

# Er Next.js egentlig det neste?

Toralf Frich

# Hvem er jeg?

Toralf



jobber i

BEKK

og sitter hos



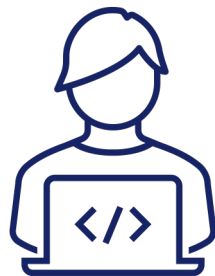
der jeg bruker

NEXT.js

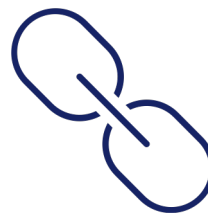
# Hva skal vi snakke om i dag?



Rendering



Next.js



Rendering i Next.js

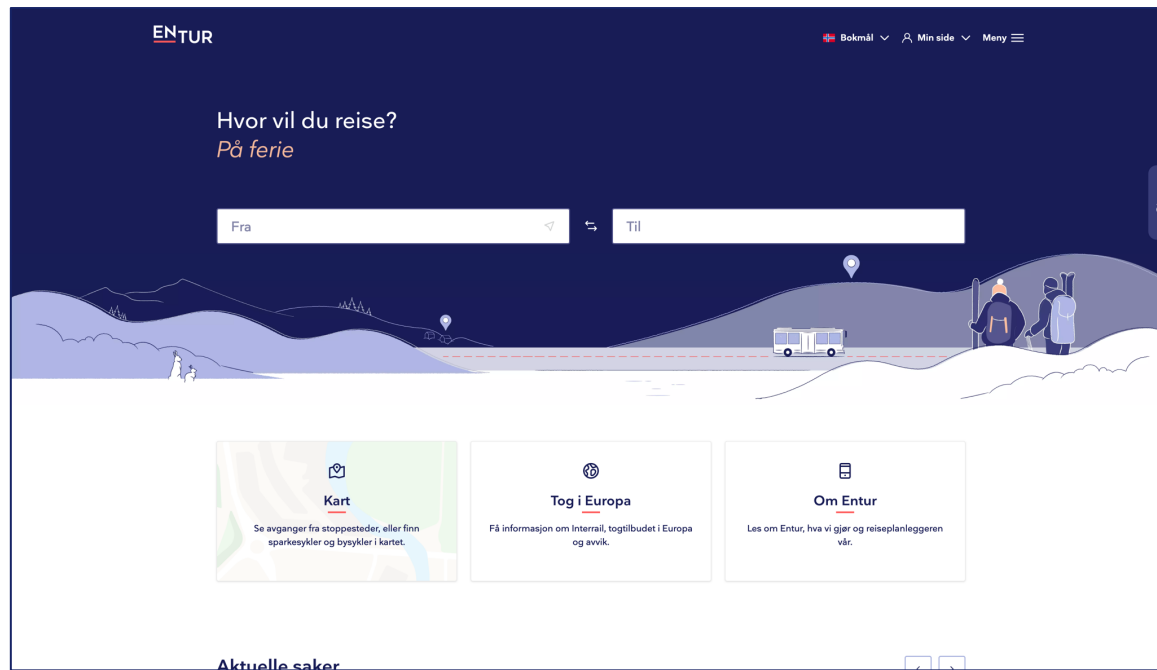
# Rendering er veien fra html...

```
<html>
```

```
  <body>
```

```
    <script>
```

