Flask Tutorial

I will show you how to create websites with python using the micro framework flask. Flask is designed for quick development of simple web applications.

What is Flask?

Flask is a known as a **micro web framework**. This means it provides some basic functionality to allow developers to build simple websites. It does not come with all the bells and whistles like some other web frameworks like django have and therefore is typically not used for complex websites. However, there is a benefit to flask's limited features. One of which is it's simplicity. You will see in these tutorial that flask makes it very easy and quick to do things. In fact in under 10 minutes we will have our first website up and running!

Installing Flask

Installing flask is pretty simple. All that is required is to use the pip command: **pip install flask.**

You can run this command from your command prompt window.

Starting a Project

Installing flask is pretty simple. All that is required is to use the pip command: **pip install flask.**

You can run this command from your command prompt window.

Starting a Project

Creating a website in flask is as easy as **creating a new python script, importing flask and starting the instance.**

I've named my python file ***tutorial1.py*** and put it in it's own folder. You can name yours whatever you'd like.

tutorial1.py

```
from flask import Flask

app = Flask(__name__)

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

And now we've created our first flask project!

Creating Web Pages

Time to make our first web page. Flask makes this pretty easy as all we need to do is define a function that will represent each page and set its URL/Path. These functions will perform some operations and return some HTML that should be rendered. For this first video we will just return some very basic HTML that's written as a string inside our python script.

We do this by decorating the function (see below)

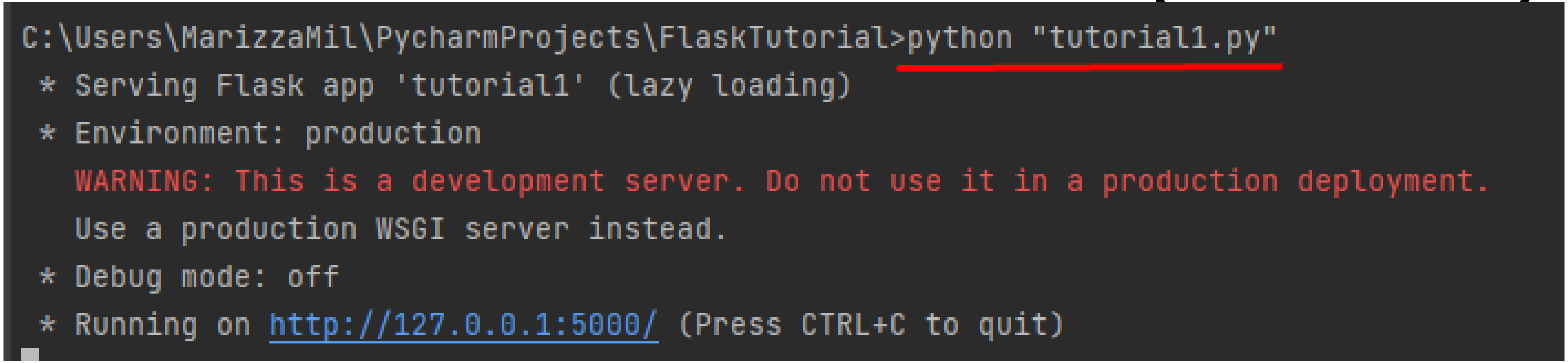
tutorial1.py

```
from flask import Flask

app = Flask(__name__)

# Defining the home page of our site
@app.route("/")  # this sets the route to this page
def home():
    return "Hello! this is the main page <h1>HELLO</h1>"  # some basic inline html

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

We've now successfully created our first web page! Time to test it out.

