

Build a Frontend for Your Web App:

From Basic HTML to JS Frameworks

About me

- Anastasia
- Backend engineer and former data scientist
- Live in Berlin

Initial condition

```
app = FastAPI()  
df: pd.DataFrame = get_data()
```

Date	Goal	Actual
Feb 12	10000	13771
Feb 13	10000	9483

Quick and dirty

```
@app.get("/view/basic-steps")
def read_basicsteps() -> Dict[str, Dict[str, int]]:
    df: pd.DataFrame = get_data()
    return df.to_dict(orient="index")
```

JSON	Raw Data	Headers
Save	Copy	Collapse All Expand All Filter JSON
▼ 2023-02-12T00:00:00:		
	Actual:	13771
	Goal:	10000
▼ 2023-02-13T00:00:00:		
	Actual:	9483
	Goal:	10000
▼ 2023-02-14T00:00:00:		
	Actual:	9520
	Goal:	10000
▼ 2023-02-15T00:00:00:		
	Actual:	10461
	Goal:	10000
▼ 2023-02-16T00:00:00:		
	Actual:	10519
	Goal:	10000
▼ 2023-02-17T00:00:00:		
	Actual:	11119
	Goal:	10000
▼ 2023-02-18T00:00:00:		
	Actual:	8813
	Goal:	10000

Pandas to help

```
@app.get("/view/pandas-steps")
def read_pandassteps():
    df = get_data()
    return HTMLResponse(df.to_html())
```

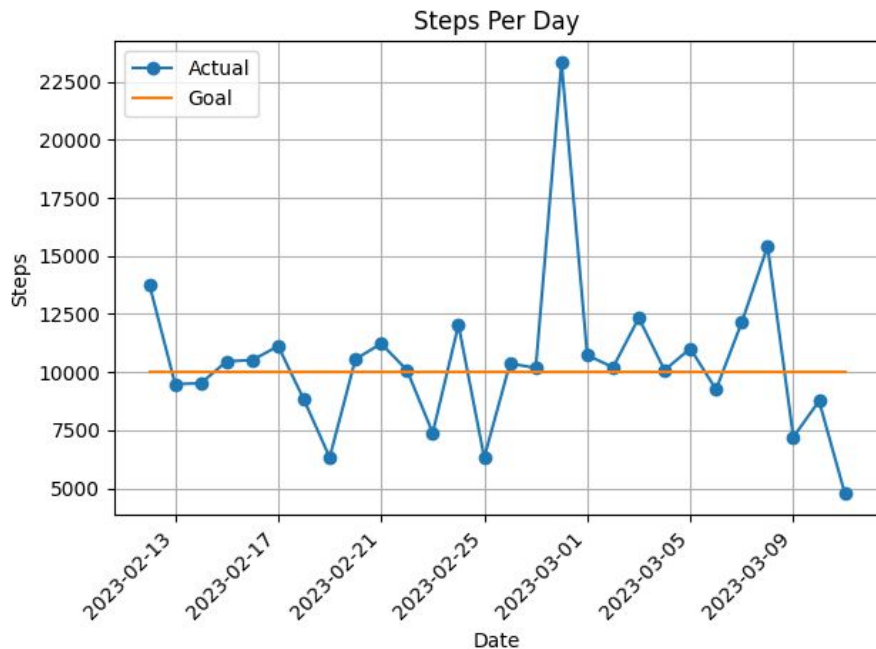
	Actual	Goal
Date		
2023-02-12	13771	10000
2023-02-13	9483	10000

Make a plot

```
@app.get("/view/matplotlib-plot")
def read_matplotlib_plot():
    df = get_data()
    create_basic_plot(df)

    buffer = io.BytesIO()
    plt.savefig(buffer, format="png")
    buffer.seek(0)

    return Response(
        content=buffer.getvalue(),
        media_type="image/png"
    )
```



HTML + Templates

```
from fastapi.templating import Jinja2Templates

templates = Jinja2Templates(directory="templates")

@app.get(
    "/view/template-table",
    response_class=HTMLResponse
)
async def template_table(request: Request):
    df = get_data()
    idx_max = df["Actual"].idxmax()

    return templates.TemplateResponse(
        name="table.html",
        context={
            "request": request,
            "df": df,
            "idx_max": idx_max,
        },
    )
```

Date	Steps
2023-02-12 00:00:00	13771
2023-02-13 00:00:00	9483
2023-02-14 00:00:00	9520

