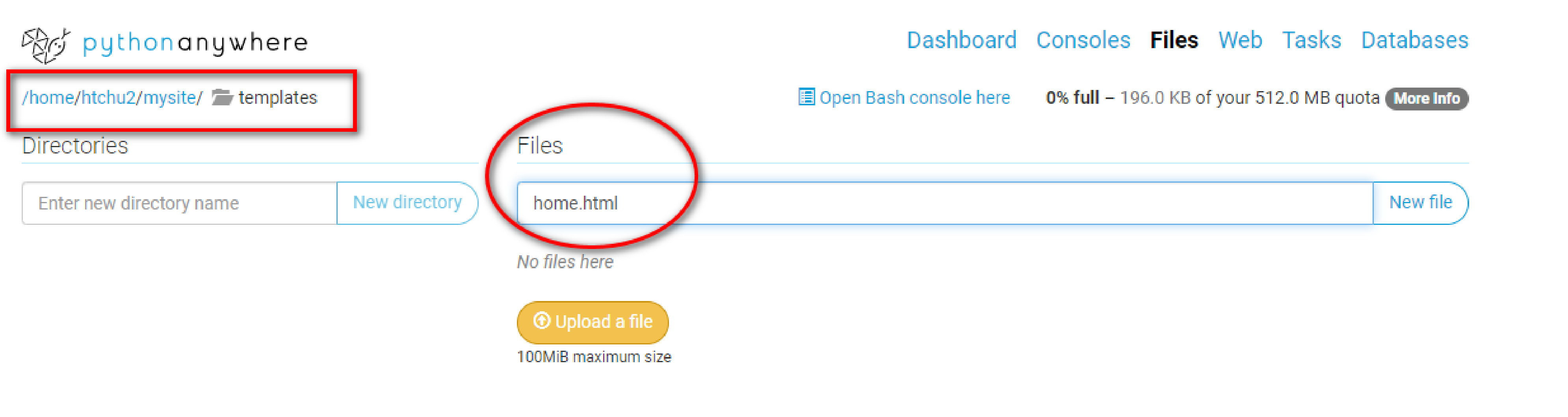
__pycache__/

Files

Enter new file name, eg hello.py

flask_app.py 2022-03-27 12:55 186 bytes

100MiB maximum size



/home/htchu2/mysite/ templates

Open Bash console here

0% full – 196.0 KB of your 512.0 MB quota

Directories

Enter new directory name

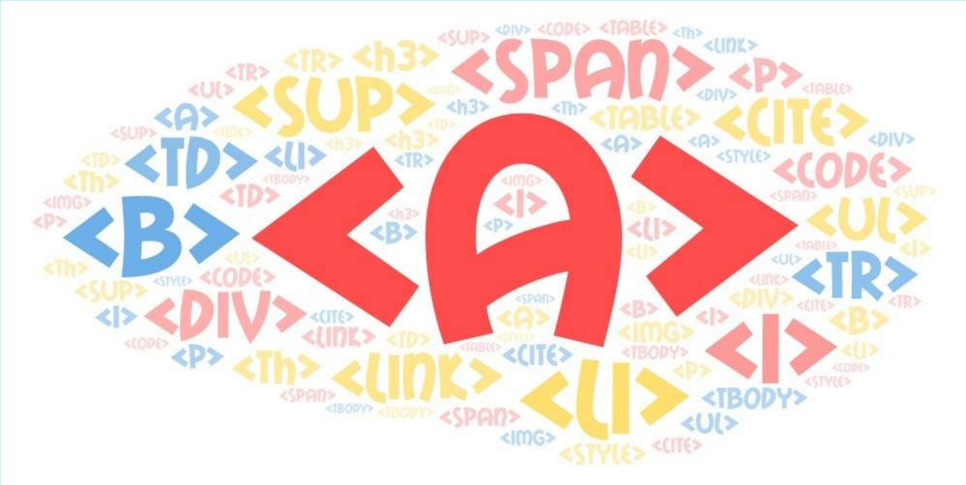
Files

home.html

No files here

