

Flask Templates

Author: Denis Rinfret

Template Inheritance

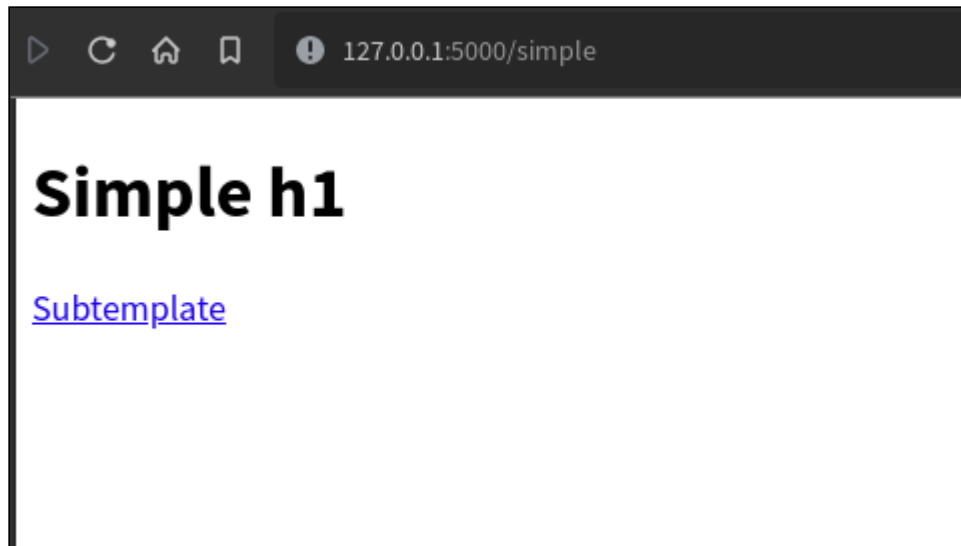
- Jinja2 supports *template inheritance*
 - create a base template that contains the common blocks of your website, such as
 - the *HTML DOCTYPE* and the `<head>` element
 - the web page banner
 - the navigation bar
 - the footer
 - etc...
 - create one template for each page that extends the base template
 - add the page specific contents to the sub-templates
- To facilitate adding page specific contents to each sub-template, define *blocks* in the base template
- Sub-templates can redefine these blocks
 - use `{% block head %}` to define the start of a block named *head* (replace *head* by whatever is appropriate)
 - use `{% endblock %}` to mark the end of a block

Simple Template Inheritance Example

- Parent template
 - project 03_flask_templates , file simple.html

```
<!DOCTYPE html>
<html lang="en">
<head>{% block head %}
    <meta charset="UTF-8">
    <title>{% block title %}Simple Template Inheritance Example{% endblock
    %}</title>
{% endblock %}
</head>
<body>
```

```
{% block contents %}
<h1>Simple h1</h1>
{% endblock %}
{% block foot %}
<a href="sub">Subtemplate</a>
{% endblock %}
</body>
</html>
```



Simple

- Child template
 - file sub.html

```
{% extends "simple.html" %}

{% block title %}Subtemplate{% endblock %}

{% block contents %}
{{ super() }}
<h3>Subtemplate h3</h3>
{% endblock %}

{% block foot %}
<a href="simple">Simple</a>
{% endblock %}
```

- A new title is defined in the `title` block
- An `h3` is added to the `contents` block
 - `super()` is called to keep the parent template's contents
- The `foot` block is replaced by a new link

- The `head` block is not modified directly (only the `title` sub-block is modified)

