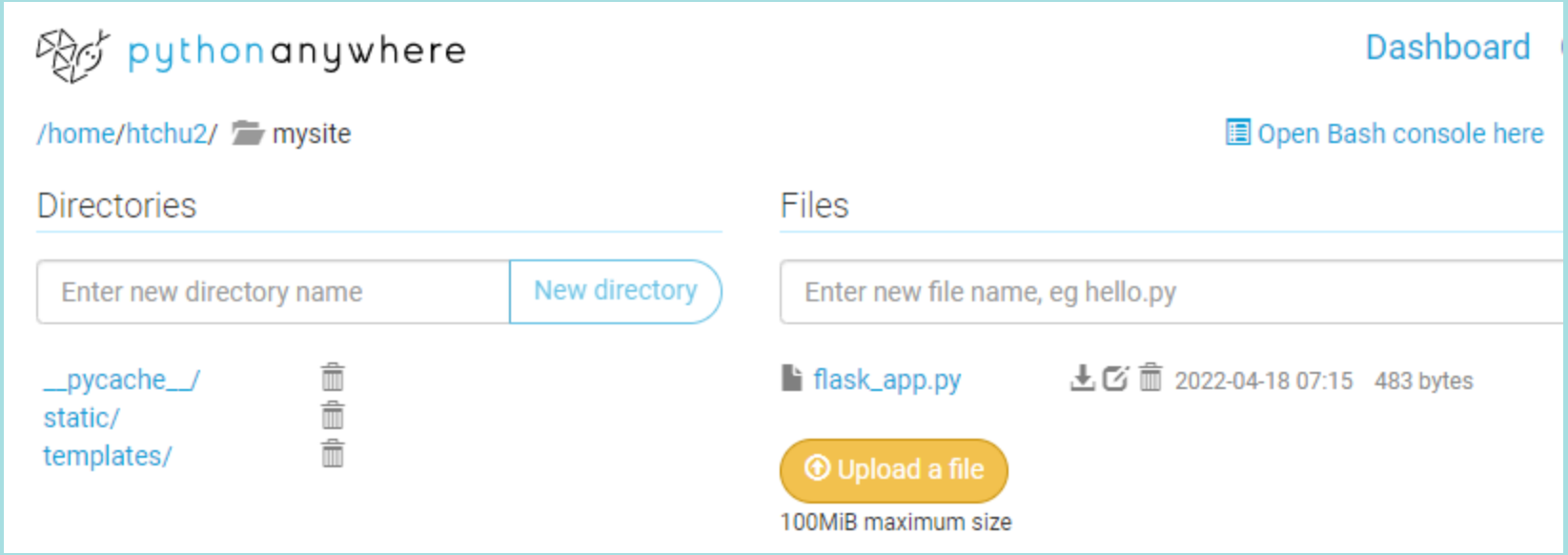
111-2進階程式設計課程(13)
Advanced Computer Programming