100MiB maximum size

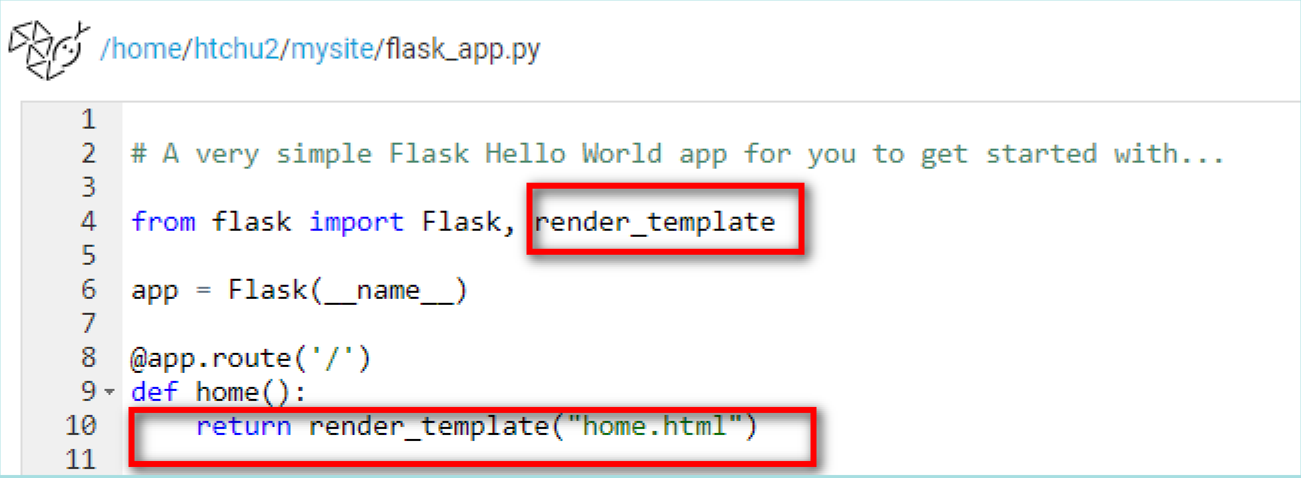
Step 8: edit home.html



- <h1>Top 10 HTML tags</h1> <br><br><br>In this article we will explain our list of top 10 HTML tags.<br><br> <h2>First tag is &lt;b&gt;&lt;/b&gt;</h2>Bold is all about making words more important.

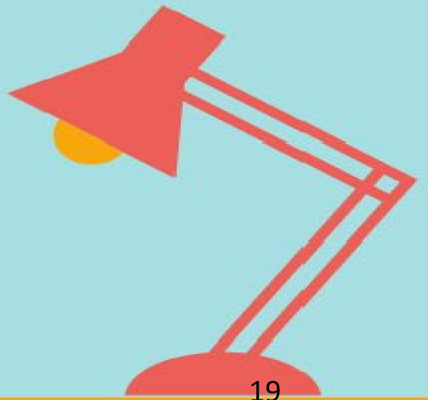
https://www.codebrainer.com/blog/top-10-html-tags