HTML + Templates

```
<table style='border-collapse: collapse;'>
  <thead style='background-color: #F5F5F5;'>
    <tr style='padding: 8px; border: 1px solid #ddd;'>
      <th>Date</th>
      <th>Steps</th>
    </tr>
  </thead>
  <tbody>
    {% for idx, row in df.iterrows() %}
    {% if idx == idx_max %}
      <tr style='background-color: #3b7a57; border: 1px solid #ddd;'>
    {% else %}
      <tr style='border: 1px solid #ddd;'>
    {% endif %}
    <td style='padding: 8px; border: 1px solid #ddd;'>
      {{ idx }}
    </td>
    <td style='padding: 8px; border: 1px solid #ddd;'>
      {{ row["Actual"] }}
    </td>
    </tr>
    {% endfor %}
  </tbody>
</table>
```

Date	Steps
2023-02-12 00:00:00	13771
2023-02-13 00:00:00	9483
2023-02-14 00:00:00	9520

HTML + Templates

```
<table style='border-collapse: collapse;'>
  <thead style='background-color: #F5F5F5;'>
    <tr style='padding: 8px; border: 1px solid #ddd;'>
      <th>Date</th>
      <th>Steps</th>
    </tr>
  </thead>
  <tbody>
    {% for idx, row in df.iterrows() %}
      {% if idx == idx_max %}
        <tr style='background-color: #3b7a57; border: 1px solid #ddd;'>
      {% else %}
        <tr style='border: 1px solid #ddd;'>
      {% endif %}
      <td style='padding: 8px; border: 1px solid #ddd;'>
        {{ idx }}
      </td>
      <td style='padding: 8px; border: 1px solid #ddd;'>
        {{ row["Actual"] }}
      </td>
    </tr>
  {% endfor %}
</tbody>
</table>
```

Date	Steps
2023-02-12 00:00:00	13771
2023-02-13 00:00:00	9483
2023-02-14 00:00:00	9520

HTML + Templates

```
<table style='border-collapse: collapse;'>
  <thead style='background-color: #F5F5F5;'>
    <tr style='padding: 8px; border: 1px solid #ddd;'>
      <th>Date</th>
      <th>Steps</th>
    </tr>
  </thead>
  <tbody>
    {% for row in rows %}
      {% if idx == idx_max %}
        <tr style='background-color: #3b7a57; border: 1px solid #ddd;'>
      {% else %}
        <tr style='border: 1px solid #ddd;'>
      {% endif %}
        <td style='padding: 8px; border: 1px solid #ddd;'>
          {{ idx }}
        </td>
        <td style='padding: 8px; border: 1px solid #ddd;'>
          {{ row["Actual"] }}
        </td>
      </tr>
    {% endfor %}
  </tbody>
</table>
```

Date	Steps
2023-02-12 00:00:00	13771
2023-02-13 00:00:00	9483
2023-02-14 00:00:00	9520

Do you want to upscale?

February 2023

today < >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
			Steps: 10729	Steps: 10204	Steps: 12330	Steps: 10083
5	6	7	8	9	10	11
Steps: 11015	Steps: 9251	Steps: 12149	Steps: 15409	Steps: 7206	Steps: 8756	Steps: 4773
12	13	14	15	16	17	18
Steps: 13771	Steps: 9483	Steps: 9520	Steps: 10461	Steps: 10519	Steps: 11119	Steps: 8813
19	20	21	22	23	24	25
Steps: 6328	Steps: 10559	Steps: 11224	Steps: 10084	Steps: 7382	Steps: 12032	Steps: 6325
26	27	28	1	2	3	4
Steps: 10370	Steps: 10179	Steps: 23310				

JS Frameworks

JS Frameworks

- Angular
- Vue
- React
- Svelte

Why Svelte?

Web frameworks

Newcomer Svelte takes the top spot as the most loved framework. React is the most wanted, desired by one in four developers.

Loved vs. Dreaded

Wa

66,202 responses



Why, Svelte?

Web frameworks and technologies

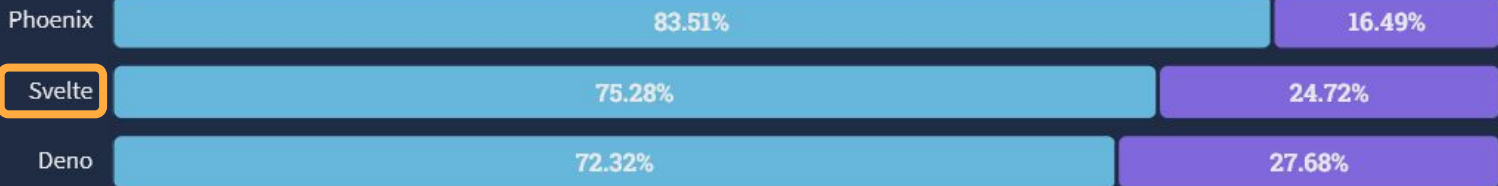
Phoenix overtakes Svelte's spot as the most loved web framework.