亞大資工系

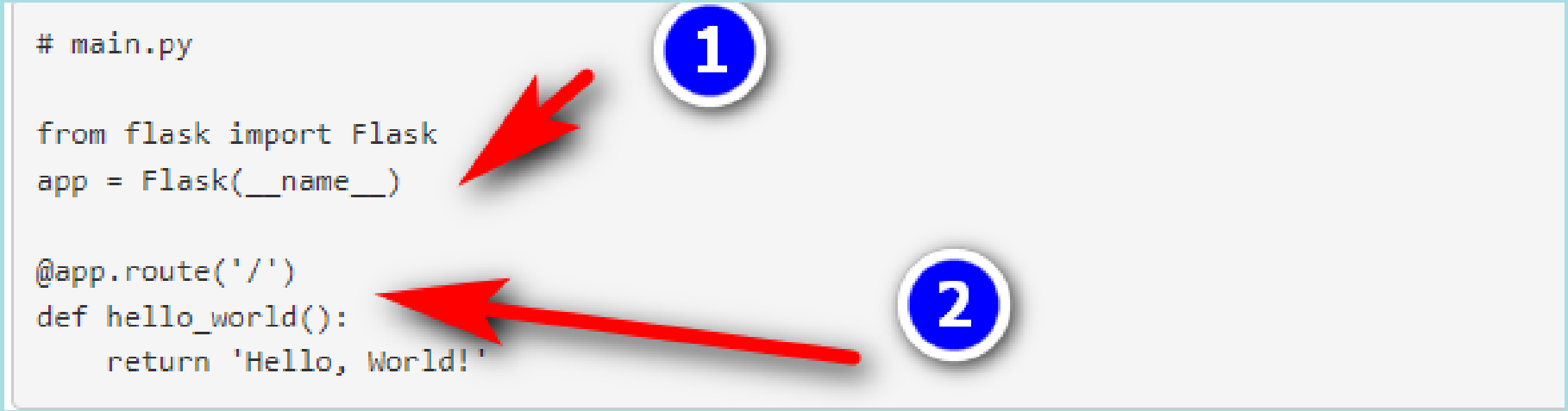
課程大綱

- 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



Basic Flask framework

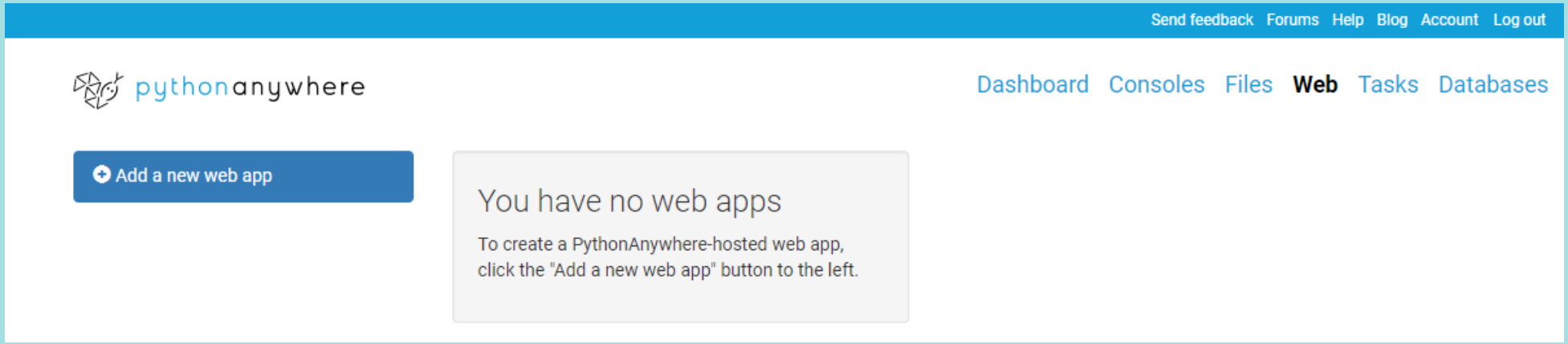



```
# main.py

from flask import Flask
app = Flask(__name__)

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

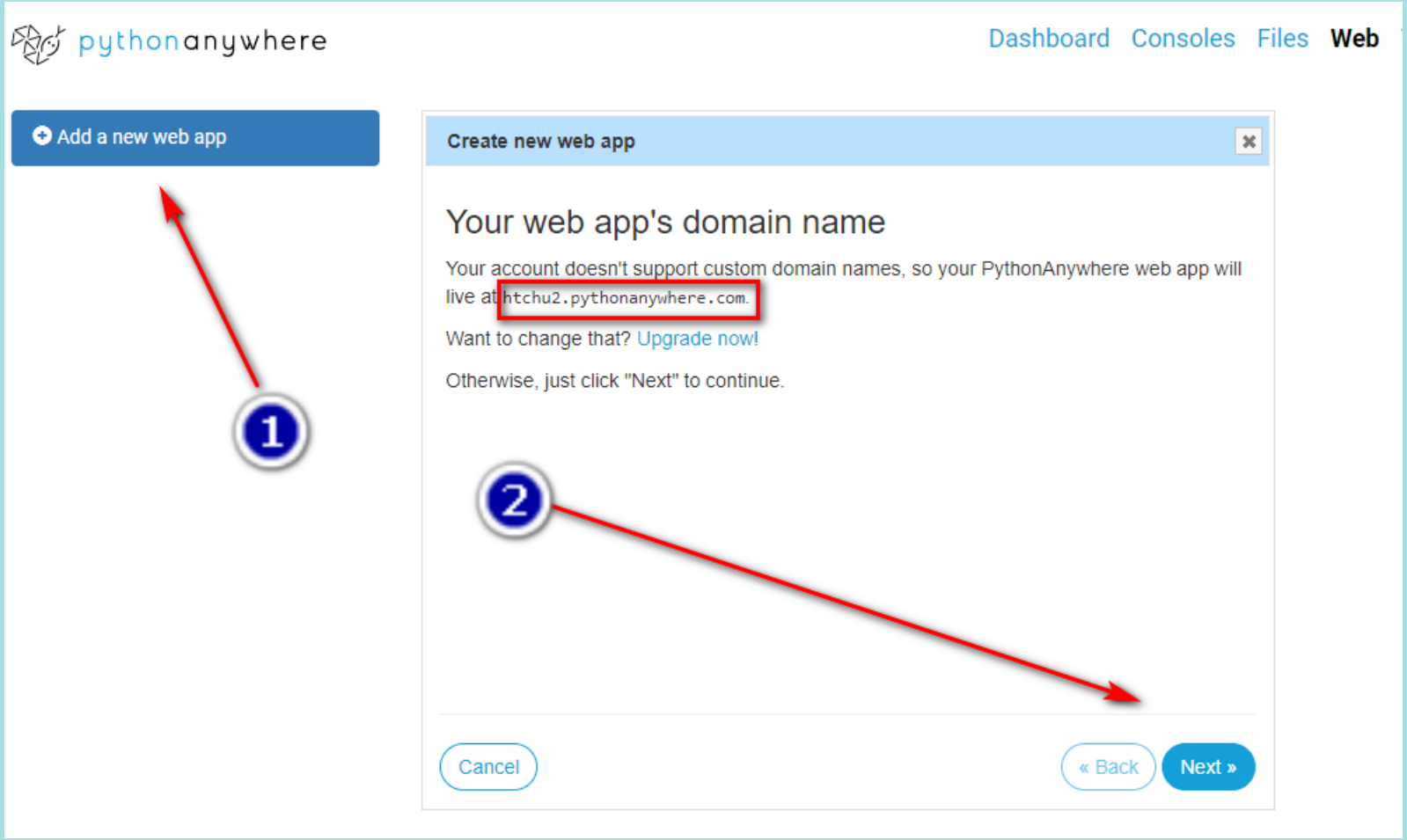