# ...til en nettside som vi kan bruke!





Klient



Pre-rendering



Klient

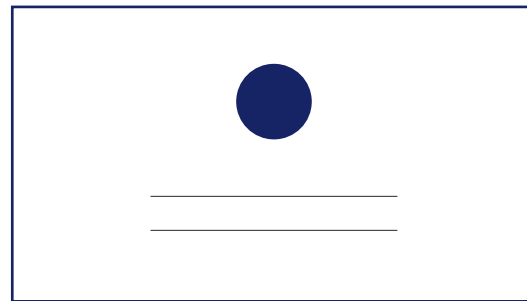
# Client side rendering



Sideforespørsel



JS kjører



HTML blir generert

HTML

JS

CSS





# Client side rendering

Initielt

Nettleseren viser ingenting

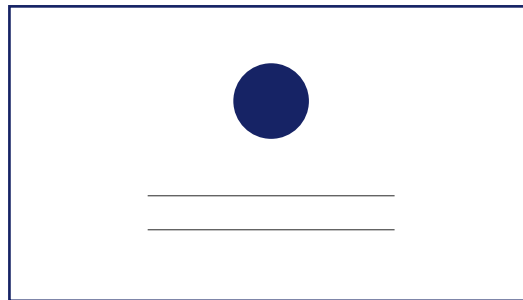


JS kjører

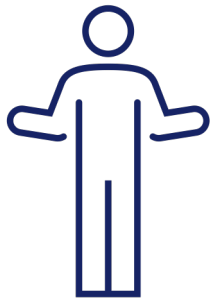


Hydrering

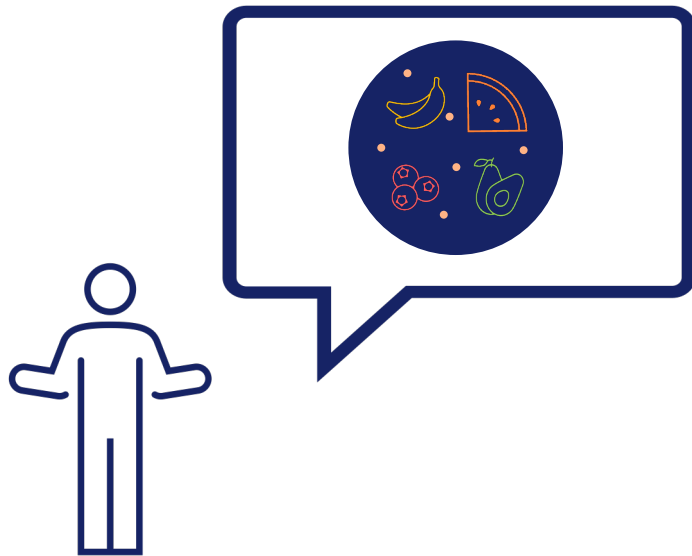
Nettsiden er interaktiv



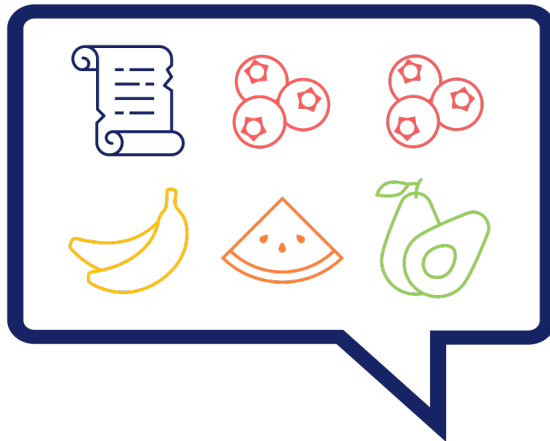
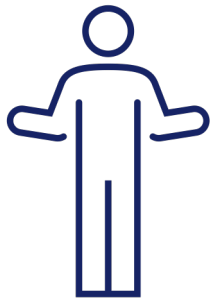
# Client side rendering



