



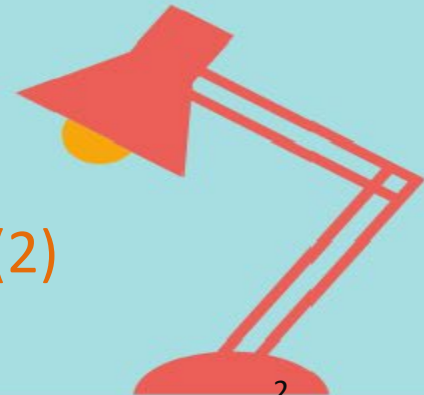
# 111-2進階程式設計課程(13)

## Advanced Computer Programming

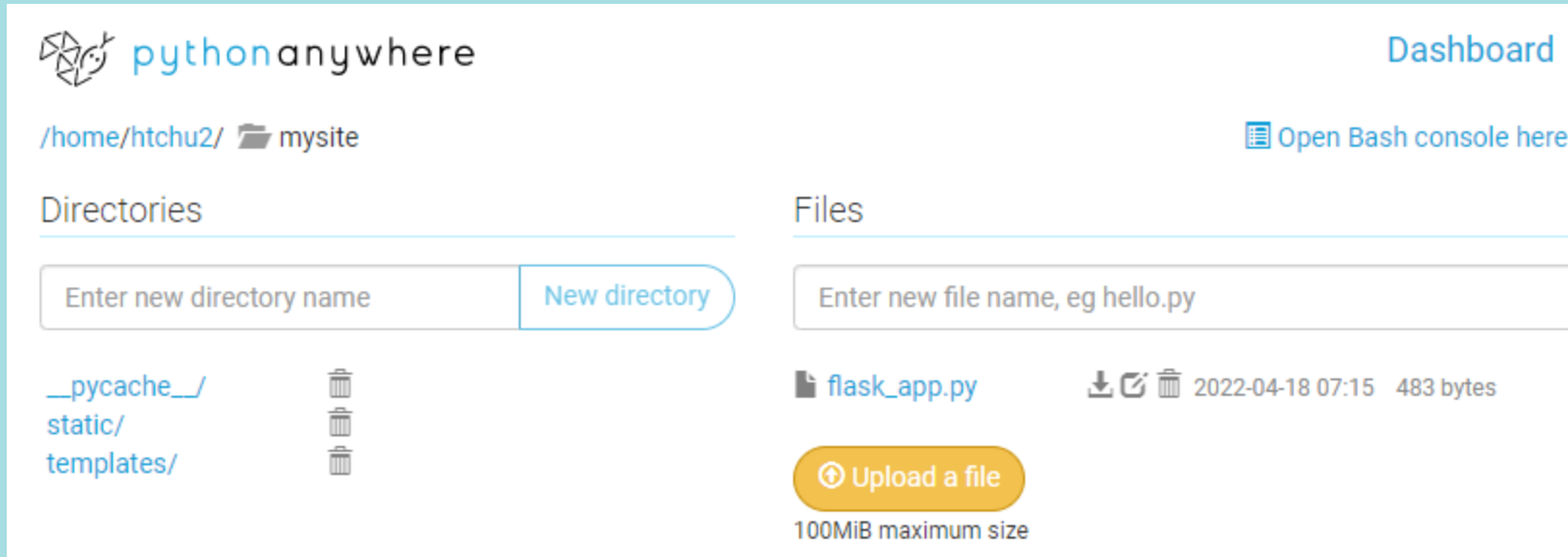
亞大資工系

# 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



# 1. Flask framework



The screenshot displays the PythonAnywhere web interface. At the top left is the PythonAnywhere logo. The top right shows a 'Dashboard' link. Below the logo, the current path is '/home/htchu2/ mysite'. A link 'Open Bash console here' is also present. The interface is divided into two main sections: 'Directories' and 'Files'. The 'Directories' section has a text input 'Enter new directory name' and a 'New directory' button. It lists three directories: '\_\_pycache\_\_/', 'static/', and 'templates/', each with a trash icon. The 'Files' section has a text input 'Enter new file name, eg hello.py'. It shows a file named 'flask\_app.py' with download, share, and delete icons, and a timestamp of '2022-04-18 07:15' and size of '483 bytes'. Below this is an 'Upload a file' button and a note '100MiB maximum size'.