(1) The line app = Flask(__name__), __name__ is used to locate the location of the currently loaded folder, which is used to determine the location of the template folder or static folder.

(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



Step 2: Add a new web app



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)

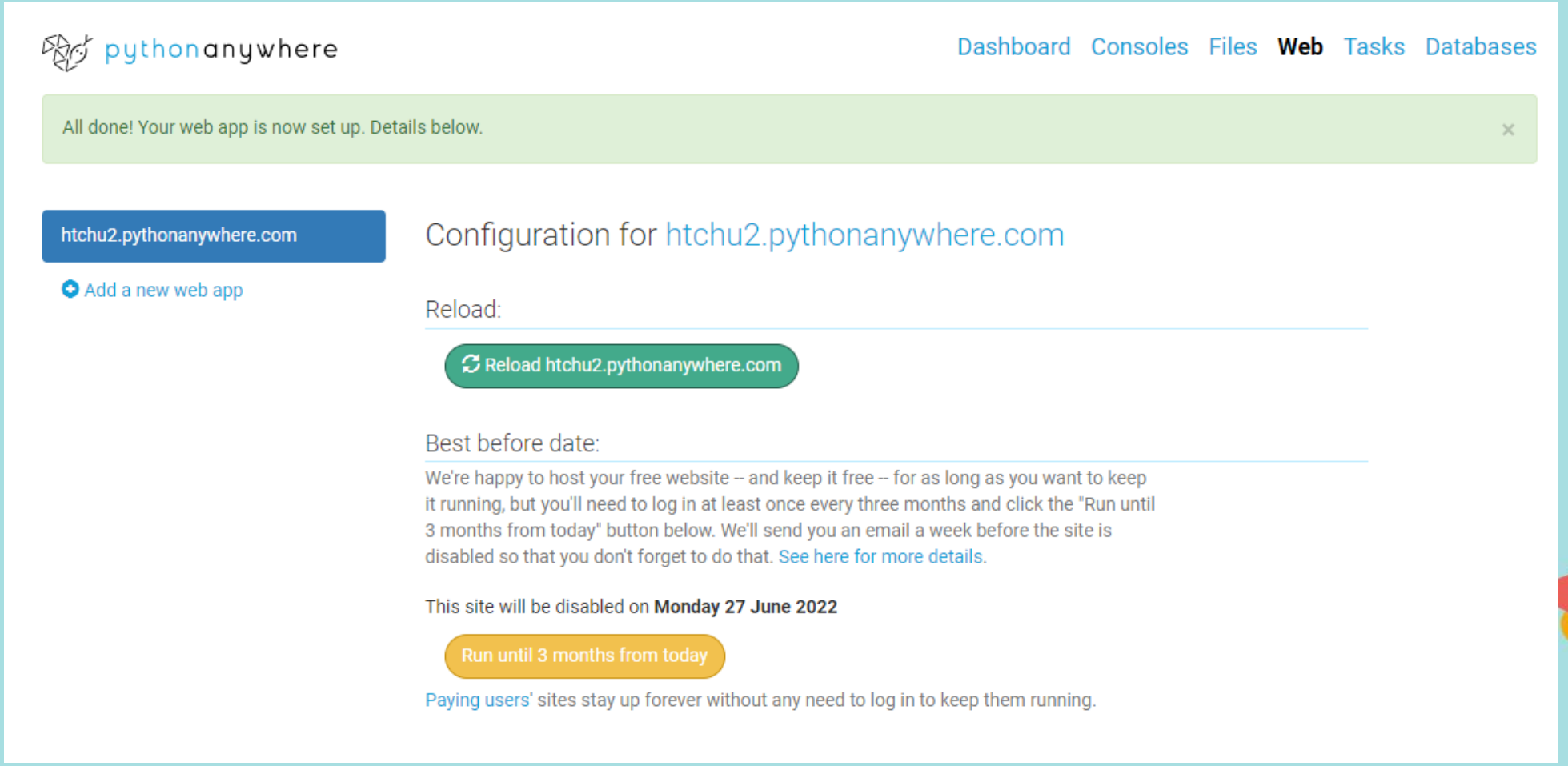
What other frameworks should we have here? Send us some feedback using the link at the top of the page!

Select a Python version

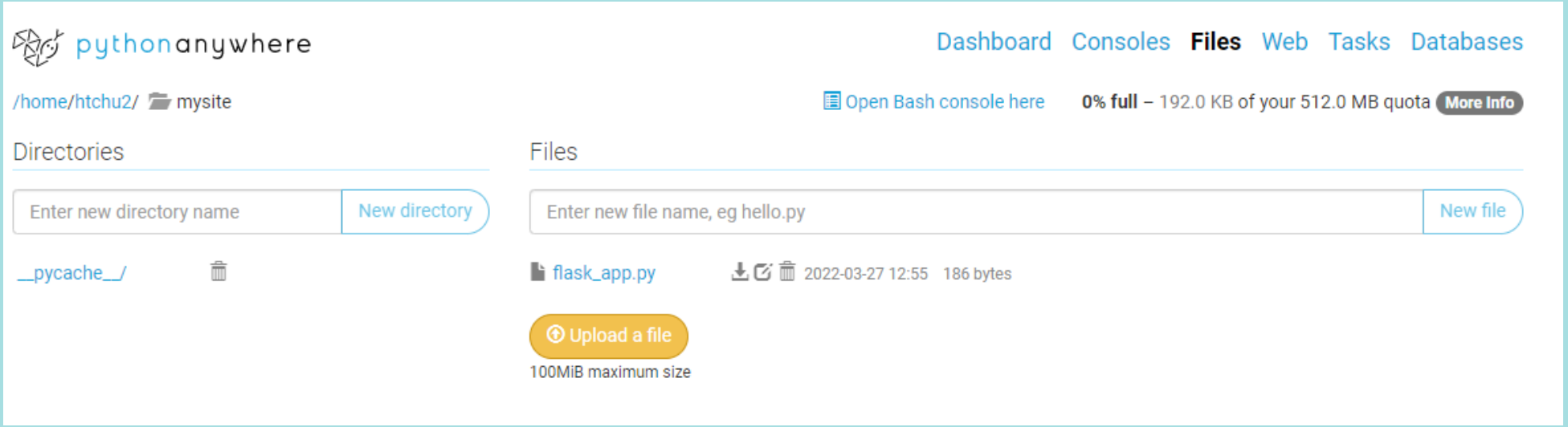
- » **Python 3.6** (Flask 2.0.0)
- » **Python 3.7** (Flask 2.0.0)
- » **Python 3.8** (Flask 2.0.0)