# Client side rendering



# Client side rendering



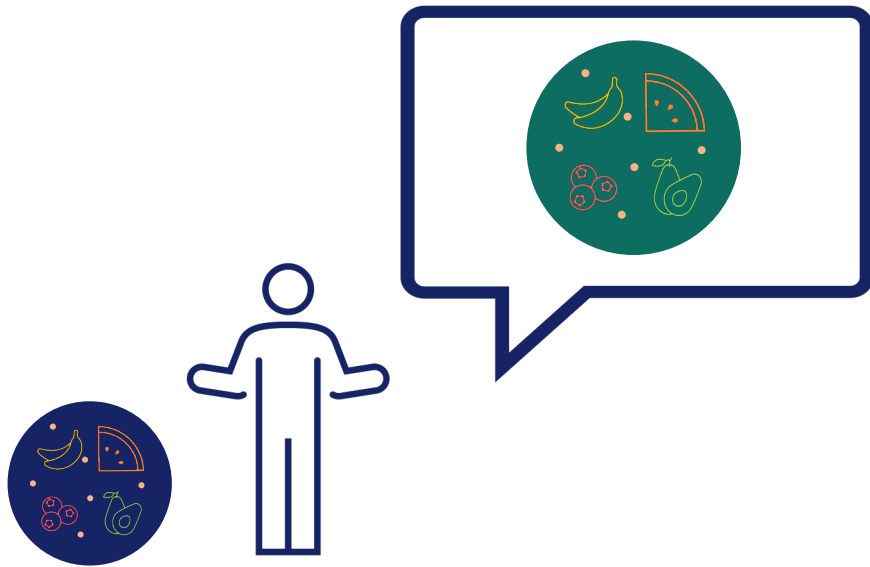
# Client side rendering



# Client side rendering



# Client side rendering



# Client side rendering





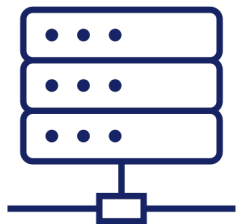
# Client side rendering



entur.no



Pre-rendering

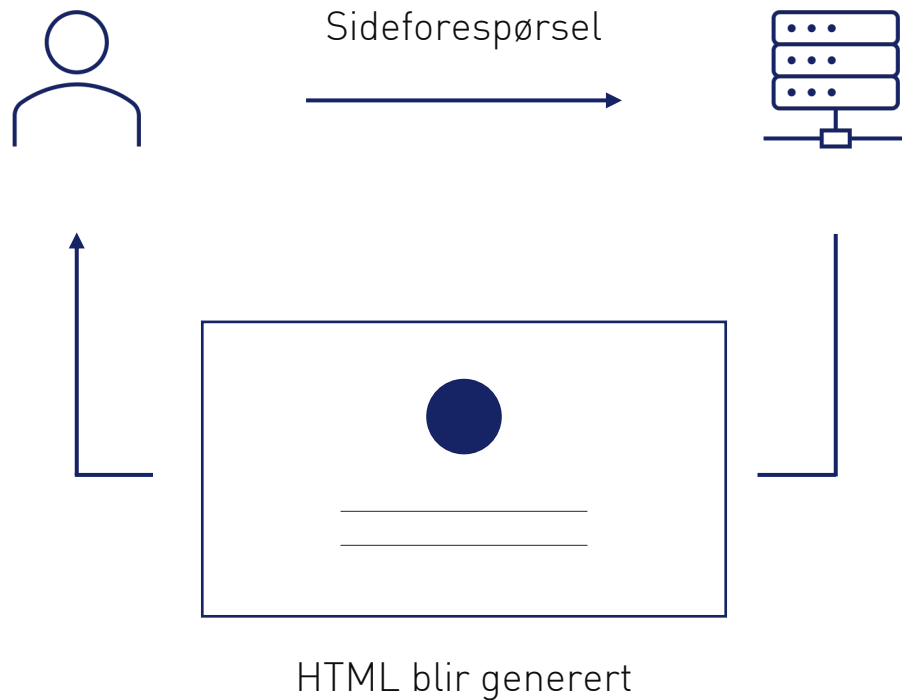


Server side rendering

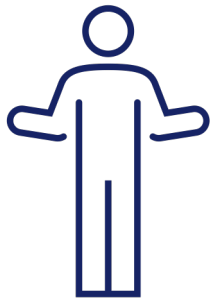


