Lisbon School of Economics & Management
Universidade de Lisboa

U LISBOA
UNIVERSIDADE DE LISBOA

Flask

web development, one drop at a time

Table of Contents

Flask

- Flask is a microframework for Python
- Lightweight WSGI (Web Server Gateway Interface) web application framework
- Began as a simple wrapper around Werkzeug and Jinja
- Created by Armin Ronacher
- BSD licensed
- https://palletsprojects.com/p/flask/

Werkzeug

- Utility library for Python
- Toolkit for Web Server Gateway Interface (WSGI) applications
- Licensed under a BSD License.
- Can realize software objects for request, response, and utility functions.
- Python 2.7 and 3.5 and late
- https://palletsprojects.com/p/werkzeug/

Jinja

- Template engine for the Python
- Licensed under a BSD License.
- Similar to the Django web framework
- Handles templates in a sandbox.
- https://palletsprojects.com/p/jinja/
- https://jinja.palletsprojects.com/en/3.1.x/
- https://github.com/pallets/jinja

Flask Application (ex01)

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello, World!’
if __name__ == "__main__":
    app.run()
```

Flask Application (ex01)

```
from flask import Flask
app = Flask(__name__)
@app.route('/’)
def index():
        return 'Hello from Flask1!’

app.run(host='0.0.0.0', port=81)
```

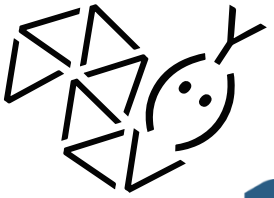
Flask Application

- Save in a folder flask_app.py
- Call:

  python flask_app.py
- In the browser:

  localhost:5000

Deployment



pythonanywhere

bitnami

replit

RED HAT® OPENSHIFT

HEROKU

WikiWikiWeb.de

Routing

- Use the route() decorator to bind a function to a URL.

```
@app.route('/')
def index():
    return 'Index Page'

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

Routing (ex02)

```
from flask import Flask
app = Flask(__name__)
@app.route('/')
def hello_world():
    return 'Hello, World (in index paage)'
@app.route('/hello')
def hello():
    return 'Hello, World (in hello page)'
if __name__ == "__main__":
    app.run()
```

Static Files

- Dynamic web applications also need static files.
- E.g.CSS and JavaScript
- Create a folder called in the package or next to the module

  /static

  url_for('static', filename='style.css')

Rendering Templates

- Flask configures the Jinja2 template engine automatically
- To render a template use the render_template() method
- https://jinja.palletsprojects.com/

Rendering Templates

- **Block**
  {% **block** %}
- **Variable**
  {{ var }}
- **Comment**
  {# comment #}

Example:

/flask_app.py
/templates
/hello.html

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def hello():
    nome="ISEG"
    return render_template('hello.html', var=nome)
if __name__ == "__main__":
    app.run()
```

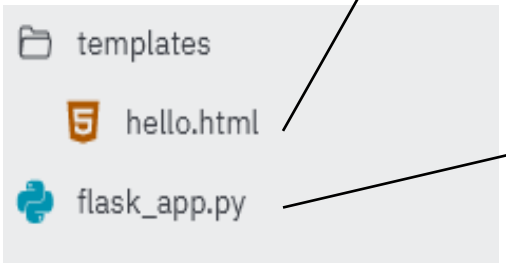
```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
</ul>
   <h1>My Webpage</h1>
   {{ var}}
</body>
</html>
```

Rendering Templates

- **Block**
  {% **block** %}
- **Variable**
  {{ var }}
- **Comment**
  {# comment #}

```
<!DOCTYPE html>
<html lang="en">
<head>
</head>
<body>
</ul>
   <h1>My Webpage</h1>
   {{ var}}
</body>
</html>
```

Example:



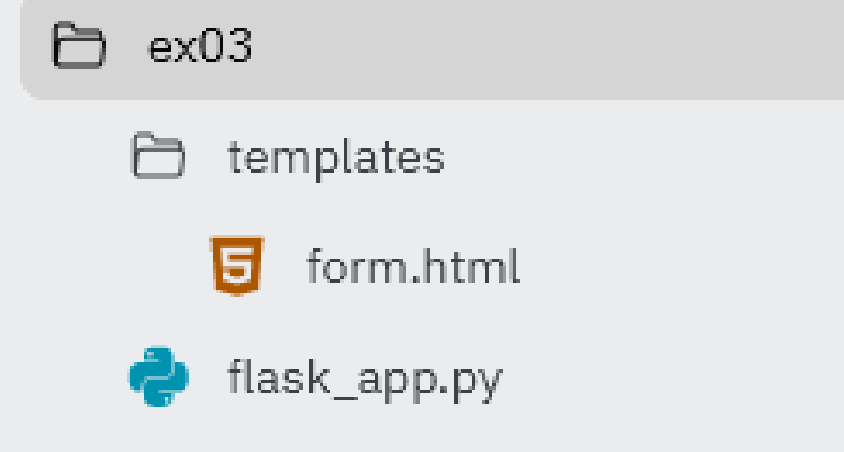

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/hello/<name>')
def hello(name):

    return render_template('hello.html', var=name)
if __name__ == "__main__":
    app.run()
```