pythonanywhere

Dashboard

/home/htchu2/ mysite

Open Bash console here

Directories

Enter new directory name

New directory

\_\_pycache\_\_/

static/

templates/

Files

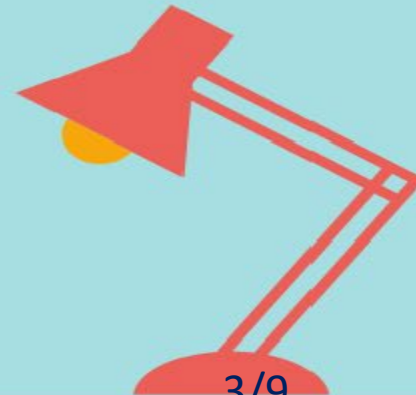
Enter new file name, eg hello.py

flask\_app.py

2022-04-18 07:15 483 bytes

Upload a file

100MiB maximum size



# Basic Flask framework

```
# main.py

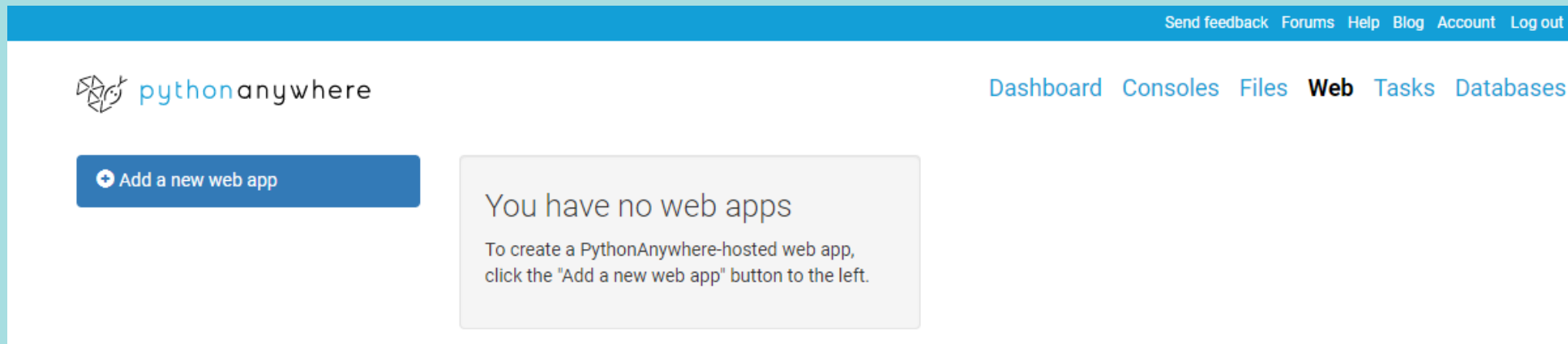
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return 'Hello, World!'
```



- (1) The line `app = Flask(__name__)`, `__name__` is used to locate the location of the currently loaded folder, which is used to determine the location of the `template` folder or `static` folder.
- (2) `@app.route('/')` A decorator that is used to register a view function for a given URL rule. This does the same thing as `add_url_rule()` but is intended for decorator usage