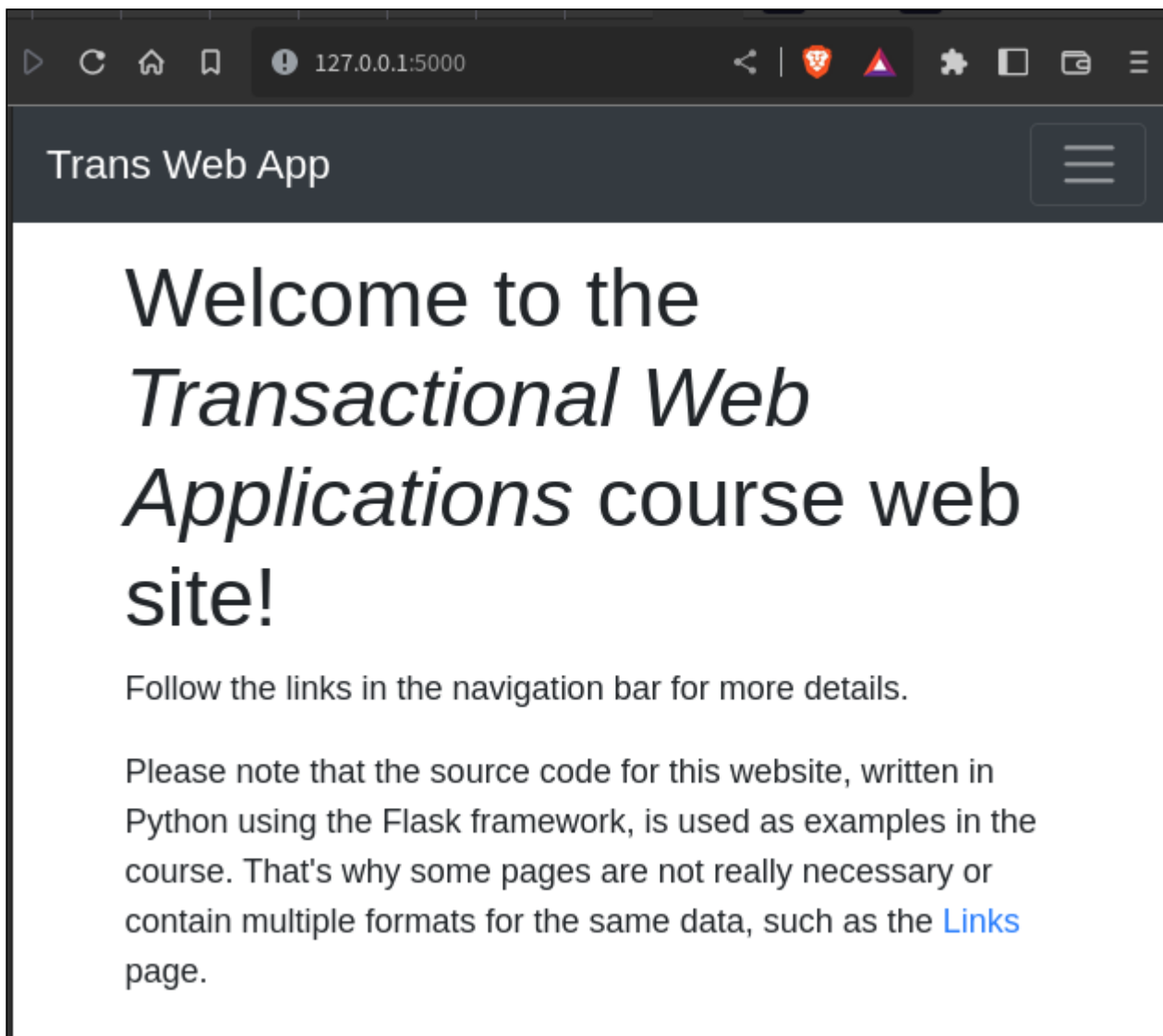
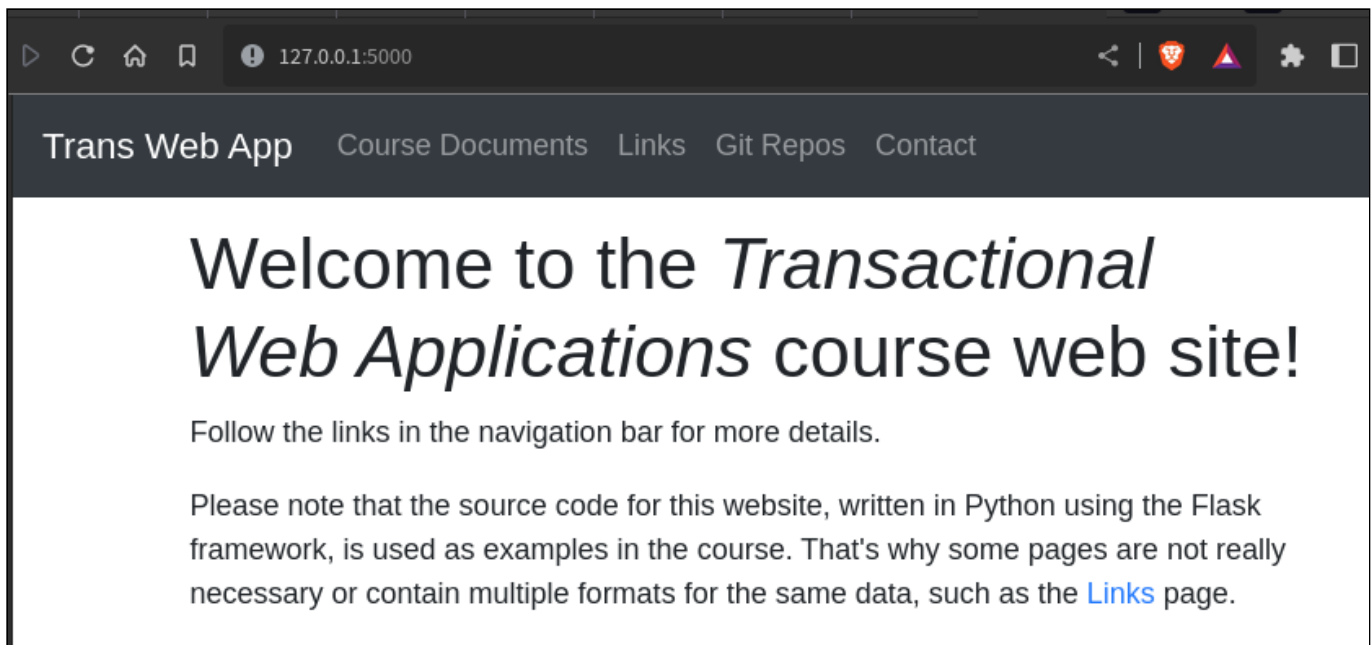