Static generation

# Server side rendering



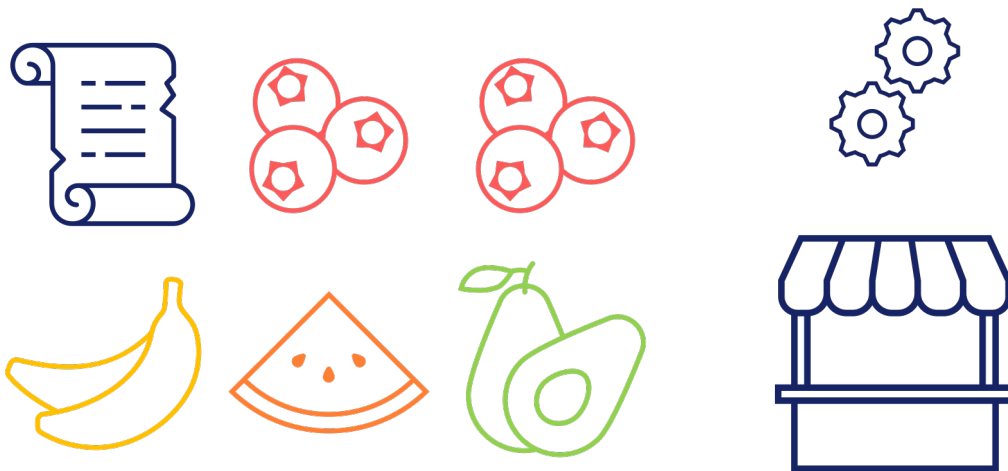
# Server side rendering



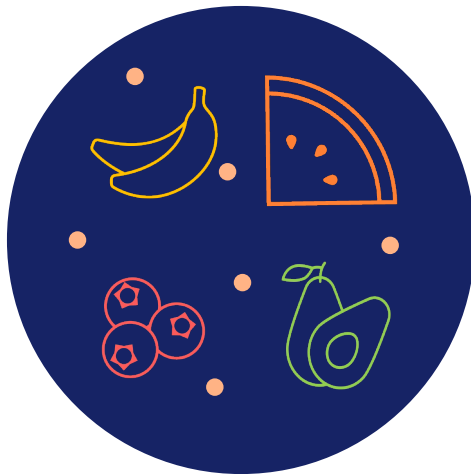
# Server side rendering



# Server side rendering

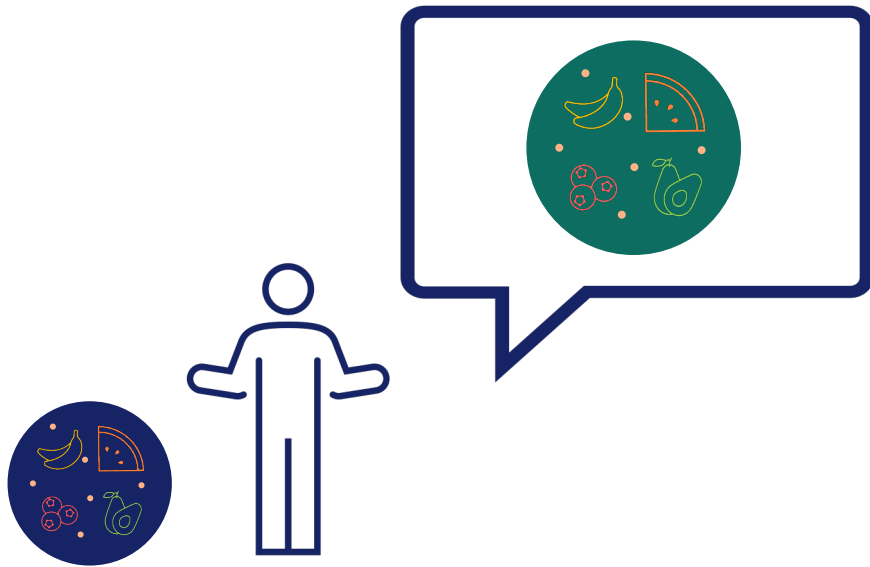


# Server side rendering

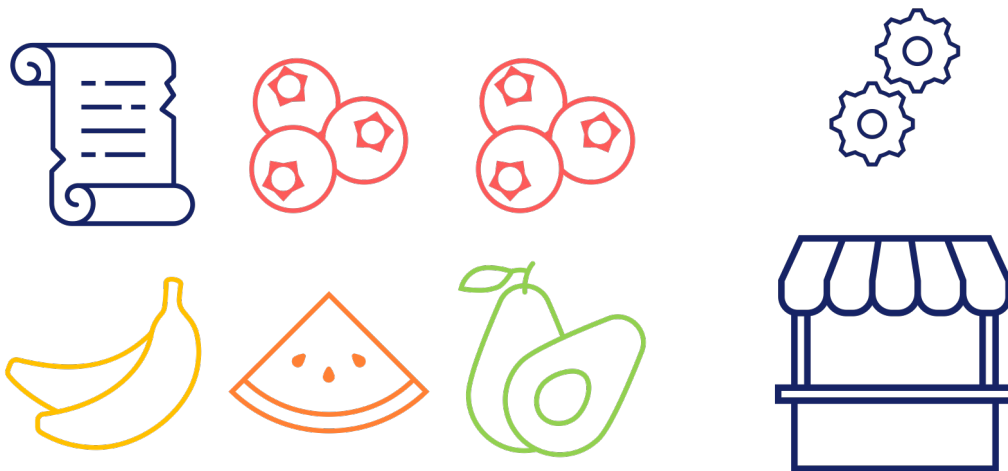




# Server side rendering

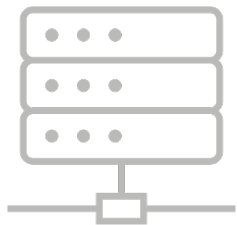