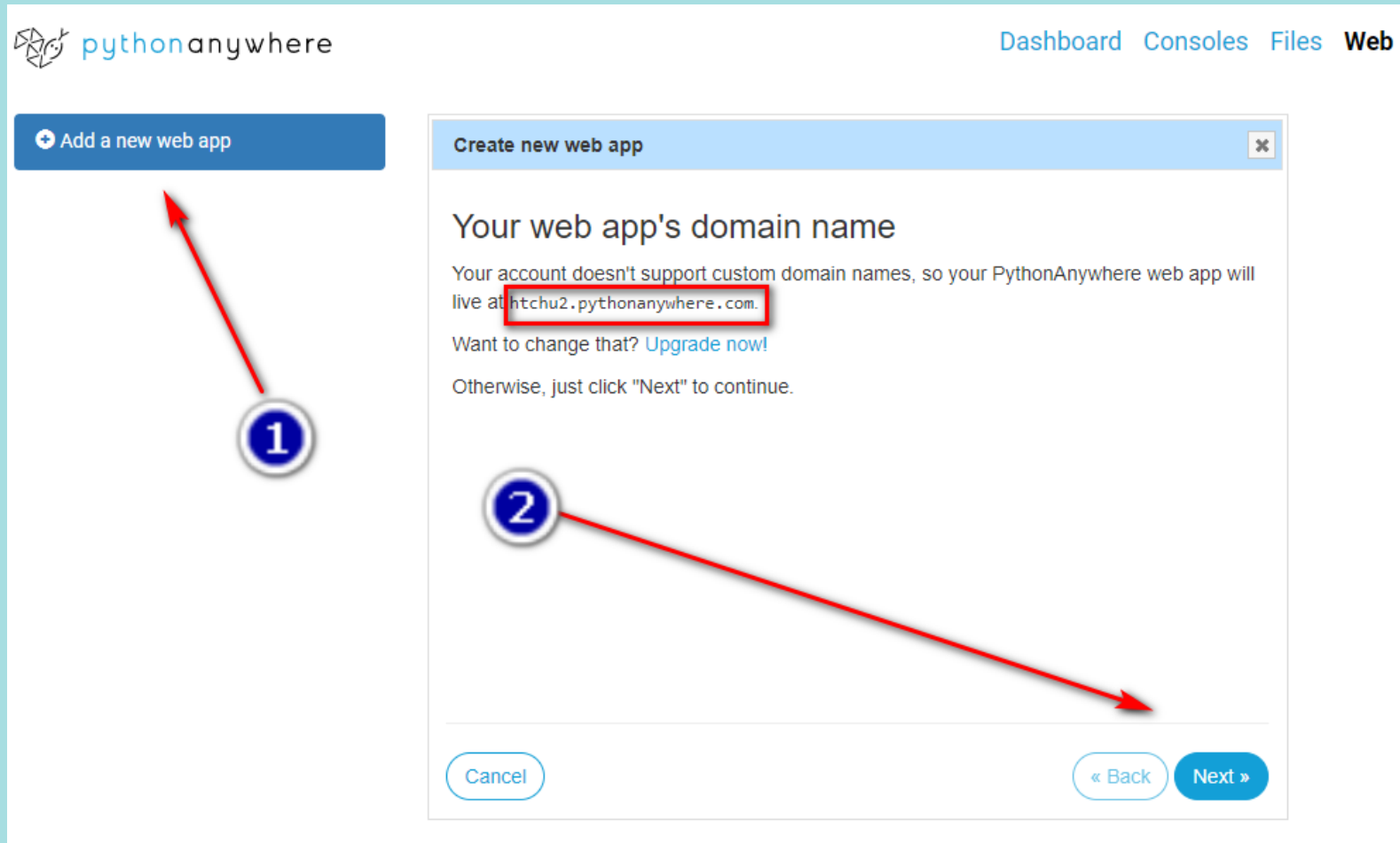
# Step 1: Go to Web tab



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.'



# 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 sidebar, 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. In the center, a modal window titled 'Create new web app' is open. Inside the modal, the heading is 'Your web app's domain name'. Below it, a message states: 'Your account doesn't support custom domain names, so your PythonAnywhere web app will live at `htchu2.pythonanywhere.com`'. The domain name is highlighted with a red box. Below this, it says 'Want to change that? [Upgrade now!](#)'. Further down, it says '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.

pythonanywhere

Dashboard Consoles Files Web

+ Add a new web app

Create 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.