Sub-template

Bootstrap Template Example

File `base.html`

- This template uses Bootstrap 4
- Almost the same as a previous example (`bs4_base.html`), but with Jinja2 blocks added



```
<!DOCTYPE html>  
<html lang="en">
```

```

<head>
  {% block head %}
    <title>{% block title %}Trans Web App{% endblock %}</title>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet"
      href="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/css/bootstrap.min
        integrity="sha384-
x00lHFLEh07PJGoPkLv1IbcEPTNtaed2xpHsD9ESMhqIYd0nLMwNLD69Npy4HI+N"
        crossorigin="anonymous">
    <script
      src="https://cdn.jsdelivr.net/npm/jquery@3.5.1/dist/jquery.slim.min.js"
        integrity="sha384-
DfXdz2htPH0lsSSs5nCTpuj/zy4C+OGpamoFVy38MVBnE+IbbVYUew+OrCXAkRkfj"
        crossorigin="anonymous"></script>
    <script
      src="https://cdn.jsdelivr.net/npm/bootstrap@4.6.2/dist/js/bootstrap.bundl
        integrity="sha384-
Fy6S3B9q64WdZWQlU+q4/2Lc9npb8tCaSX9FK7E8HnRr0Jz8D60P9d05Vg3Q9ct"
        crossorigin="anonymous"></script>

    <style>
      body {
        padding-top: 50px
      }
    </style>
  {% endblock %}
</head>
<body>

  {% block navbar %}
    <nav class="navbar navbar-expand-md bg-dark navbar-dark fixed-top">
      <a class="navbar-brand" href="/">Trans Web App</a>
      <button class="navbar-toggler" type="button" data-toggle="collapse"
        data-target="#collapsibleNavbar">
        <span class="navbar-toggler-icon"></span>
      </button>
      <div class="collapse navbar-collapse" id="collapsibleNavbar">
        <ul class="navbar-nav">
          <li class="nav-item">
            <a class="nav-link" href="/documents">Course Documents</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/links">Links</a>
          </li>
          <li class="nav-item">
            <a class="nav-link" href="/git">Git Repos</a>
          </li>
          <li class="nav-item">