# Server side rendering



# Server side rendering





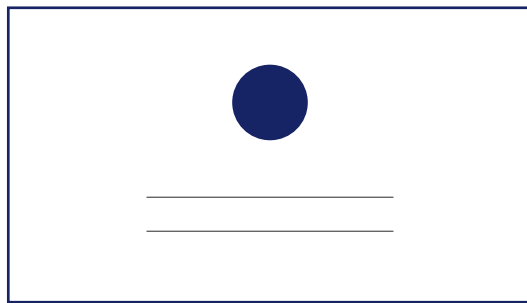
Server side rendering



Static generation

# Static generation

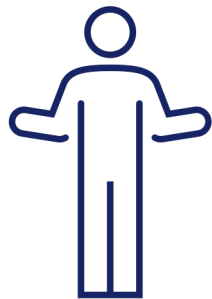
Bygging av applikasjon



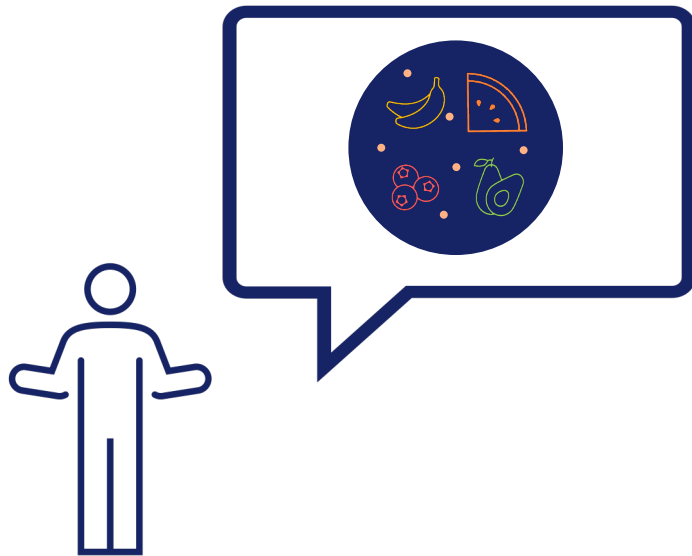
HTML blir generert



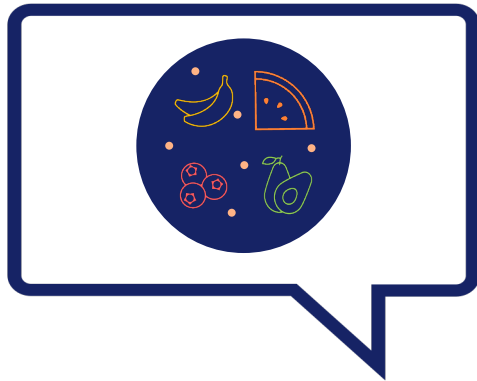
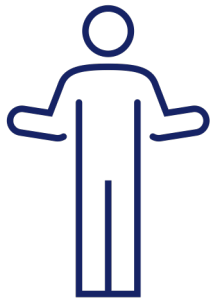
# Static generation