Step 9: edit flask_app.py

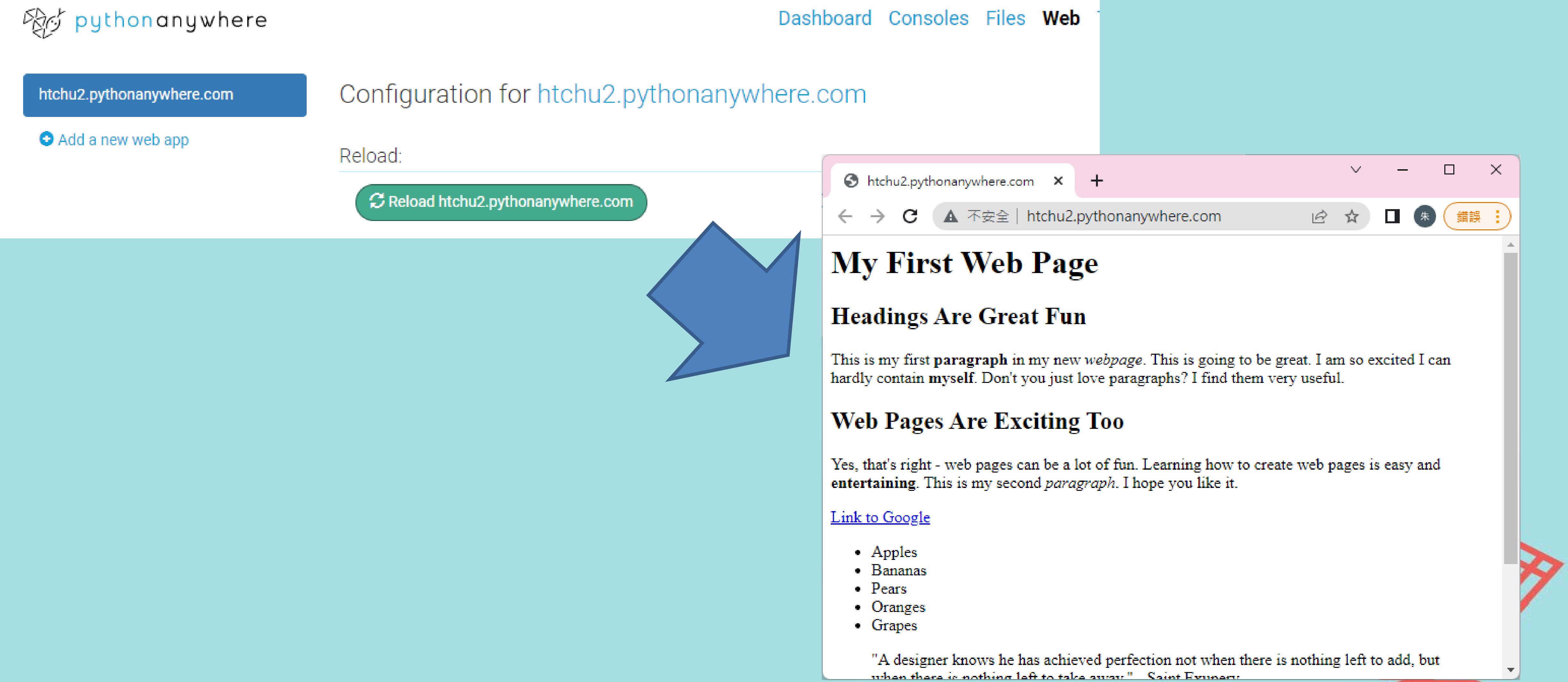


/home/htchu2/mysite/flask_app.py

```

# A very simple Flask Hello World app for you to get started with...

from flask import Flask, render_template

app = Flask(__name__)

@app.route('/')
def home():
    return render_template("home.html")

```



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
不安全 | htchu2.pythonanywhere.com
錯誤

