



Tag 12

Webserver

12. Mai 2022

Ablauf

- Rückblick elfter Tag
- Templating mit Jinja
- Session-Handling
- Projektarbeit

Rückblick Flask

Templating

```
<html><head><title>Hello World with HTML and CSS</title><link rel="stylesheet" href="/static/stylesheet.css" /></head><body><h1>Hallo World</h1>&#x2B50;Seasons greetings <b>{name}</b>&#x1F332;</body></html>
```

- HTML von */demo2* ist jetzt noch nicht eine ganze Seite, aber schon maximal unübersichtlich.
- Gibt es das auch in hübsch und lesbar?

Ja, fast: jinja templates

template.html.jinja:

```
<html>
<head>
  <title>{{ variable|escape }}</title>
</head>
<body>
  {% for item in item_list %}
    {{ item }}{% if not loop.last %},{% endif %}
  {% endfor %}
</body>
</html>
```

*.py:

```
from flask import render_template
...
def handle_request():
    some_argument = 'irgendwas'
    return render_template('template.html.jinja', some_argument = some_argument)
```

Argumente / Daten

- Beliebige Daten, welche dem Template zur Verfügung stehen sollen.
- Hier eine Instanz vom Typ <InformationBlock>

```
class InformationBlock:
    def __init__(self, title:str, infos:dict):
        self._title = title
        self._infos = infos

    @property
    def title(self):
        return self._title

    @property
    def keys(self):
        return self._infos.keys()

    def get_info(self, key:str):
        return self._infos[key]
```

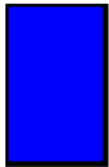
```
<h1>{{info.title|escape}}</h1>
<table>
    <tr>
        <th>Key</th>
        <th>Value </td>
    </tr>
    {% for key in info.keys %}
    <tr>
        <td>{{key|escape}}</td>
        <td>{{info.get_info(key)|escape}}</td>
    </tr>
    {% endfor %}
</table>
```

Übung SVG

- Erstelle und verwende ein Template, welches ein SVG Snippet produziert.

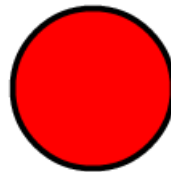
127.0.0.1/svg?shape=rect

Some SVG



127.0.0.1/svg?shape=rect

Some SVG



127.0.0.1/svg

Some SVG

Did not understand 'no arg', so here is an SVG Text

SVG: https://www.w3schools.com/graphics/svg_examples.asp

Flask inkl. Jinja: https://www.tutorialspoint.com/flask/flask_templates.htm

Session handling

- Einfaches, Cookie-basiertes sessionhandling ist bereits vorhanden
- Ein session-Objekt ist ein Dictionary, an welches wir beliebige(*) Daten anhängen können.
- Extrem einfaches An/Abmelden → ob ein Eintrag mit dem Schlüssel *username* im session-Objekt ist

Hands on

- Benutzer soll beim Login noch ein Nickname eingeben, welcher dann ebenfalls angezeigt werden soll,