Cancel « Back Next »



# 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**

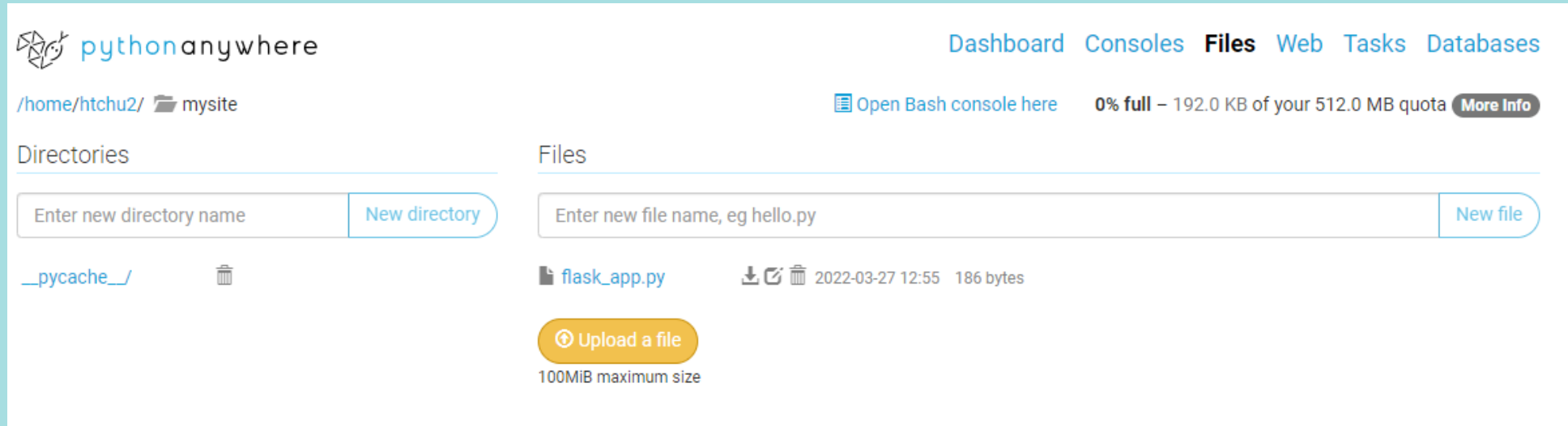
Run until 3 months from today

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






# Step 5: check the files



The screenshot shows 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 status bar indicating **0% full** – 192.0 KB of your 512.0 MB quota, with a link to [Open Bash console here](#) and a **More Info** button. The main content area is divided into two sections: Directories and Files. The Directories section has a text input field for "Enter new directory name" and a "New directory" button. Below this, a directory named `__pycache__` is listed with a trash icon. The Files section has a text input field for "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 section, there is an "Upload a file" button and a note "100MiB maximum size".


pythonanywhere

Dashboard Consoles **Files** Web Tasks Databases

[/home/htchu2/](#)  mysite [Open Bash console here](#) **0% full** – 192.0 KB of your 512.0 MB quota [More Info](#)





Directories

Enter new directory name [New directory](#)

`__pycache__` 

Files

Enter new file name, eg hello.py [New file](#)


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

[Upload a file](#)

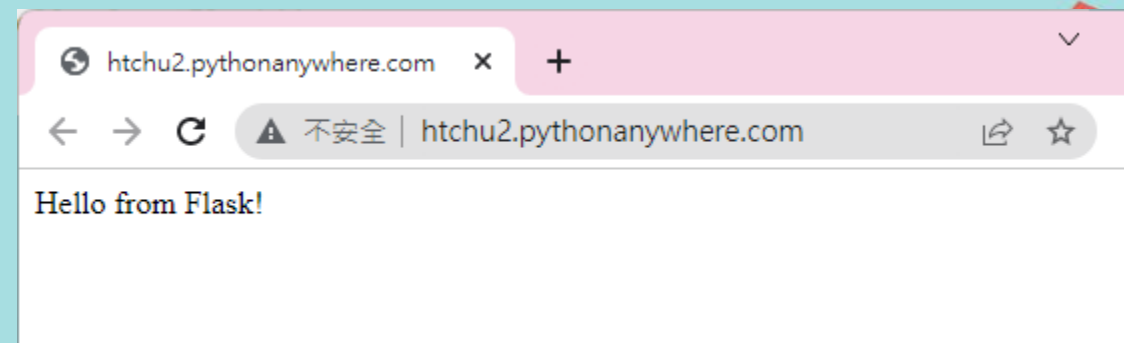
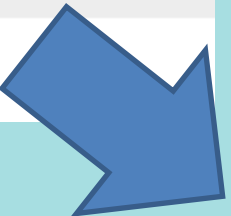
100MiB maximum size



