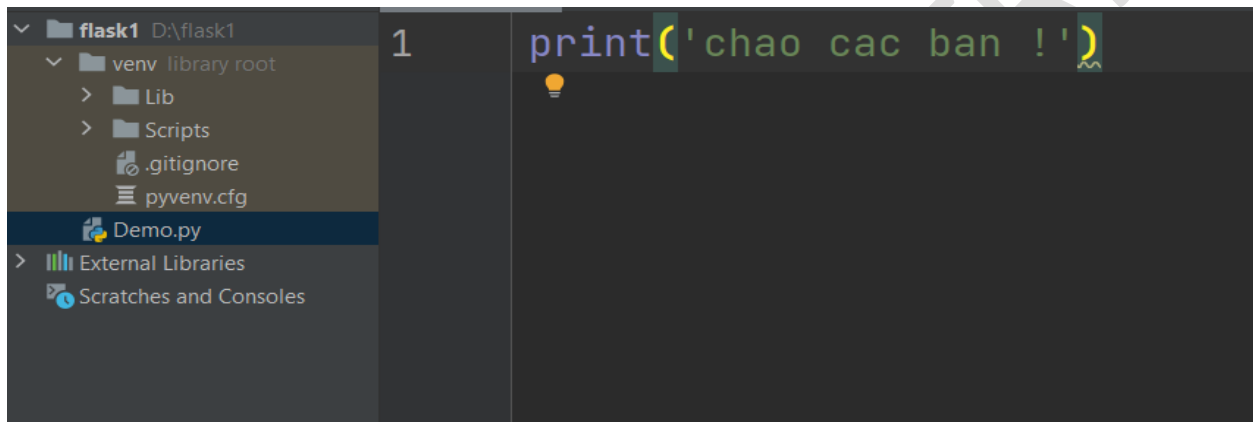


# Hướng dẫn các bước tạo ứng dụng Flask đầu tiên

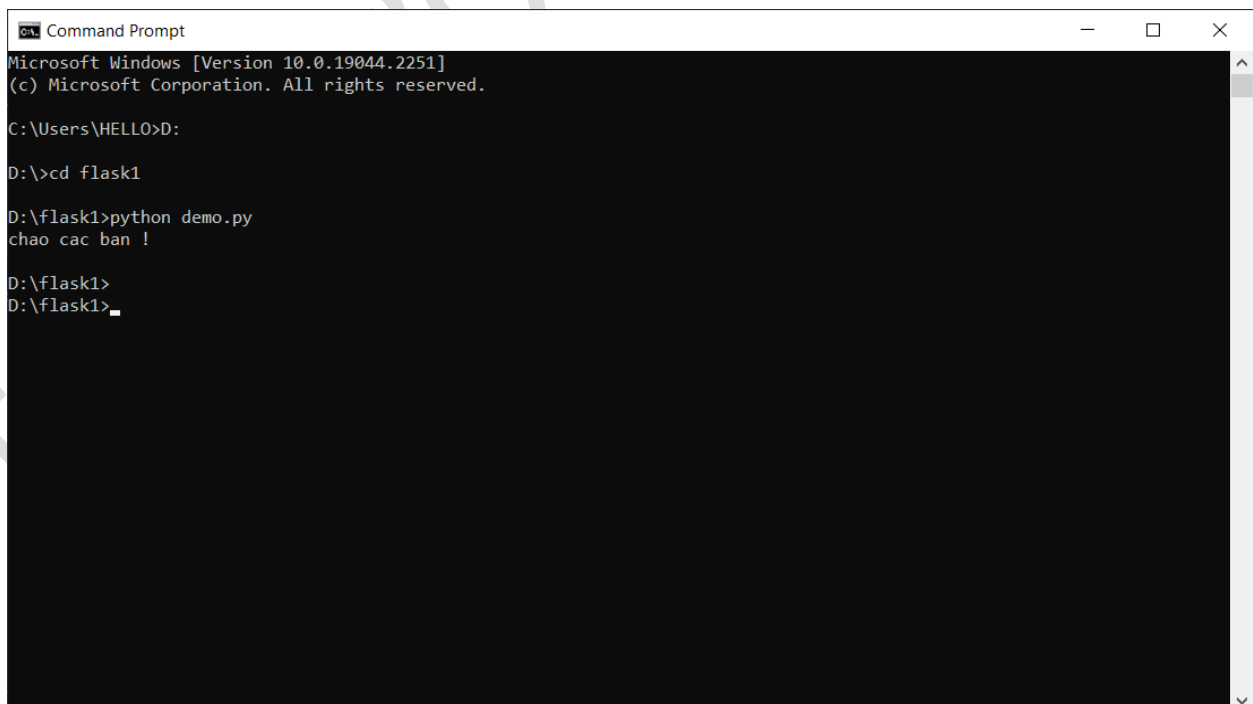
## Giáo viên: Nguyễn Hùng Cường

### Bước 1: Tạo ứng dụng Flask

Mở một trình soạn thảo, sau đó tạo mới một ứng dụng Python thông thường. Tiếp theo, ta tạo một file mã nguồn Python để kiểm tra. Ở đây ta đã tạo ứng dụng với tên là flask1, và tạo một file Python với tên là Demo.py



Ta có thể mở console ra để thực thi thử file Python này, ta thấy file đã được thực thi thành công.



## Bước 2: Viết mã cho ứng dụng Flask đầu tiên

Tiếp theo, ta mở terminate rồi dùng pip để cài đặt Flask framework cho ứng dụng như sau.

```
Terminal: Local x + v
PS D:\flask1> pip install flask
Collecting flask
  Downloading Flask-2.2.2-py3-none-any.whl (101 kB)
    |████████████████████████████████████████████████████████████████████████████████| 101 kB 2.2 MB/s
Collecting Werkzeug>=2.2.2
  Downloading Werkzeug-2.2.2-py3-none-any.whl (232 kB)
    |████████████████████████████████████████████████████████████████████████████████| 232 kB 6.4 MB/s
Collecting itsdangerous>=2.0
  Downloading itsdangerous-2.1.2-py3-none-any.whl (15 kB)
Collecting Jinja2>=3.0
  Downloading Jinja2-3.1.2-py3-none-any.whl (133 kB)
    |████████████████████████████████████████████████████████████████████████████████| 133 kB 2.2 MB/s
Collecting click>=8.0
  Using cached click-8.1.3-py3-none-any.whl (96 kB)
Collecting colorama
```

Tiếp theo, ta tạo một file mã nguồn Python với tên là `First.py`, rồi viết mã để tạo một chương trình Flask đơn giản đầu tiên như sau (chú ý cần import Flask)

Ý nghĩa của đoạn code đầu tiên như sau:

- Đầu tiên, ta import class Flask, sau đó khởi tạo đối tượng Flask với tên là app.
- Sau đó, ta định nghĩa function hello\_world(), ở đây ta đã khai báo một route gắn với function này, ứng với url / thì chương trình sẽ gọi function này.
- Cuối cùng, ta thực thi ứng dụng với method run()