Running Your Website

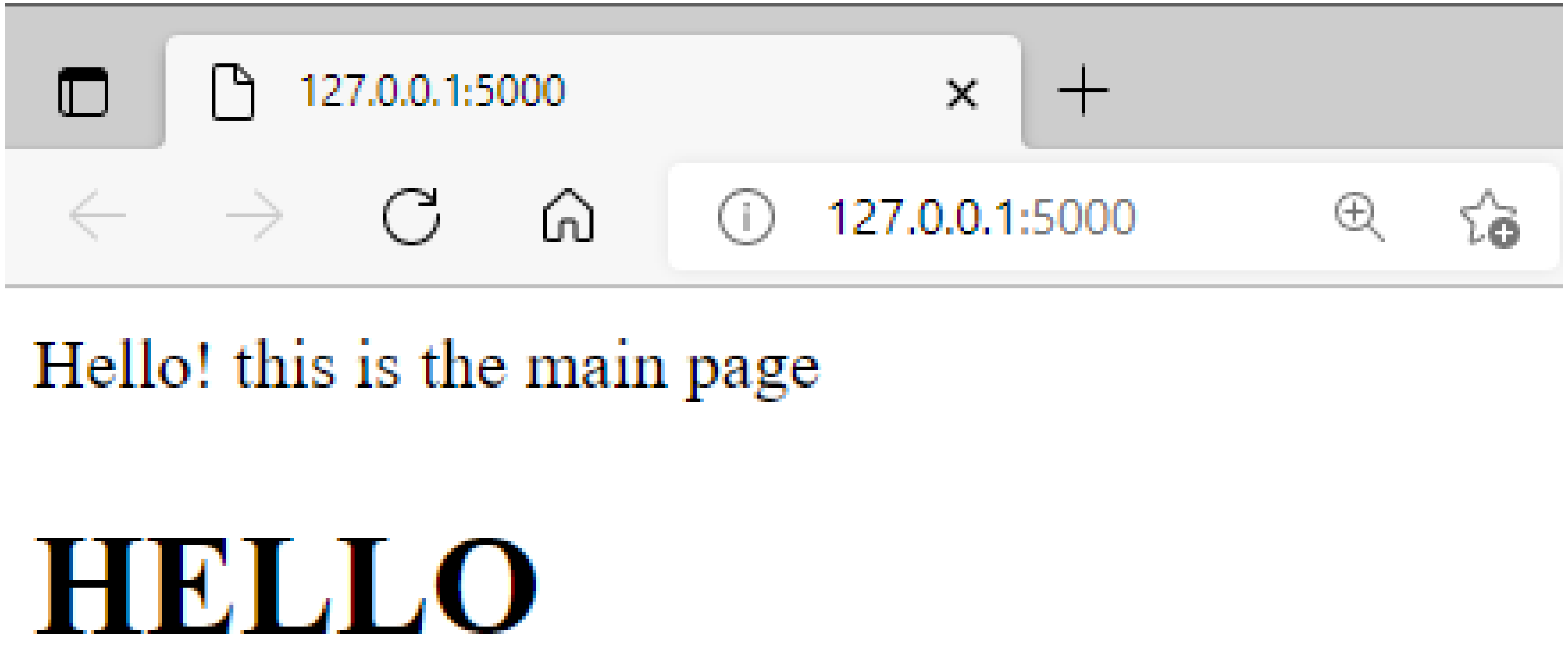
Time to make our first web page. Flask makes this pretty easy as all we need to do is define a function that will represent each page and set its URL/Path. These functions will perform some operations and return some HTML that should be rendered. For this first video we will just return some very basic HTML that's written as a string inside our python script.
We do this by decorating the function (see below)



```
C:\Users\MarizzaMil\PycharmProjects\FlaskTutorial>python "tutorial1.py"
 * Serving Flask app 'tutorial1' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

You may notice some warnings when starting your script. Just ignore these for now.

Running Your Website



Hello! this is the main page

HELLO

Once you've connected to your site you should see something like the image above.

Dynamic URLs

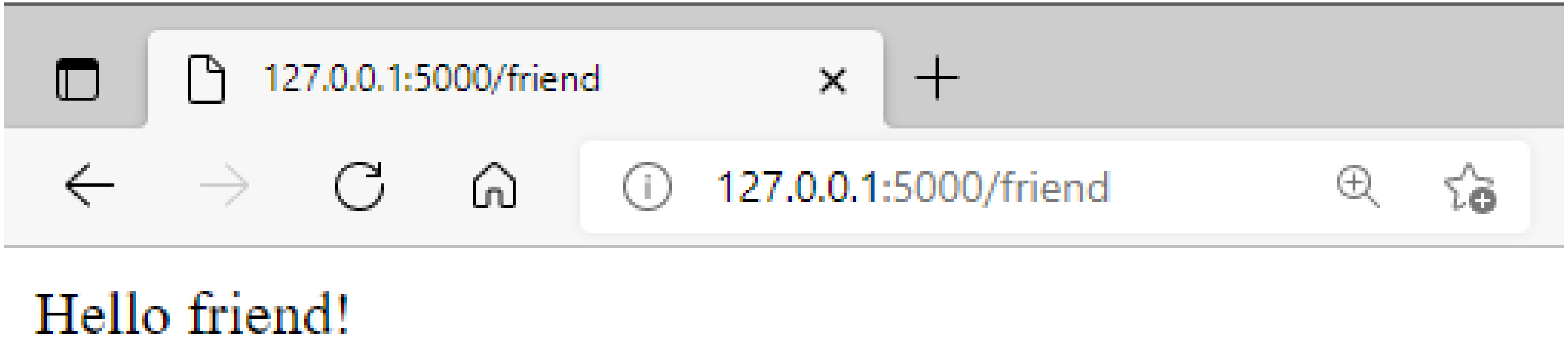
Now let's move to our next web page! This time however we are going to access it using something called a dynamic url/path. This essentially allows us to route any url that matches a specific pattern to a specific web page. This can be useful for things like posts. Where each post may be displayed the exact same way and merely differ in content. By creating a dynamic path to all of our posts we can read the post Id from the address bar and pass that to a function that can display that post. Rather than creating a web page/function for each of our posts.
To define a dynamic path you simply put a variable name inside of . Like this: .

tutorial1.py

```
from flask import Flask

app = Flask(__name__)

@app.route("/")
def home():
    return "Hello! this is the main page <h1>HELLO</h1>"

@app.route("/<name>")
def user(name):
    return f"Hello {name}!"

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

Dynamic URLs

Have a look at the example above. This will allows us to route any URL to this function, pass the name variable to our function and display "Hello name!" on our web page. Let's give it a try.



127.0.0.1:5000/friend

127.0.0.1:5000/friend

Hello friend!

Redirects

The last thing to mention is how to redirect users to other pages from within our python code. To do this we will start by importing some more functions from Flask

I'll create another web page called admin to illustrate how to use these functions.

The **url_for** function takes an argument that is the name of the function that you want to redirect to. In this example since we want to redirect to the home page we will put the name of the function for the home page which is "name".

Now whenever we visit /admin we will be redirected home.

tutorial1.py

```
from flask import Flask, redirect, url_for

app = Flask(__name__)

@app.route("/")
def home():
    return "Hello! this is the main page <h1>HELLO</h1>"

@app.route("/<name>")
def user(name):
    return f"Hello {name}!"

@app.route("/admin")
def admin():
    return redirect(url_for("home"))

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