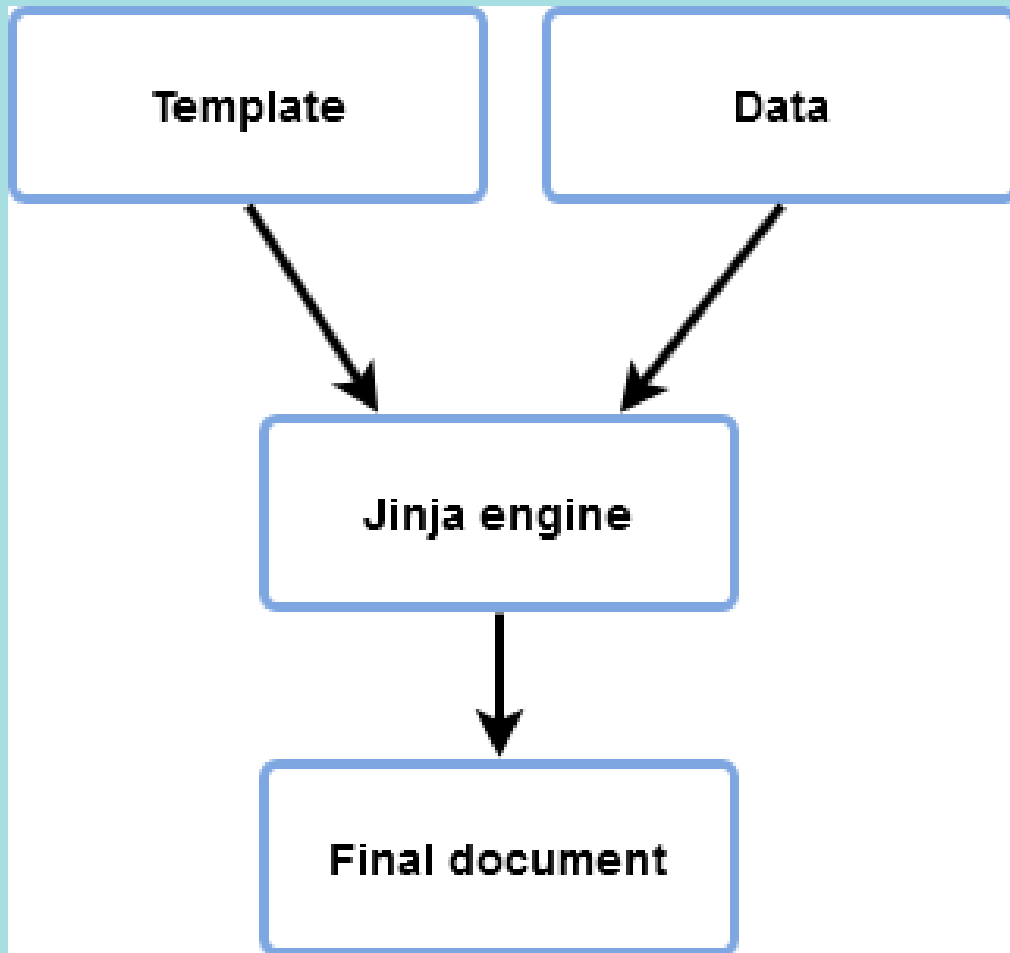
# Step 6: check the program and the web app