- » **Python 3.9** (Flask 2.0.0)

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



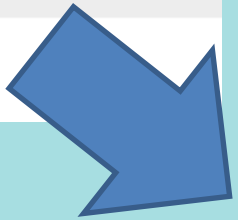
Step 5: check the files

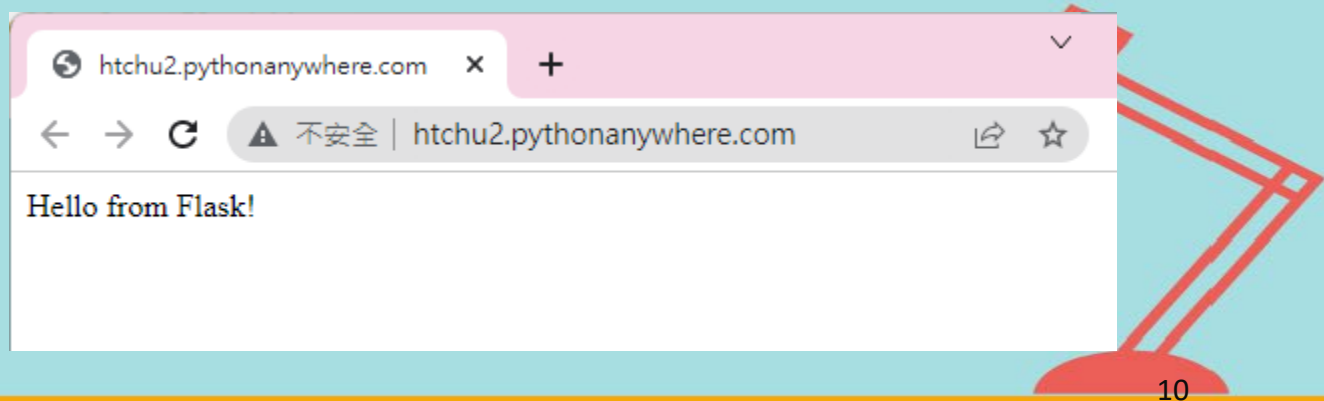


Step 6: check the program and the web app

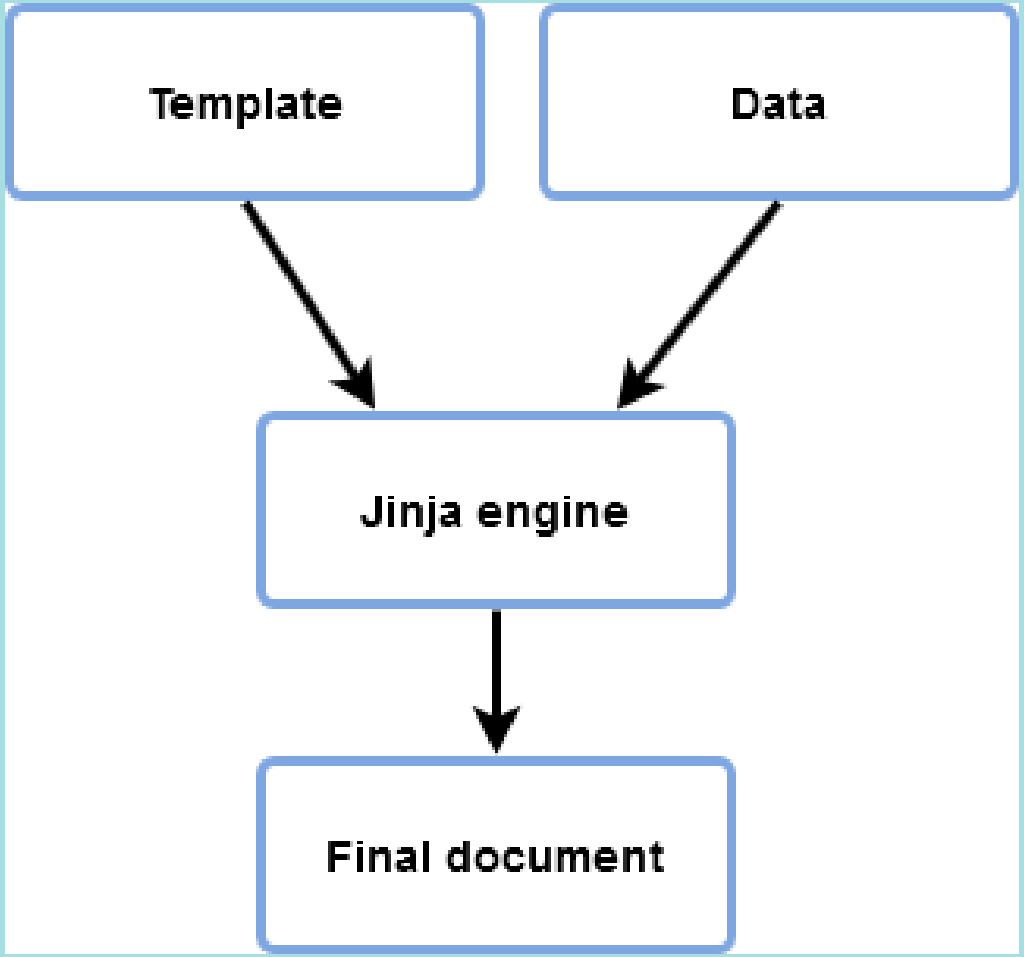
/home/htchu2/mysite/flask_app.py