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 Exupery

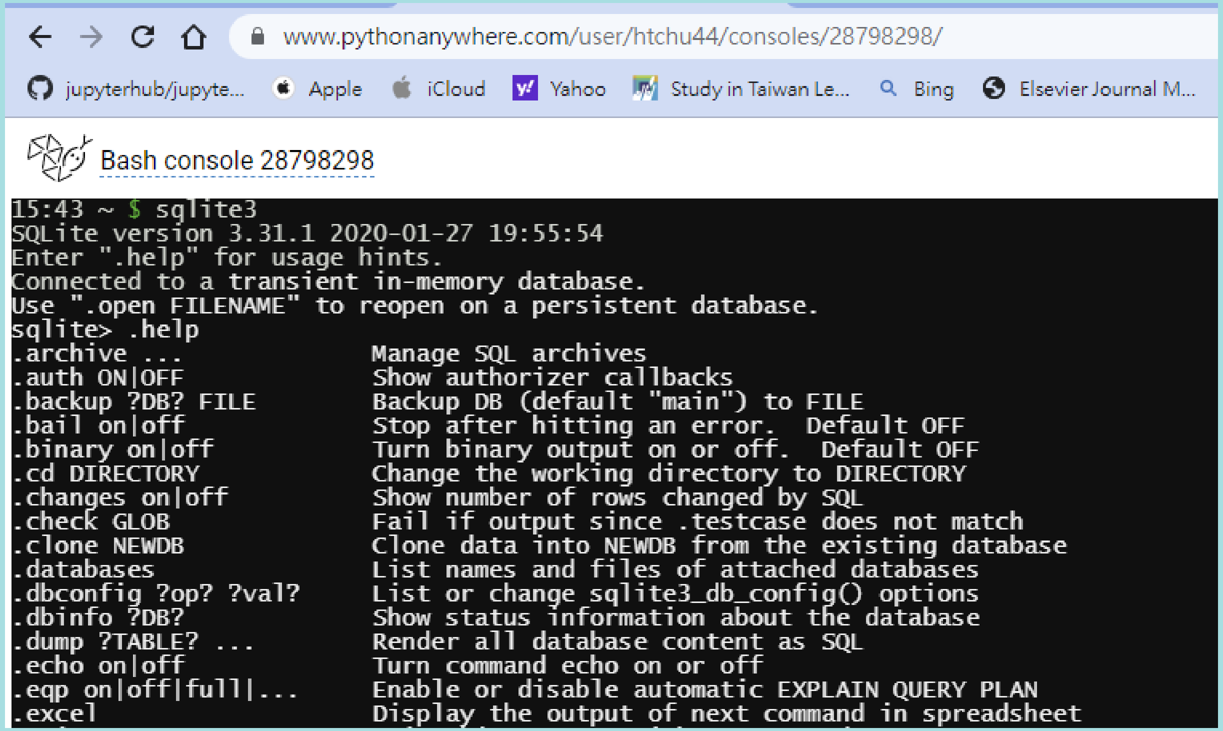
SQLite

- SQLite是小型關聯式數據庫管理系統，它包含在一個相對小的C程式庫中。與許多其它數據庫管理系統不同，SQLite不是一個客戶端/伺服器結構的數據庫引擎，而是被整合在用戶程式中。

- SQLite實現了大多數SQL標準。它使用動態的、弱類型的SQL語法。它作為嵌入式數據庫，是應用程式，如網頁瀏覽器，在本地/客戶端儲存資料的常見選擇。它可能是最廣泛部署的數據庫引擎，因為它正在被一些流行的瀏覽器、作業系統、嵌入式系統所使用。

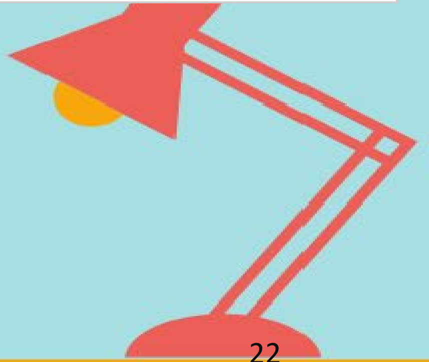
- SQLite是D. Richard Hipp建立的。

Meta Commands



```
15:43 ~ $ sqlite3
SQLite version 3.31.1 2020-01-27 19:55:54
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .help
.archive ...             Manage SQL archives
.auth ON|OFF             Show authorizer callbacks
.backup ?DB? FILE        Backup DB (default "main") to FILE
.bail on|off             Stop after hitting an error.  Default OFF
.binary on|off           Turn binary output on or off.  Default OFF
.cd DIRECTORY            Change the working directory to DIRECTORY
.changes on|off          Show number of rows changed by SQL
.check GLOB              Fail if output since .testcase does not match
.clone NEWDB             Clone data into NEWDB from the existing database
.databases               List names and files of attached databases
.dbconfig ?op? ?val?     List or change sqlite3_db_config() options
.dbinfo ?DB?             Show status information about the database
.dump ?TABLE? ...        Render all database content as SQL
.echo on|off             Turn command echo on or off
.eqp on|off|full|...     Enable or disable automatic EXPLAIN QUERY PLAN
.excel                   Display the output of next command in spreadsheet
```

Command	Description
.show	Displays current settings
.databases	Provides database names and files
.quit	Quit sqlite3 program
.tables	Show current tables
.schema	Display schema of table
.header	Display the output table header
.mode	Select mode for the output table
.dump	Dump database in SQL text format