# Static generation

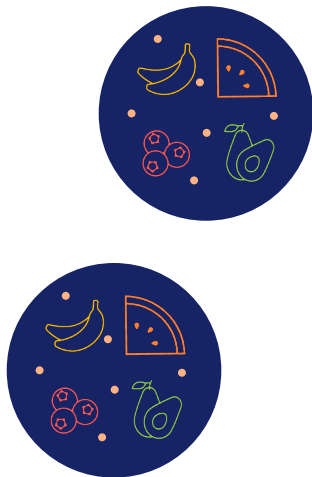


# Static generation

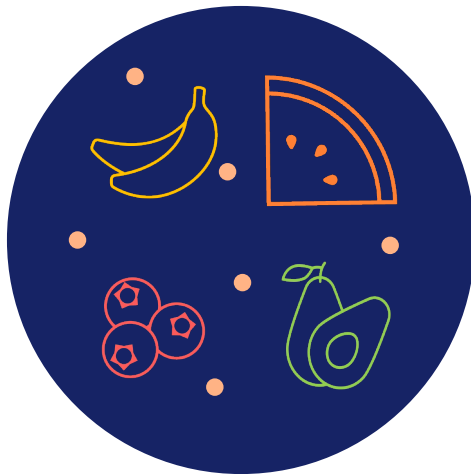




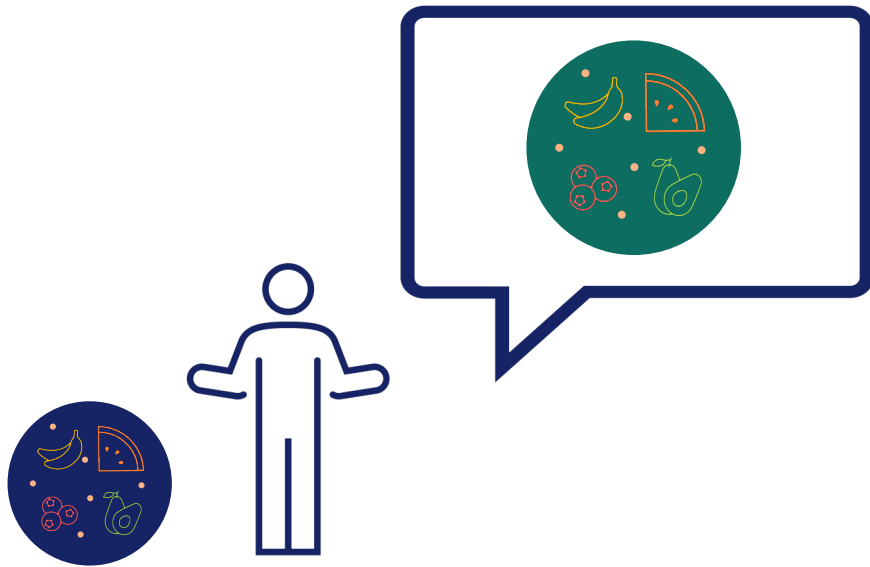
# Static generation



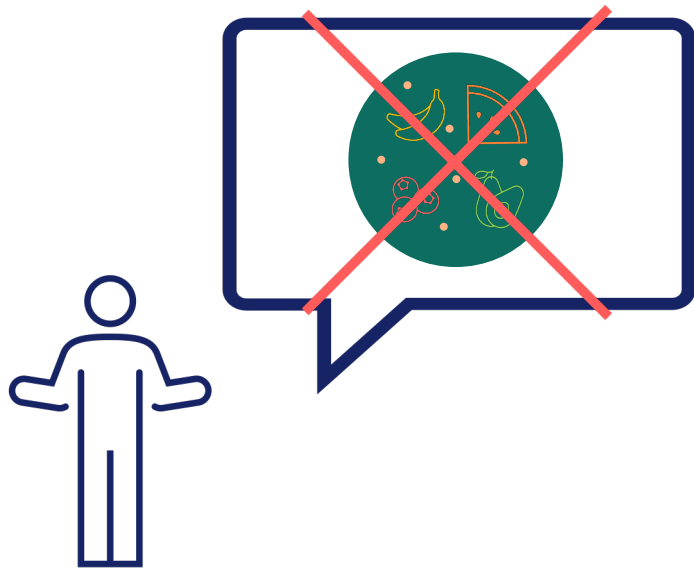
# Static generation



# Static generation



# Static generation

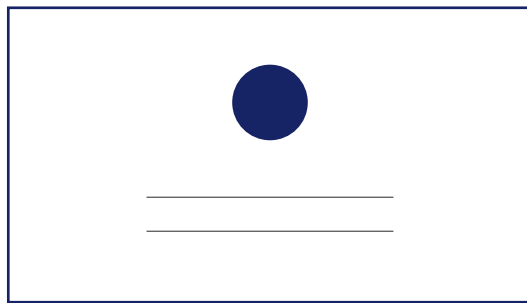