```
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

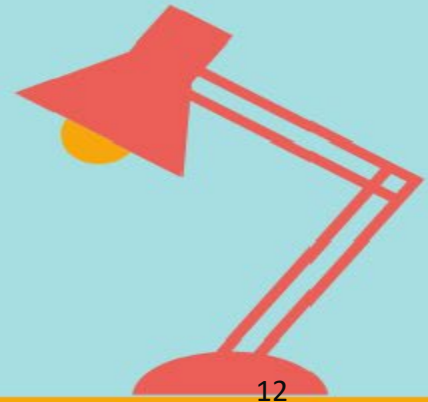


- Jinja2 essentially needs two source ingredients, template and data that will be used to render the final document.

https://ttl255.com/jinja2-tutorial-part-1-introduction-and-variable-substitution/

Jinja-template engine

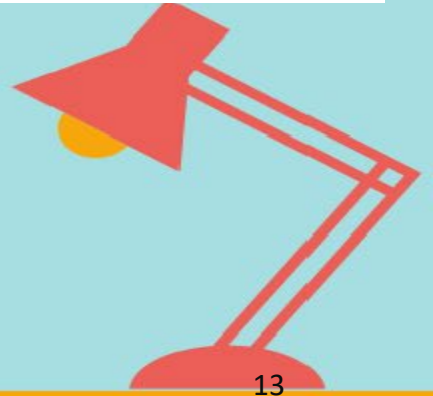
- **delimiters**
  - {% ... %} for Statements
  - {{ ... }} for Expressions to print to the template output
  - {# ... #} for Comments not included in the template output
  - # ... ## for Line Statements
- **extensions**
  - {% extends “filename" %}
- **blocks**
  - {% block blockname %}
  - {% endblock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



A template

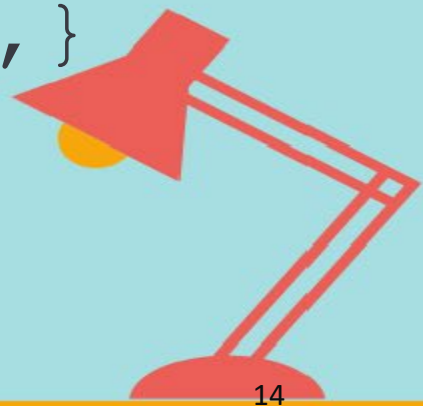
```
<h1>{{ username }}'s profile</h1>
{% if bio %}
    <p>{{ bio }}</p> {# The indentation here is for readability only, not necessary #}
{% else %}
    <p>The self-introduction is empty. </p>
{% endif %} {# Most Jinja statements need to declare closing #}
```

{{ ... }} is used to mark variables.
{% ... %} is used to mark statements, such as if statements, for statements, etc.
{# ... #} is used to write comments.