 `/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
```



## 2. Jinja template engine

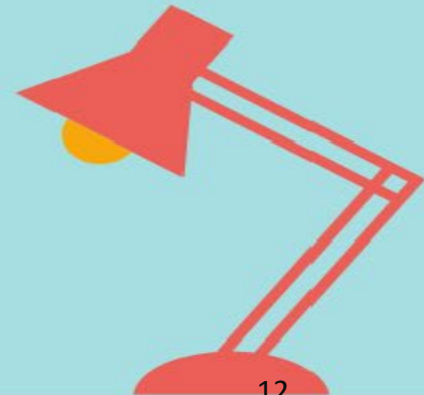


- Jinja2 essentially needs two source ingredients, template and data that will be used to render the final document.



# Jinja-template engine

- **delimiters**
  - `{% ... %}` for Statements
  - `{{ ... }}` for Expressions to print to the template output
  - `{# ... #}` for Comments not included in the template output
  - `# ... ##` for Line Statements
- **extensions**
  - `{% extends "filename" %}`
- **blocks**
  - `{% block blockname %}`
  - `{% endblock %}`



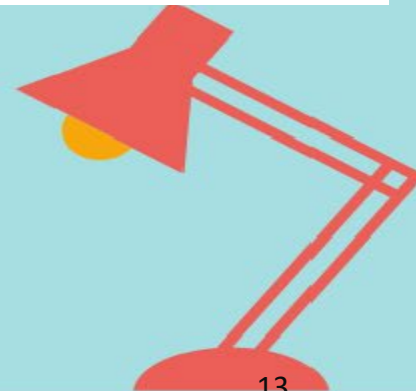
# A template