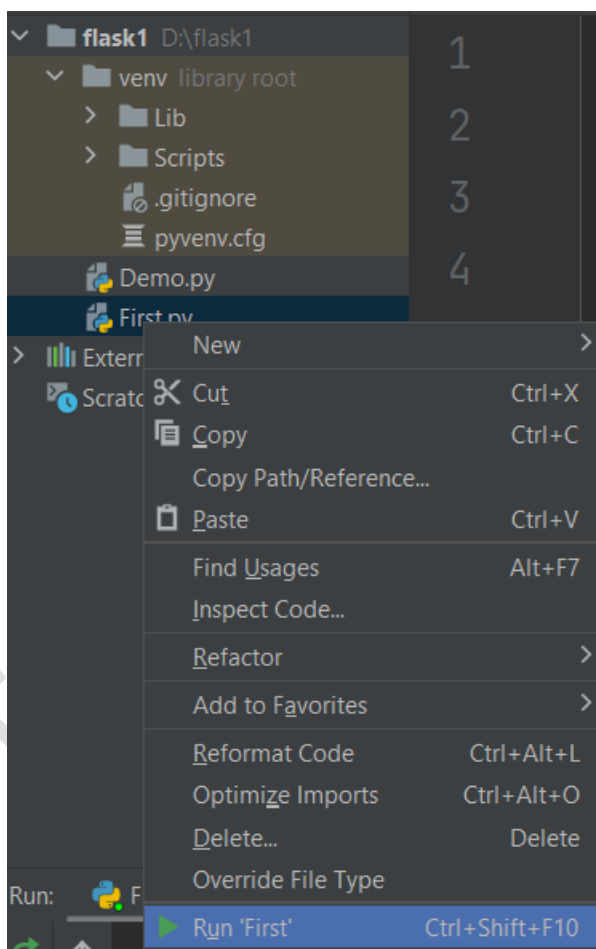
```
First.py x
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return "Lập trình Flask Thầy Cường - Hello World !"

if __name__ == '__main__':
    app.run()
```

### Bước 3: Thực thi ứng dụng và xem kết quả

Sau khi đã viết mã xong, hãy thực thi file First.py vừa tạo ở bước trên và xem kết quả.



Ta sẽ thấy trong cửa sổ Terminate hiển thị thông báo như hình bên dưới.

```
First
D:\flask1\venv\Scripts\python.exe D:/flask1/First.py
* Serving Flask app 'First'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [07/Dec/2022 23:59:16] "GET / HTTP/1.1" 200 -
```

Để mở ứng dụng trên web, ta gõ địa chỉ <http://127.0.0.1:5000> thì sẽ thấy ứng dụng thực thi function `hello_world()` như hình bên dưới đúng như mong muốn.

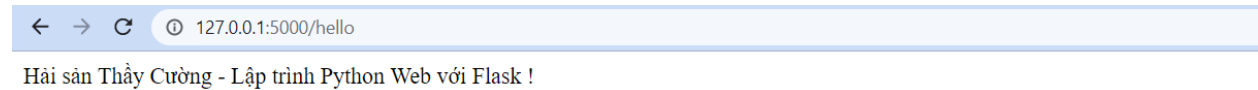


Ta có thể sửa lại nội dung của function `hello_world()` như sau.

```
Pro... Demo.py First.py
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route('/hello')
5 def hello_world():
6     return "Hải sản Thầy Cường - Lập trình Python Web với Flask !"
7
8 if __name__ == '__main__':
9     app.run()
10
11

First
D:\flask1\venv\Scripts\python.exe D:/flask1/First.py
* Serving Flask app 'First'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
127.0.0.1 - - [08/Dec/2022 00:10:18] "GET / HTTP/1.1" 404 -
```

Thực thi lại ứng dụng, rồi mở lại trình duyệt, ta thấy nội dung mới đã được hiển thị lại đúng như mong muốn.