Angular.js is in its third year as the most dreaded. React.js completes its fifth year as most wanted.

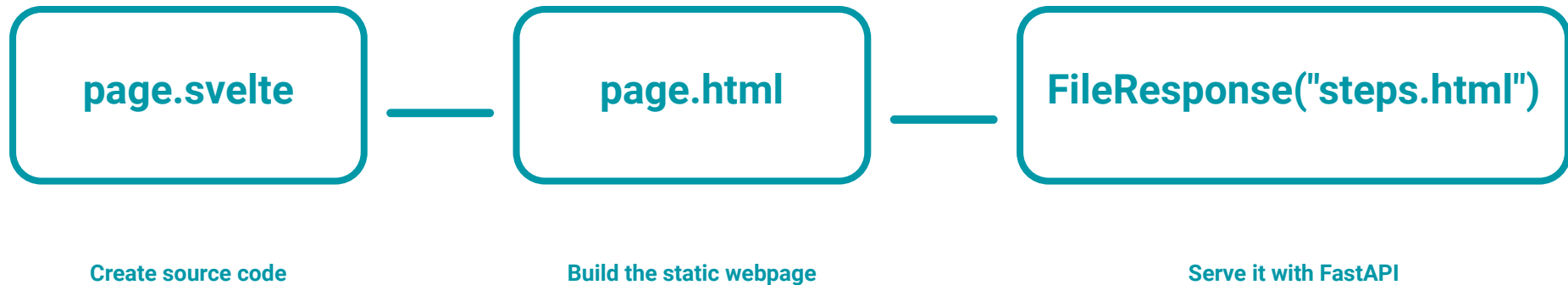
Loved vs. Dreaded

Wanted vs. Dreaded

57,654 responses



Plan



Create Svelte project

```
npm create svelte@latest client  
cd client  
npm install  
npm run dev
```

Create a page

```
<main>
  {#if stepsData}
    <div class="calendar">
      <FullCalendar
        {options}
      />
    </div>
  {:else}
    <p>Loading data...</p>
  {/if}

</main>
```

Static site generation

Svelte documentation will guide you through it

Static site generation

[Edit this page on GitHub](#)

To use SvelteKit as a static site generator (SSG), use `adapter-static`.

This will prerender your entire site as a collection of static files. If you'd like to prerender only some pages and dynamically server-render others, you will need to use a different adapter together with the `prerender` option.

Usage

Install with `npm i -D @sveltejs/adapter-static`, then add the adapter to your `svelte.config.js`:

```
svelte.config.js

import adapter from '@sveltejs/adapter-static';

export default {
  kit: {
    adapter: adapter({
      // default options are shown. On some platforms
      // these options are set automatically – see below
      pages: 'build',
      assets: 'build',
      fallback: null,
      precompress: false,
      strict: true
    })
  }
};
```

...and add the `prerender` option to your root layout:

```
src/routes/+layout.js

// This can be false if you're using a fallback (i.e. SPA mode)
export const prerender = true;
```

Back to Python

```
location = os.path.dirname(os.path.realpath(__file__))
frontend = os.path.join(location, "client", "build")

app.mount("/", StaticFiles(directory=frontend, html=True))

@app.get("/svelte-steps")
def js_frontend():
    return FileResponse(os.path.join(frontend, "steps.html"))
```

You are beautiful

February 2023

today < >

Sun	Mon	Tue	Wed	Thu	Fri	Sat
29	30	31	1	2	3	4
			Steps: 10729	Steps: 10204	Steps: 12330	Steps: 10083
5	6	7	8	9	10	11
Steps: 11015	Steps: 9251	Steps: 12149	Steps: 15409	Steps: 7206	Steps: 8756	Steps: 4773
12	13	14	15	16	17	18
Steps: 13771	Steps: 9483	Steps: 9520	Steps: 10461	Steps: 10519	Steps: 11119	Steps: 8813
19	20	21	22	23	24	25
Steps: 6328	Steps: 10559	Steps: 11224	Steps: 10084	Steps: 7382	Steps: 12032	Steps: 6325
26	27	28	1	2	3	4
Steps: 10370	Steps: 10179	Steps: 23310				

Wrap up

Pandas df as a table / Matplotlib plots as bytes

★ lightning fast

★ no learning curve for python devs

☁️ hard to make sense of data

☁️ not much power over output

HTML / templating

★ simple

☁️ cumbersome

JS framework

★ powerful

☁️ steep learning curve for python devs

☁️ time consuming

Reading list

[Source code of the talk](#)

[MDN Web Docs: Understanding client-side JavaScript frameworks](#)

[Stack Overflow Annual Developer Survey](#)

[Svelte: Static site generation](#)

React examples: [Tech week](#), [silencio.es](#), [Zelt](#)