```
<h1>{{ username }}'s profile</h1>
{% if bio %}
    <p>{{ bio }}</p> {# The indentation here is for readability only, not
necessary #}
{% else %}
    <p>The self-introduction is empty. </p>
{% endif %} {# Most Jinja statements need to declare closing #}
```

{{ ... }} is used to mark variables.

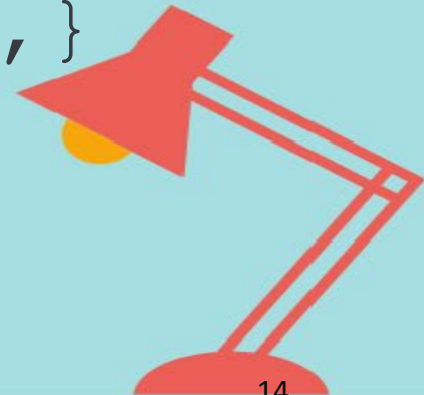
{% ... %} is used to mark statements, such as if statements, for statements, etc.

{# ... #} is used to write comments.

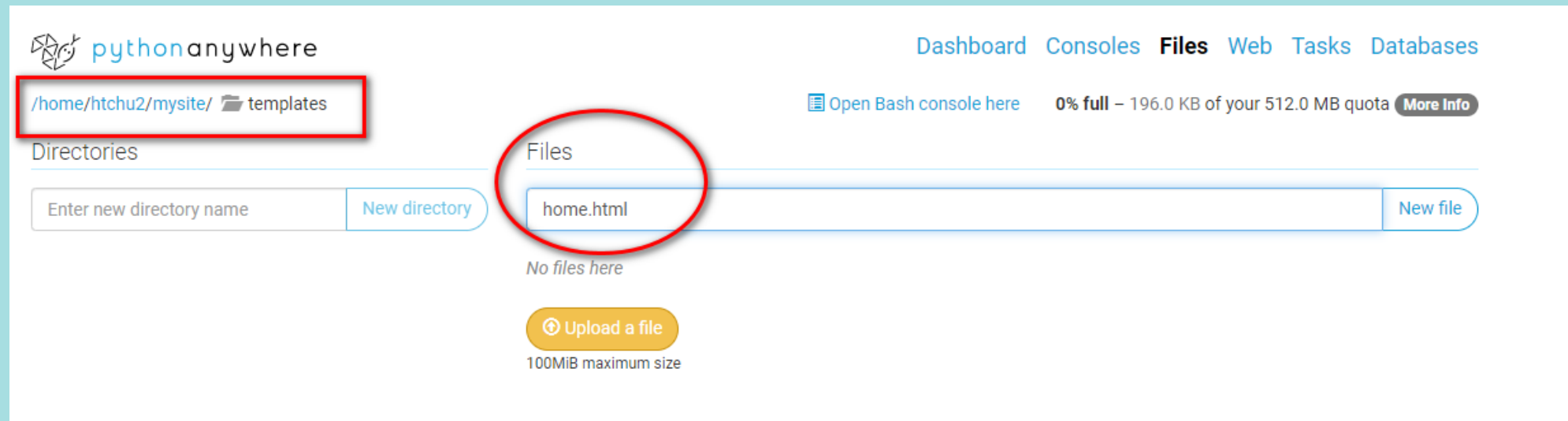
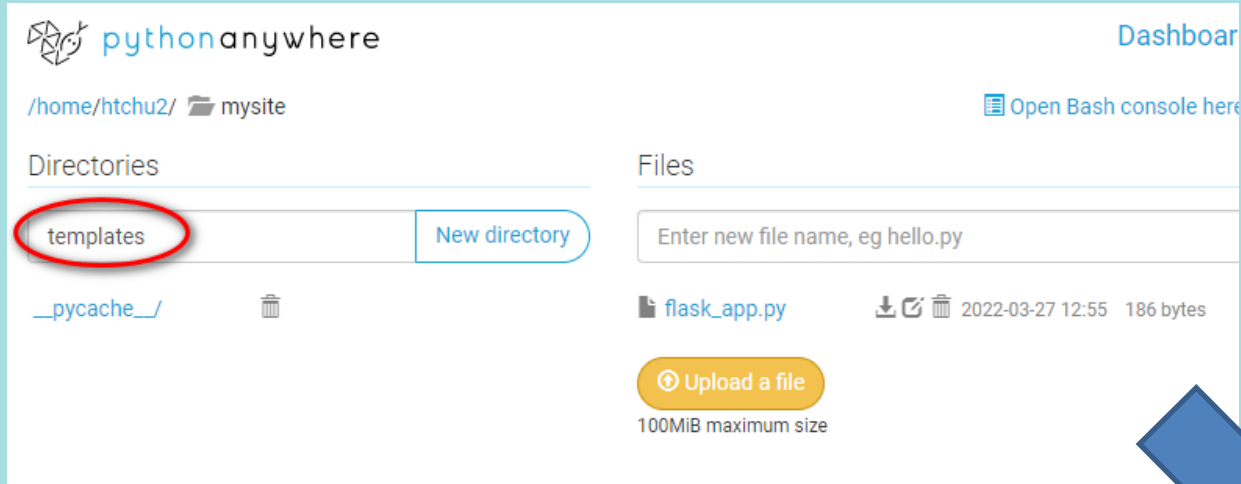


# Data

- `data = {`
- `"name": "Hsueh-Ting Chu",`
- `"expertise": "University teacher",`
- `"location": "Taichung, Taiwan",`
- `"web": "http://htchu.pythonanywhere.com/",`
- `"github": "https://github.com/htchu/", }`



# Step 7: add html templates



# Step 8: edit home.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
  <title>Bootstrap-Flask Demo Application</title>
  <!-- Latest compiled and minified CSS -->
  <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg320mUcww7on3RYdg4Va+PmSTsz/K68vbdEjh4u"
crossorigin="anonymous">
  <link rel=stylesheet type=text/css href="{{ url_for('static',
filename='style.css') }}">
</head>
<body>
```



# Step 9: create a static folder and add a css file

pythonanywhere

Dashboard Consoles **Files** Web Tasks Databases

/home/htchu/mysite/ **static** 1

Open Bash console here 15% full – 78.2 MB of your 512.0 MB quota [More Info](#)

Directories

Enter new directory name [New directory](#)

Files

Enter new file name, eg hello.py [New file](#)

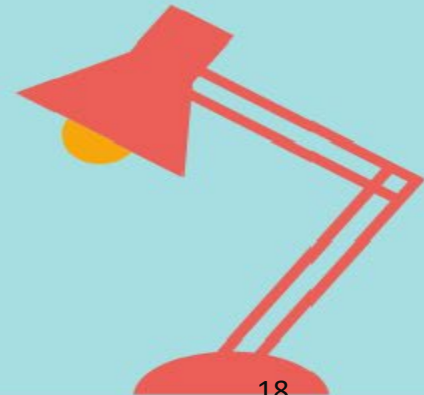
**style.css** 2 2022-04-19 00:31 952 bytes