Rendering Templates

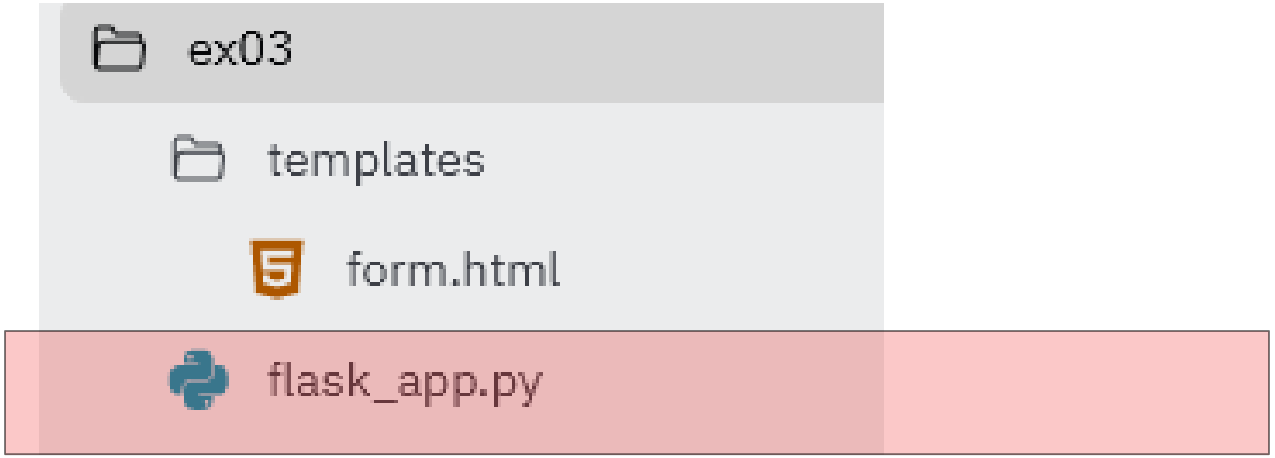

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>My Webpage</title>
</head>
<body>
   <ul id="navigation">
      <li><a href="http://site1.pt">site 1</a></li>
      <li><a href="http://site2.pt">site 2</a></li>
   </ul>
   <h1>My Webpage</h1>
   ISEG
</body>
</html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>My Webpage</title>
</head>
<body>
   <ul id="navigation">
   {% for item in navigation %}
      <li><a href=“ {{ item.href }} "> {{ item.caption }} </a></li>
   {% endfor %}
   </ul>
   <h1>My Webpage</h1>
   {{ a_variable }}
   {# a comment #}
</body>
</html>
```

Rendering Templates (ex03)



Rendering Templates (ex03)

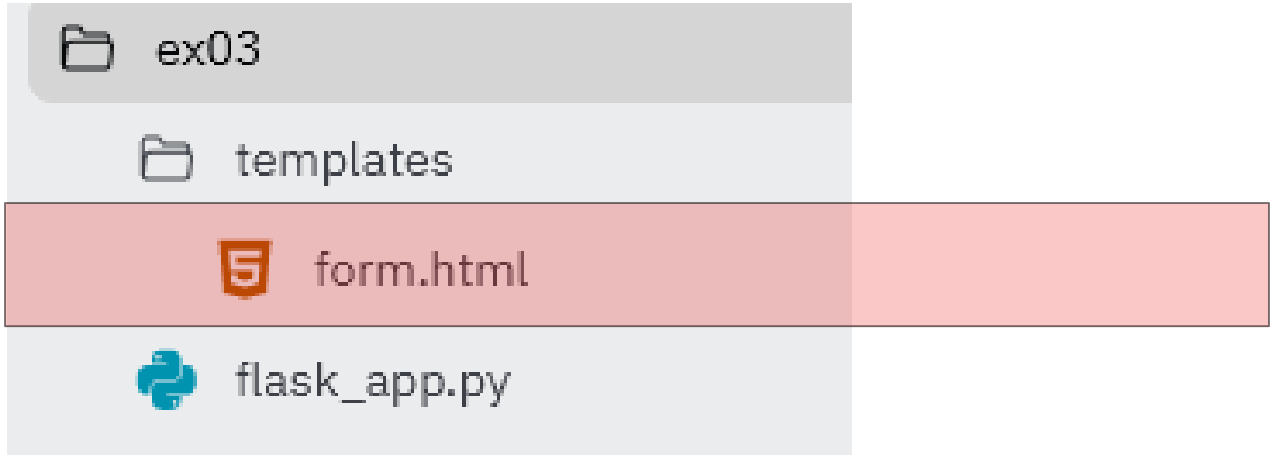


Rendering Templates (ex03)

flask_app.py File