```

```

        <a class="nav-link" href="/contact">Contact</a>
    </li>
</ul>
</div>
</nav>
{% endblock %}
<br>

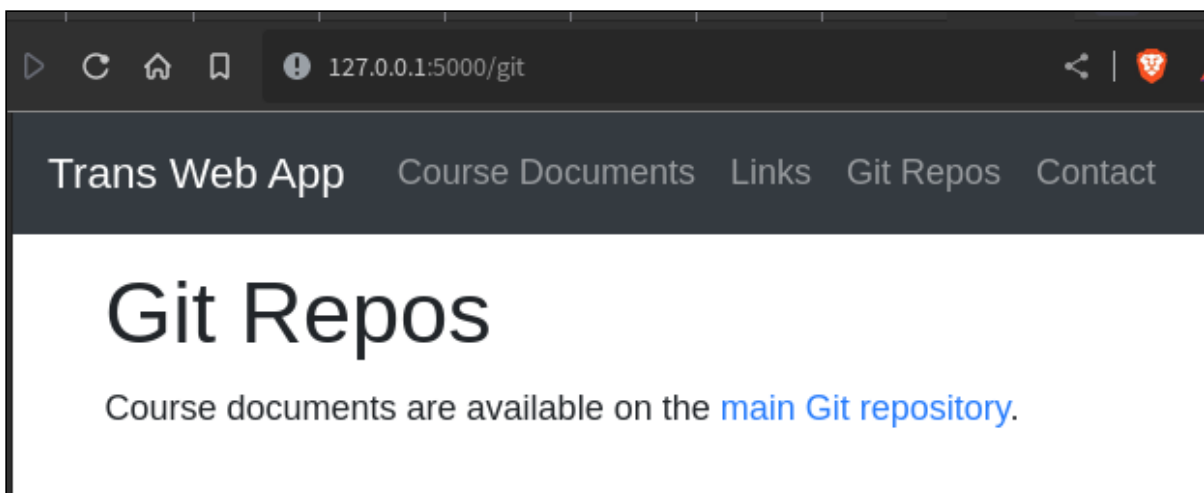
{% block content %}
    <div class="container">
        <h1>Welcome to the <em>Transactional Web Applications</em> course web
        site!</h1>
        <p>Follow the links in the navigation bar for more details.</p>
        <p>Please note that the source code for this website, written in Python
        using the Flask framework, is used
            as examples in the course. That's why some pages are not really
            necessary or contain multiple formats for
            the same data, such as the <a href="/links">Links</a> page.</p>
    </div>
{% endblock %}

</body>
</html>

```

File git.html

- Template rendered by the `/git` endpoint
 - extends the base template
 - calls `super()` in the `title` block and adds a string to it
 - the content block is completely redefined here
 - the other blocks are left as-is



assignments

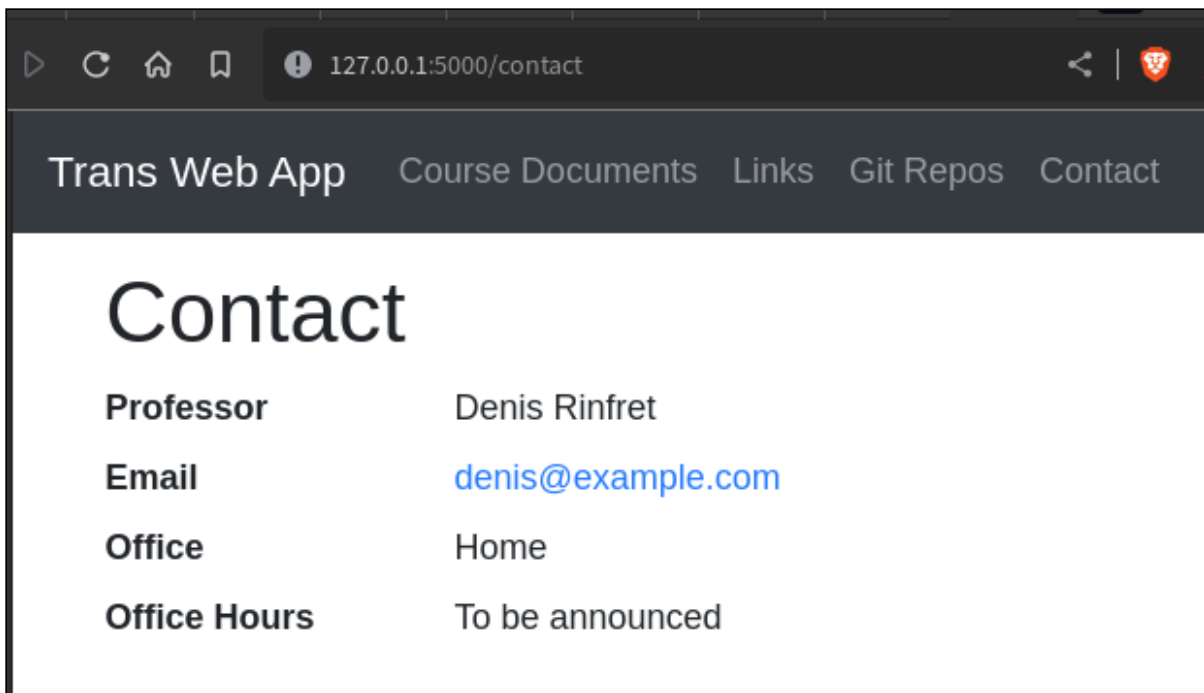
```
{% extends "base.html" %}

{% block title %}{{ super() }} - Git Repos{% endblock %}

{% block content %}
    <div class="container">
        <h1>Git Repos</h1>
        <p>Course documents are available on the
            <a href="https://github.com/profdenis/trans-web-app"
              target="_blank">main Git repository</a>.</p>
        </div>
{% endblock %} <!-- content -->
```