[Upload a file](#)


100MiB maximum size

# Step 10: edit style.css


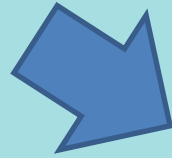
```
body{  
    margin-top:20px;  
    color: #1a202c;  
    text-align: left;  
    background-color: #e2e8f0;  
}  
.main-body {  
    padding: 15px;  
}
```



# Step 11: 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  
5  
6 app = Flask(__name__)  
7  
8 @app.route('/')  
9 def hello_world():  
10     return 'Hello from Flask!'  
11  
12
```

 /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 12: Reload web

 pythonanywhere Dashboard Consoles Files **Web**

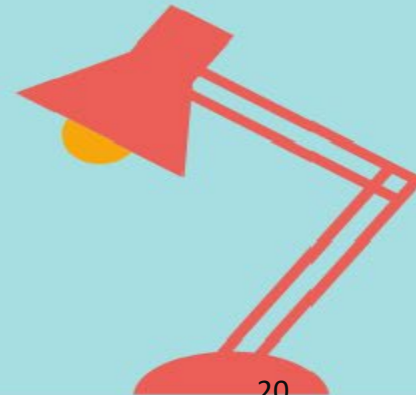
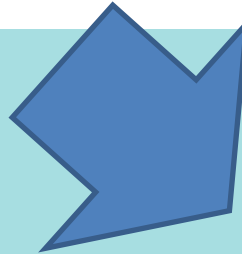
[htchu2.pythonanywhere.com](#)

[+ Add a new web app](#)

Configuration for [htchu2.pythonanywhere.com](#)

Reload:

[↻ Reload htchu2.pythonanywhere.com](#)



# The Jinja2 Template Engine

```
{% extends "layout.html" %}
{% block body %}
    <ul>
        {% for user in users %}
            <li><a href="{{ user.url }}">{{ user.username }}</a></li>
        {% endfor %}
    </ul>
{% endblock %}
```

{% ... %} for Statements

{{ ... }} for Expressions to print to the template output

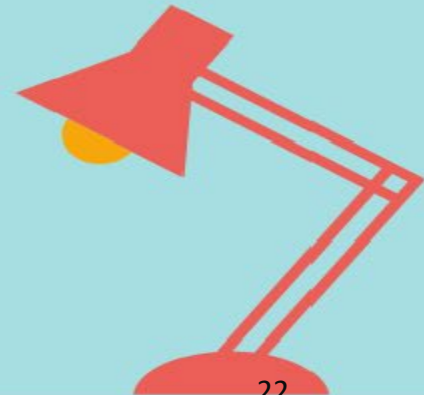
{# ... #} for Comments not included in the template output



# Variables and Tests

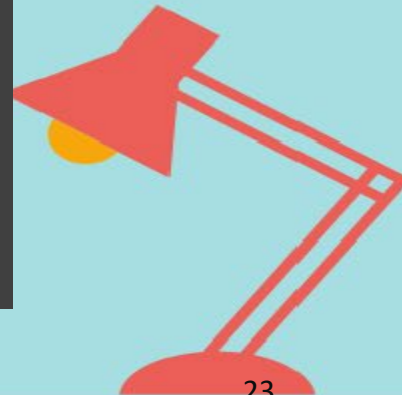
```
{{ foo.bar }}  
{{ foo['bar'] }}
```

```
{% if loop.index is divisibleby 3 %}  
{% if loop.index is divisibleby(3) %}
```



# Template and Block

```
{% extends "base.html" %}
{% block title %}Index{% endblock %}
{% block head %}
    {{ super() }}
    <style type="text/css">
        .important { color: #336699; }
    </style>
{% endblock %}
{% block content %}
    <h1>Index</h1>
    <p class="important">
        Welcome to my awesome homepage.
    </p>
{% endblock %}
```





Thanks!

Q&A