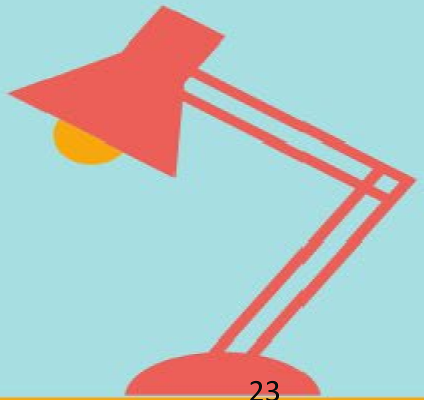


https://www.sqlite.org/cli.html

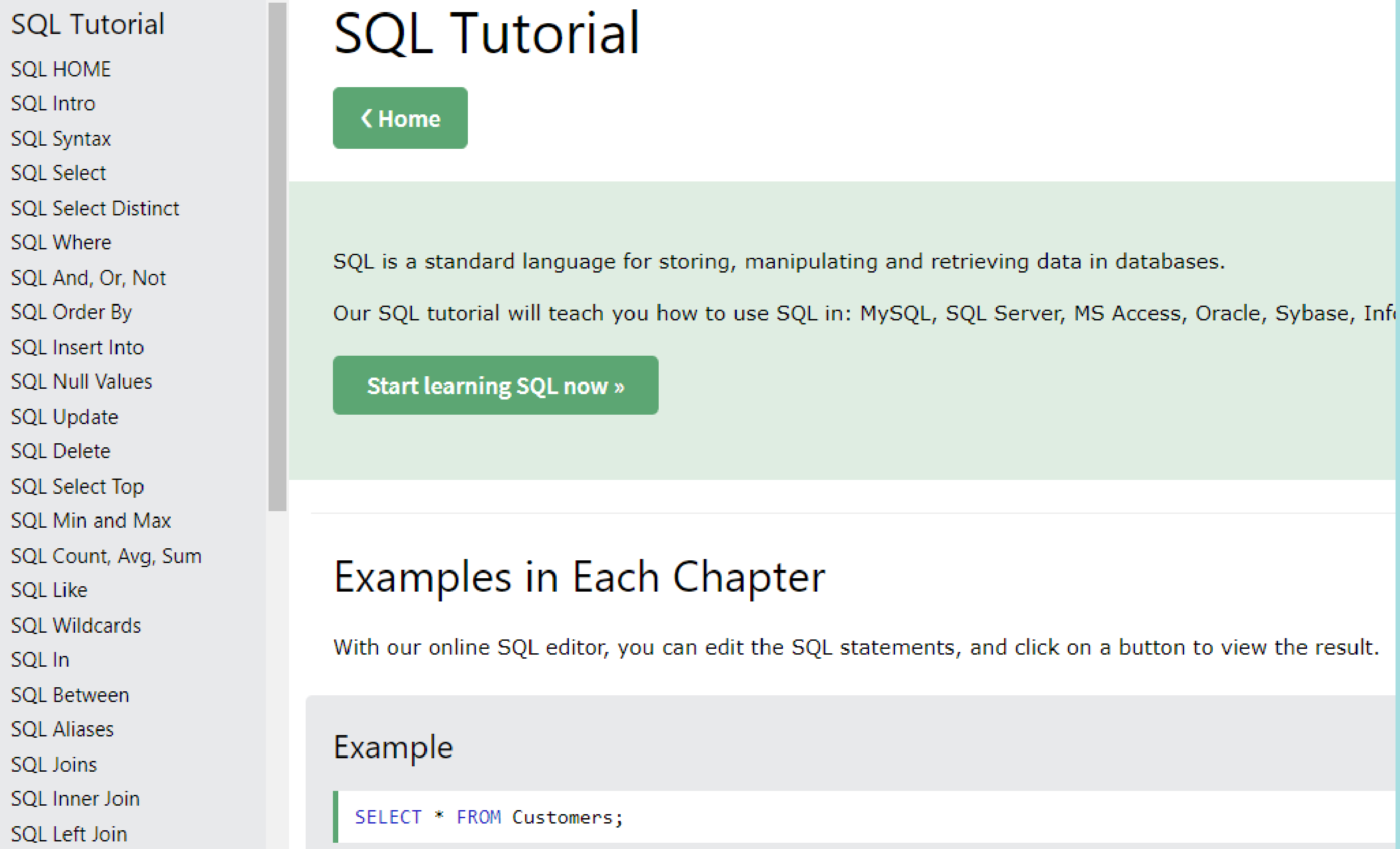
SQL(Structured Query Language) commands

case insensitive

- DDL-Data Definition Language
  - CREATE TABLE
  - DROP TABLE
  - ALTER TABLE
- DML-Data Manipulation Language
  - INSERT
  - UPDATE
  - DELETE
- DQL-Data Query Language
  - SELECT



SQL Tutorial



SQL Tutorial

SQL HOME
SQL Intro
SQL Syntax
SQL Select
SQL Select Distinct
SQL Where
SQL And, Or, Not
SQL Order By
SQL Insert Into
SQL Null Values
SQL Update
SQL Delete
SQL Select Top
SQL Min and Max
SQL Count, Avg, Sum
SQL Like
SQL Wildcards
SQL In
SQL Between
SQL Aliases
SQL Joins
SQL Inner Join
SQL Left Join

SQL Tutorial

SQL is a standard language for storing, manipulating and retrieving data in databases.

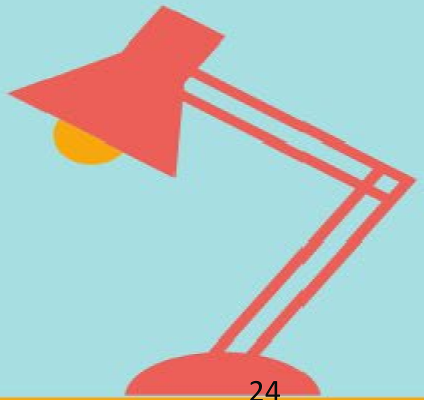
Our SQL tutorial will teach you how to use SQL in: MySQL, SQL Server, MS Access, Oracle, Sybase, Inf

Examples in Each Chapter

With our online SQL editor, you can edit the SQL statements, and click on a button to view the result.

Example

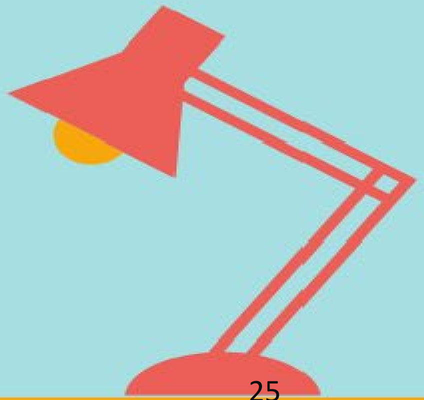