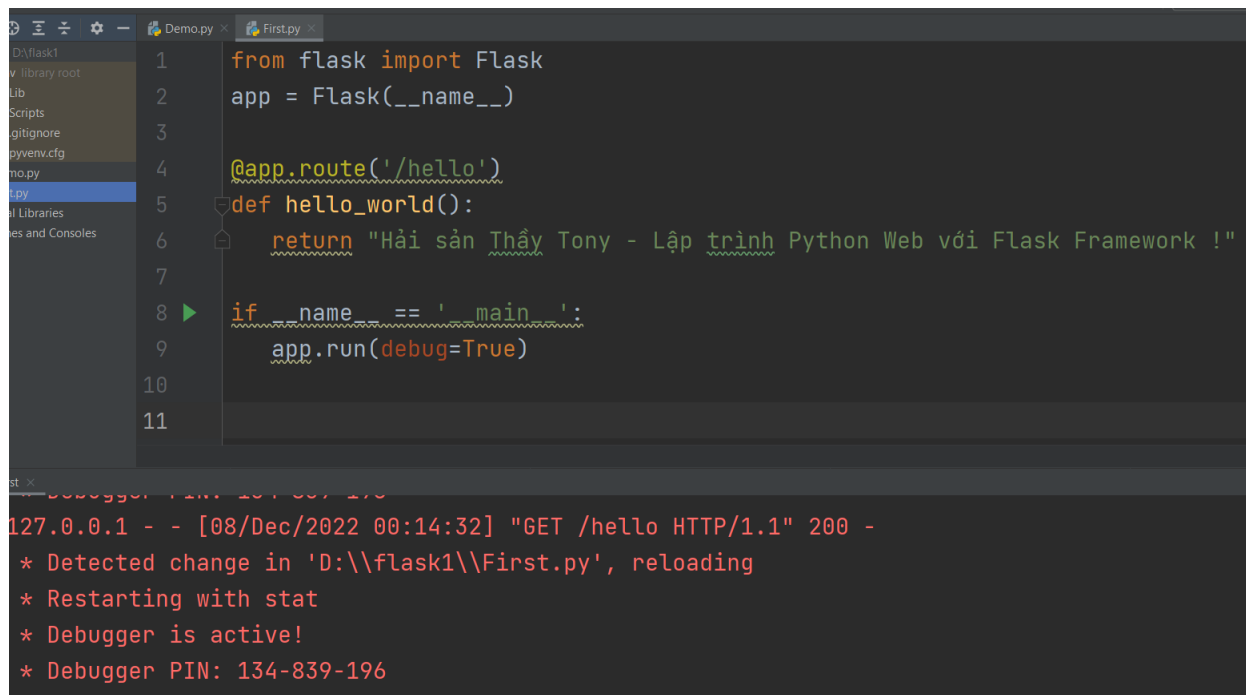
Như ở phần trên, mỗi lần thay đổi mã nguồn, ta phải thực thi lại ứng dụng thì mới thấy nội dung mới được cập nhật. Để tránh việc phải bật đi bật lại nhiều lần ứng dụng, ta có thể truyền đối số `debug=True` vào method `run()`, để server sẽ tự refresh lại mỗi khi ta có thay đổi mã nguồn như sau.

```
from flask import Flask
app = Flask(__name__)

@app.route('/hello')
def hello_world():
    return "Hải sản Thầy Cường - Lập trình Python Web với Flask Framework !"

if __name__ == '__main__':
    app.run(debug=True)
```

Sau đó, ta cập nhật lại mã nguồn của function `hello_world()` như sau. Ta thấy trong màn hình Terminal đã hiển thị thông báo như hình bên dưới, đã bật chế độ debug.



The screenshot shows a code editor with a file named `First.py`. The code defines a Flask application with a single route `/hello` that returns a Vietnamese greeting. The terminal output shows the application running on `127.0.0.1:5000`, detecting a file change, and reloading. The debugger is active with a PIN of `134-839-196`.

```
1 from flask import Flask
2 app = Flask(__name__)
3
4 @app.route('/hello')
5 def hello_world():
6     return "Hải sản Thầy Tony - Lập trình Python Web với Flask Framework !"
7
8 if __name__ == '__main__':
9     app.run(debug=True)
```

```
Debugger PIN: 134-839-196
127.0.0.1 - - [08/Dec/2022 00:14:32] "GET /hello HTTP/1.1" 200 -
* Detected change in 'D:\\flask1\\First.py', reloading
* Restarting with stat
* Debugger is active!
* Debugger PIN: 134-839-196
```

Refresh lại cửa sổ trình duyệt, ta thấy nội dung mới đã được hiển thị đúng như mong muốn.