```
from flask import Flask, render_template
app = Flask(__name__)
@app.route('/')
def render_static():
    return render_template('form.html')
@app.route('/hello')
def hello():
    return 'Hello, World (in hello page)'
if __name__ == "__main__":
    app.run()
```

Rendering Templates (ex03)

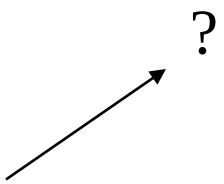


Rendering Templates (ex03)

form.html File

```
<html>
   <body>
      <form action = "/save/" method = "POST">
         <p>author <input type = "text" name = "Author" /></p>
         <p>phrase <input type = "text" name = "Phrase" /></p>
         <p><input type = "submit" value = "submit" /></p>
      </form>
   </body>
</html>
```

Rendering Templates (ex03)



form.html
File

```
<html>
   <body>
      <form action = "/save/" method = "POST">
         <p>author <input type = "text" name = "Author" /></p>
         <p>phrase <input type = "text" name = "Phrase" /></p>
         <p><input type = "submit" value = "submit" /></p>
      </form>
   </body>
</html>
```

HTTP Methods

- Web applications use different HTTP methods when accessing URLs.
- By default, a route only answers to GET requests.
- use the methods argument of the route() decorator to handle different HTTP methods.

HTTP Methods

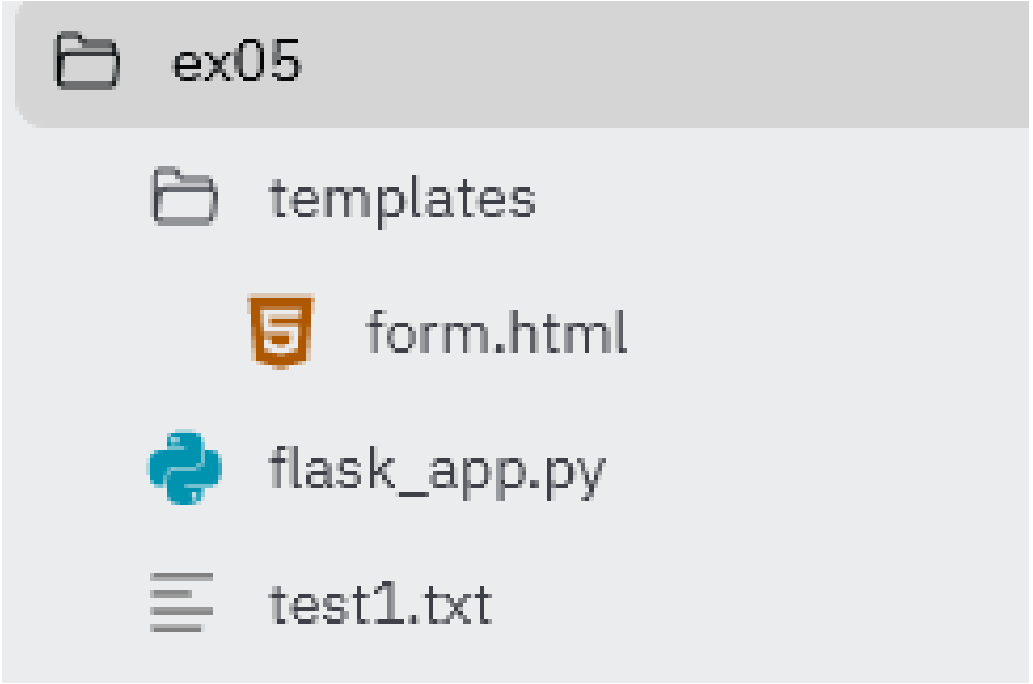
```
@app.route('/save/', methods=['GET', 'POST'])

def getFromForm():
    author = request.form['Author']+"
    text1=  request.form['Phrase']
    return "thank you"
```

Rendering Templates (ex04)