File contact.html

- Template rendered by the /contact endpoint
 - extends the base template
 - it simply lists contact information in a <dl>



Contact

```
{% extends "base.html" %}

{% block title %}{{ super() }} - Contact{% endblock %}

{% block content %}
    <div class="container">
        <h1>Contact</h1>
        <dl class="row">
```

```

<dt class="col-sm-3">Professor</dt>
<dd class="col-sm-9">Denis Rinfret</dd>

<dt class="col-sm-3">Email</dt>
<dd class="col-sm-9"><a
href="mailto:denis@example.com">denis@example.com</a> </dd>

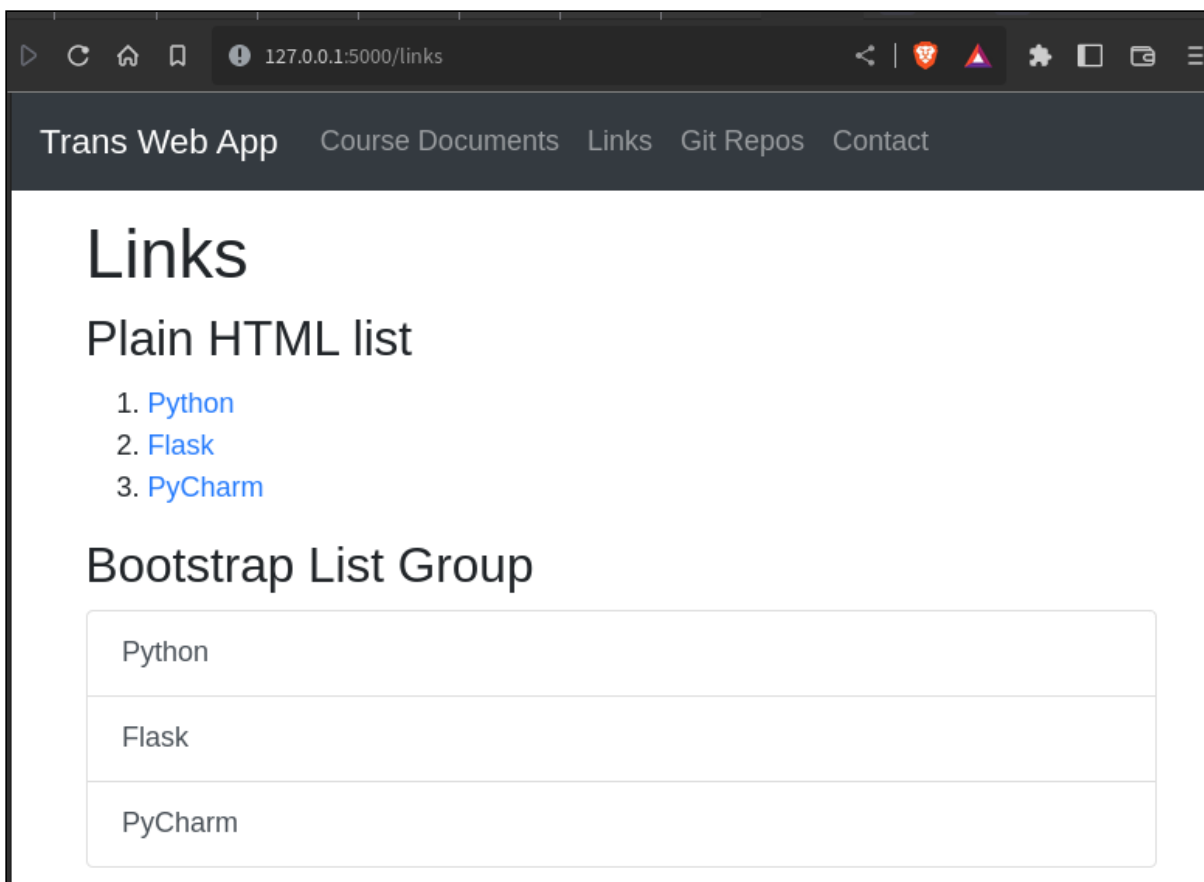
<dt class="col-sm-3">Office</dt>
<dd class="col-sm-9">Home</dd>

<dt class="col-sm-3">Office Hours</dt>
<dd class="col-sm-9">To be announced</dd>
</dl>
</div>
{% endblock %} <!-- content -->

