

# LU10a - Statische Webseiten veröffentlichen (Nachtrag)

## Definitionen / Unterscheidung

<b>Browser</b> <ul style="list-style-type: none"> <li>• Applikation, um Webinhalte (HTML, CSS, JS) darzustellen</li> </ul>	<b>Webserver</b> <ul style="list-style-type: none"> <li>• Nimmt HTTP(S)-Requests entgegen und liefert Antworten zurück an den Browser</li> <li>• Die Inhalte können statisch oder dynamisch sein</li> </ul>
<b>Frontend</b> <ul style="list-style-type: none"> <li>• Der Teil der Webseite, der an den Aufrufer ausgeliefert wird und im Browser läuft</li> </ul>	<b>Backend</b> <ul style="list-style-type: none"> <li>• Läuft auf dem Server und verarbeitet Anfragen</li> <li>• Stellt Daten oder Funktionen über APIs bereit</li> </ul>
<b>Statische Webseite</b> <ul style="list-style-type: none"> <li>• Besteht aus fertigen Dateien (HTML, CSS, JS), die direkt ausgeliefert werden</li> </ul>	<b>Dynamische Webseite</b> <ul style="list-style-type: none"> <li>• Inhalte werden abhängig vom Request zur Laufzeit erzeugt (.php → .html, ...)</li> </ul>

## Static vs dynamic hosting

Ob etwas statisch deploybar ist, hängt nicht von den Features ab, sondern davon, ob der Server zur Laufzeit am Rendering beteiligt ist.

Es ist also eine Architekturfrage, ob man Server-Side-Rendering (SSR) einsetzt oder nicht.

Salopp gesagt, können die meisten Webapplikationen in ein statisches Frontend und ein (allfälliges) dynamisches Backend aufgeteilt werden.

Statische Webseiten sind möglich bei

- Vanilla
- Angular
- React
- Vite
- Astro (default)

Ausnahmen sind

- PHP
- Next.js mit API/SSR
  - `getServerSideProps`, Routing (`route.js`, `route.ts`, `„/api/*“`), `NextResponse`, `cookies()`, `headers()`
  - Express mit Templates (EJS, Pug)

## Next.js

```
const nextConfig = {  
  output: 'export',  
};  
  
module.exports = nextConfig;
```

```
npm run build
```

From:  
<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:  
<https://wiki.bzz.ch/de/modul/ffit/3-jahr/cicd/learningunits/lu10/a?rev=1777390287>

Last update: **2026/04/28 17:31**