```
from flask import Flask, render_template, request
app = Flask(__name__)
@app.route('/')
def render_static():
    return render_template('form.html')
@app.route('/save/', methods=['GET', 'POST'])
def hello():
    return 'Hello!!!'
if __name__ == "__main__":
    app.run()
```

New flask_app.py File

Save Data (ex05)

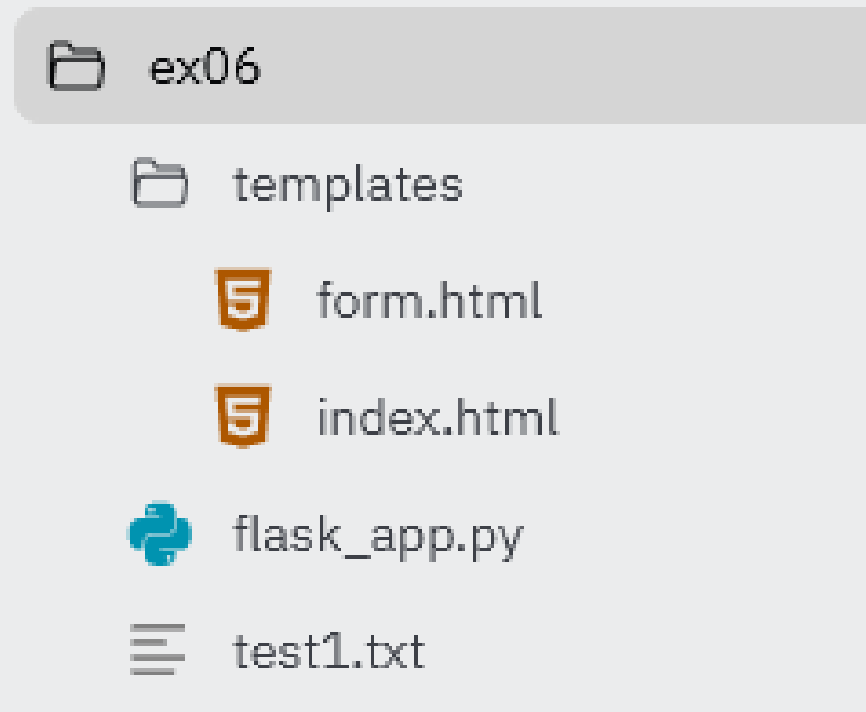


Save Data (ex05)

```
from flask import Flask, render_template, request
app = Flask(__name__)
@app.route('/')
def render_static():
        return render_template('form.html')
@app.route('/save/', methods=['GET', 'POST'])
def index():
        data = request.form['Author']+" - "+request.form['Phrase']
        fo= open("test1.txt", "a+")
        fo.write(data+"\n")
        fo.close()
        return "thank you"
if __name__ == '__main__’:
            app.run(debug = True)
```

flask_app.py File

Tiny App (ex06)



Tiny App (ex06)

flask_app.py File

```
from flask import Flask, request, render_template
app = Flask(__name__)
@app.route('/save/', methods=['GET', 'POST'])
def write():
        data = request.form['Author']+" - "+request.form['Phrase']
        fo= open("test1.txt", "a+")
        fo.write(data+"\n")
        fo.close()
        return render_template('index.html')
@app.route('/read/')
def read():
        fo= open("test1.txt", "r")
        data1=fo.read()
        fo.close()
        return data1
@app.route('/')
def render_static():
    return render_template('form.html')
if __name__ == '__main__':
    app.run(debug = True)
```

Tiny App (ex06)

form.html
File

```
<html>
  <body>
    <form action = "/save/" method = "POST">
      <p>author <input type = "text" name = "Author" /></p>
      <p>phrase <input type = "text" name = "Phrase" /></p>
      <p><input type = "submit" value = "submit" /></p>
    </form>
  </body>
</html>
```

Tiny App (ex06)

index.html File

```
<html>
    <body>
        menu
        <p><a href="/read/">list data</a></p>
        <a href="../">form</a>
    </body>
</html>
```

- https://github.com/masterfloss/python-web
- https://flask.palletsprojects.com/
- https://www.geeksforgeeks.org/python-introduction-to-web-development-using-flask/