```

File links.html

- Template rendered by the `/links` endpoint
 - extends the base template
 - this example shows 2 ways to list a number of files using a `for` loop
 - normally, only 1 of the 2 would be used in a page




```

{% extends "base.html" %}

{% block title %}{% super() %} - Links{% endblock %}

{% block content %}
    <div class="container">
        <h1>Links</h1>
        <h3>Plain HTML list</h3>
        <ol>
            {% for link in links_list %}
                <li><a href="{{ link[1] }}">{{ link[0] }}</a></li>
            {% endfor %}

        </ol>
        <h3>Bootstrap List Group</h3>
        <div class="list-group">
            {% for link in links_list %}
                <a href="{{ link[1] }}"
                    class="list-group-item list-group-item-action">{{ link[0] }}
            </a>
            {% endfor %}
        </div>
    </div>
{% endblock %} <!-- content -->

```

Generating a Plain HTML List

- `links_list` is a Python list containing tuples of the form *(text, href)*, where
 - *text* is the link text to be displayed, and
 - *href* is the URL
- So we loop over all the tuples in the list with a `for` loop
- And we build an `<a>` element using
 - `{{ link[1] }}` to get the value of *href*, and
 - `{{ link[0] }}` to get the text

Generating a Bootstrap List Group

- We use the same `for` loop as before, except that the loop is located inside a `<div class="list-group">`
- The `<a>` element is generated in the same way, except that we need to add a couple Bootstrap classes to display each link correctly
 - `class="list-group-item list-group-item-action">`

The /links Endpoint

- For now, we build a list of tuples for each slide manually in the endpoint
- In the `documents` endpoint, the list will be built with data read from a CSV file
- How to read data from a database will be covered later on