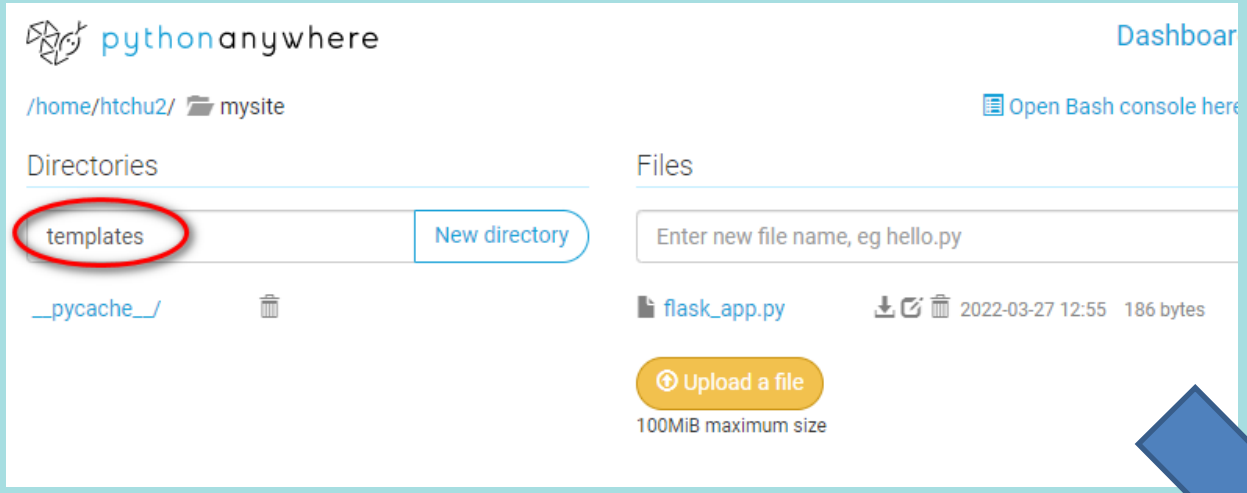


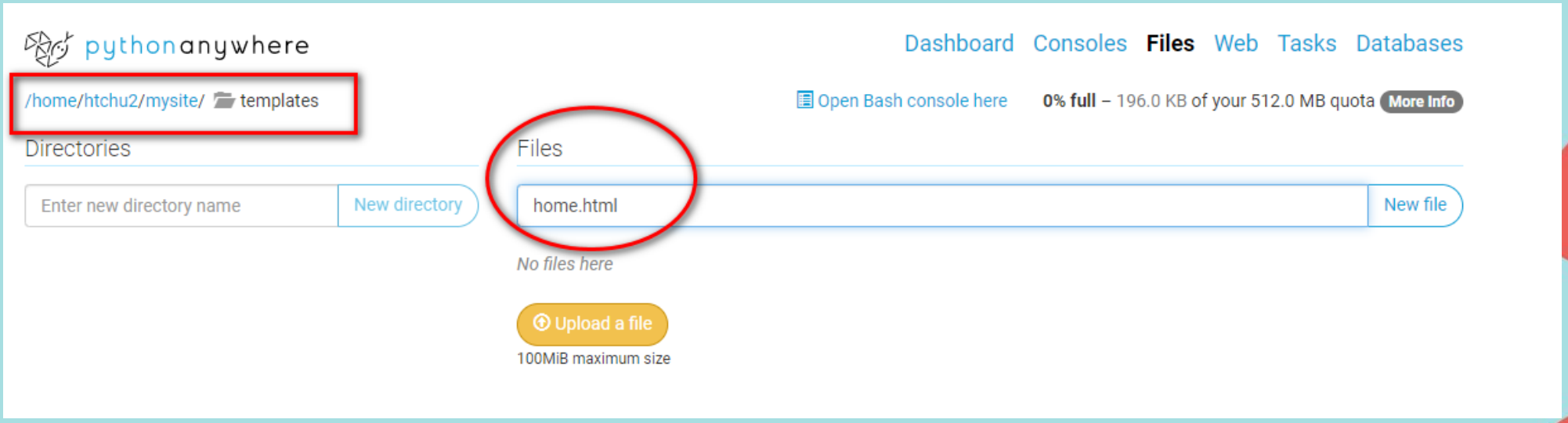
Data

- 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

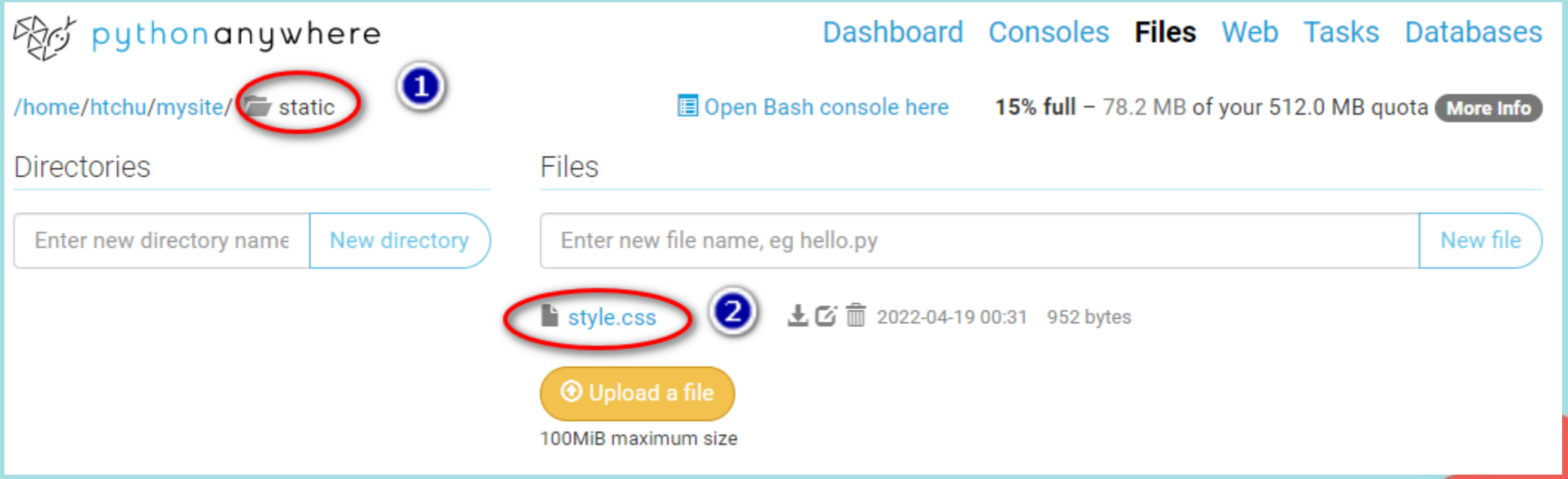




Step 8: edit home.html

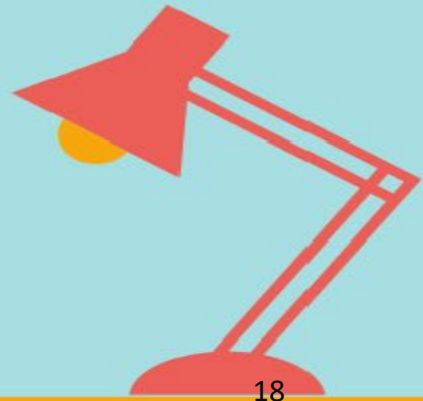
```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-
fit=no">
    <title>Bootstrap-Flask Demo Application</title>
    <!-- Latest compiled and minified CSS -->
    <link rel="stylesheet"
href="https://cdn.jsdelivr.net/npm/bootstrap@3.3.7/dist/css/bootstrap.min.css"