```
SELECT * FROM Customers;
```



https://www.w3schools.com/sql/

SQL Examples

```
CREATE TABLE news (
        news_id INTEGER NOT NULL PRIMARY KEY AUTOINCREMENT,
        news_caption TEXT NOT NULL,
        news_time DATETIME DEFAULT CURRENT_TIMESTAMP,
        news_url TEXT NULL);

ALTER TABLE news ADD news_txt TEXT NULL;

INSERT INTO news (news_caption, news_url)
VALUES ('N1', 'https://aaa.com/xxx111.aspx');

INSERT INTO news (news_caption, news_url)
VALUES ('N1', 'https://aaa.com/xxx111.aspx');

SELECT * FROM news;
```



Python sqlite3 -SQLite DB-API 2.0

```
import sqlite3

con = sqlite3.connect('example.db')

cur = con.cursor()

# Create table

cur.execute('CREATE TABLE stocks (date text, trans text, symbol text, qty real, price real)')

# Insert a row of data

cur.execute("INSERT INTO stocks VALUES ('2006-01-05','BUY','RHAT',100,35.14)")

# Save (commit) the changes

con.commit()

# We can also close the connection if we are done with it.

# Just be sure any changes have been committed or they will be lost.

con.close()
```

https://docs.python.org/zh-tw/3/library/sqlite3.html

Thanks! Q&A