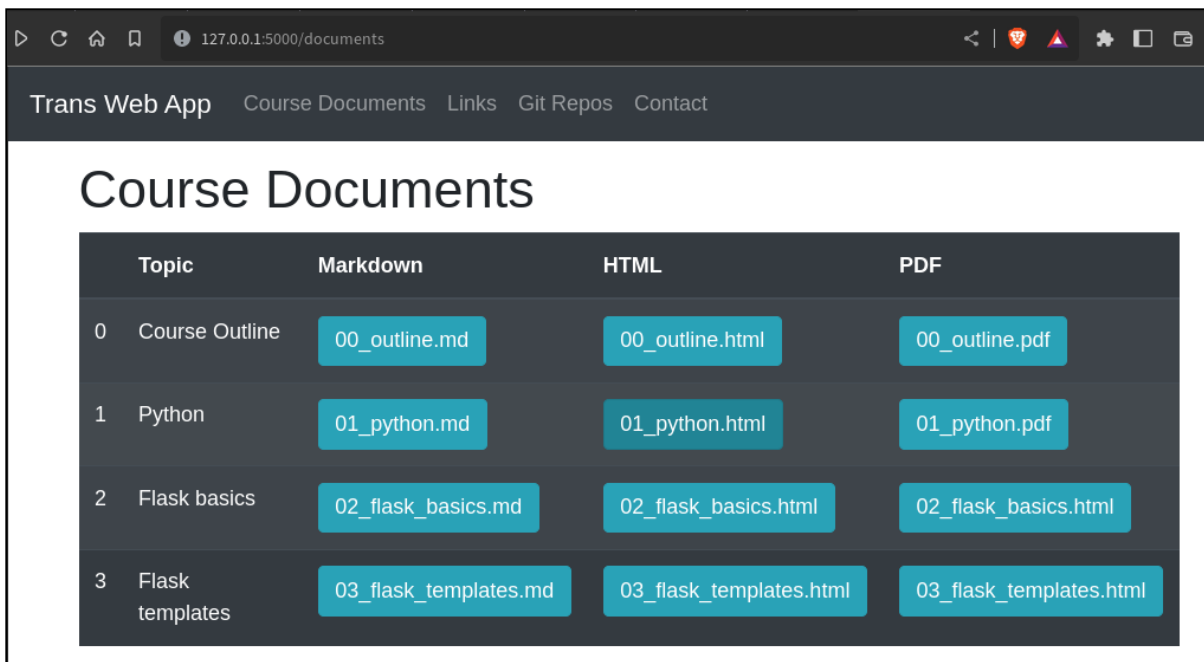
```
@app.route('/links')
def slides():
    prefix = '/static/'
    links_list = [('Python', 'https://www.python.org'),
                  ('Flask', 'https://flask.palletsprojects.com/en/2.2.x/'),
                  ('PyCharm', 'https://www.jetbrains.com/pycharm/')]
    return render_template('links.html', links_list=links_list)
```

Notes

- These files are *static* files, not executed or processed by Flask in any way
- Static files are usually put inside the `static/` folder
- Flask has a special `/static/<file_to_be_returned_as_is>` endpoint to handle these
- In a live deployment, the web server (not Flask) will serve the static files to get better performance

File `documents.html`

- Template rendered by the `/documents` endpoint
 - extends the base template
 - it builds a table from a Python list passed to the template
 - data read from a CSV file is used to build the list
 - the contents of this CSV file is given next
 - the first column is the *id*, the second is the *topic*, followed by 3 file names, one for each format:
 - Markdown
 - HTML
 - PDF
- the endpoint code and CSV files are shown after the screenshot but before the template



Documents

File: documents.csv

```
id,topic,markdown,html,pdf
0,Course Outline,00_outline.md,00_outline.html,00_outline.pdf
1,Python,01_python.md,01_python.html,01_python.pdf
2,Flask basics,02_flask_basics.md,02_flask_basics.html,02_flask_basics.html
3,Flask templates,03_flask_templates.md,03_flask_templates.html,03_flask_templates.html
```

The /documents Endpoint

```
@app.route('/documents')
def documents():
    # TODO: put the prefix in the app config dictionary
    prefix = '/static/'
    with open('data/documents.csv') as f:
        doc_list = list(csv.reader(f))[1:]
    return render_template('documents.html', doc_list=doc_list, prefix=prefix)
```

File documents.html

```
{% extends "base.html" %}

{% block title %}{{ super() }} - Course Documents{% endblock %}

{% block content %}
    <div class="container">
```

```

<h1>Course Documents</h1>
<table class="table table-dark table-striped table-hover table-responsive-md">
  <thead>
    <tr>
      <th></th>
      <th>Topic</th>
      <th>Markdown</th>
      <th>HTML</th>
      <th>PDF</th>
    </tr>
  </thead>
  <tbody>
    {% for doc in doc_list %}
      <tr>
        {% for col in doc[:2] %}
          <td>{{ col }}</td>
        {% endfor %}
        {% for col in doc[2:] %}
          <td><a class="btn btn-info" href="{{ prefix }}{{
col }}">{{ col }}</a></td>
        {% endfor %}
      </tr>
    {% endfor %}
  </tbody>
</table>
</div>
{% endblock %} <!-- content -->

```