integrity="sha384-BVYiiSIFeK1dGmJRAkycuHAHRg32OmUcww7on3RYdg4Va+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



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

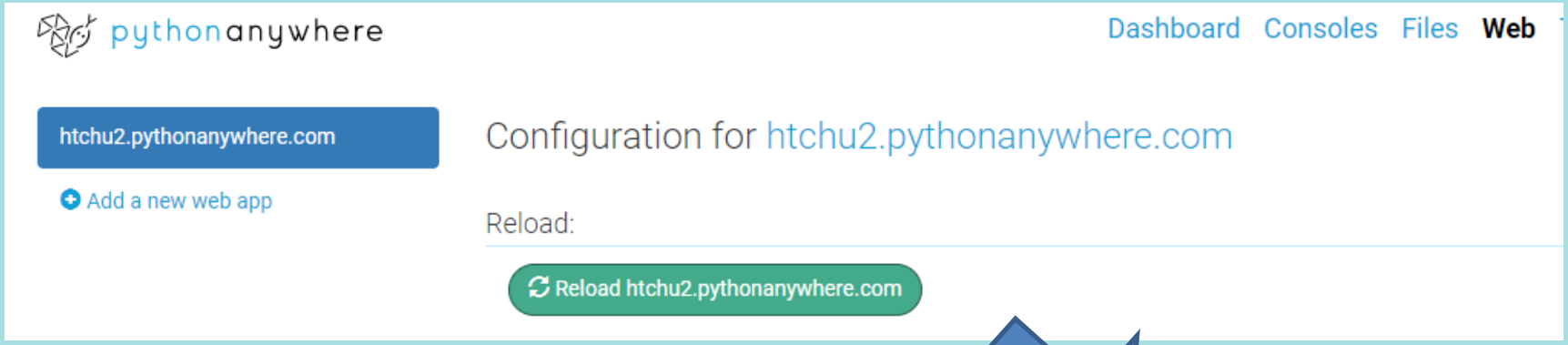
/home/htchu2/mysite/flask_app.py

```
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



The Jinja2 Template Engine

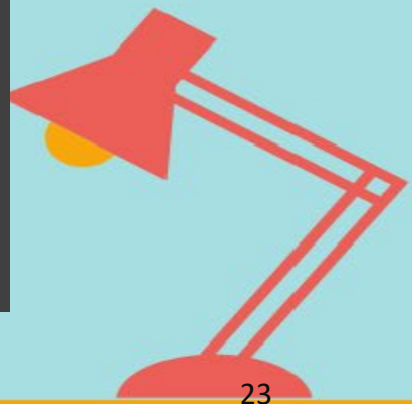
```
{% extends "layout.html" %}
{% block body %}
  <ul>
  {% for user in users %}
    <li><a href="{{ user.url }}">{{ user.username }}</a></li>
  {% endfor %}
  </ul>
{% endblock %}



{% ... %} 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