# Static generation med data fetching

Bygging av applikasjon



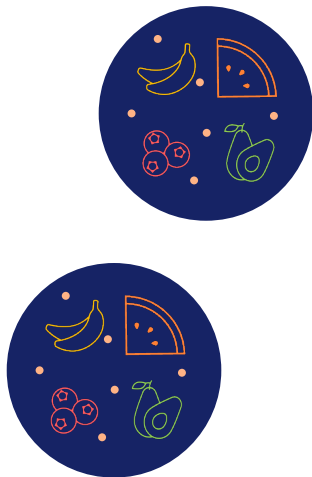
HTML blir generert



# Static generation med data fetching



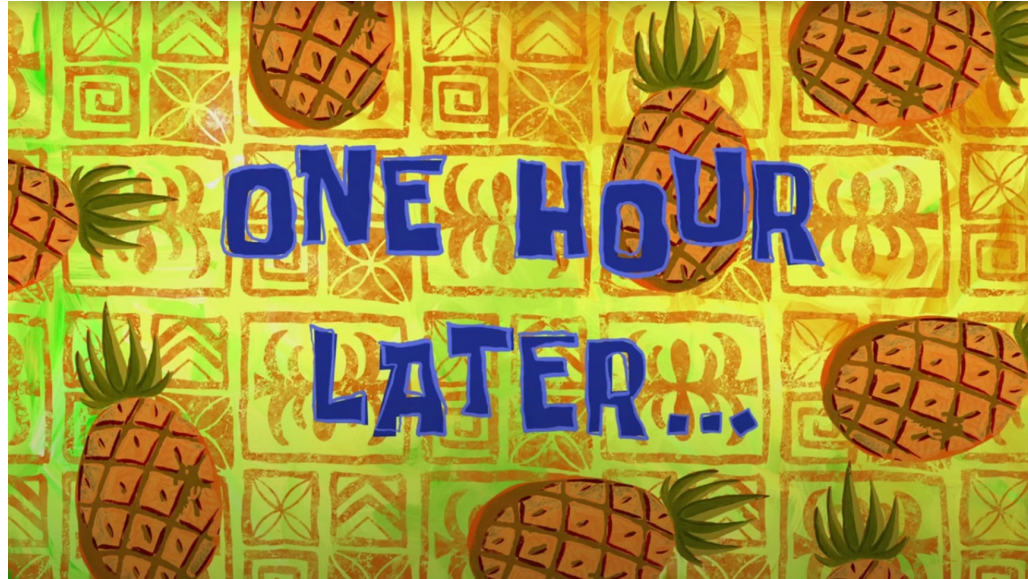
# Static generation med data fetching



# Static generation med data fetching



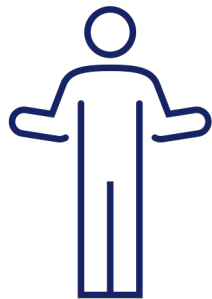




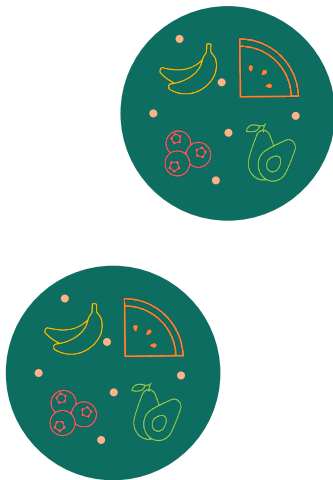
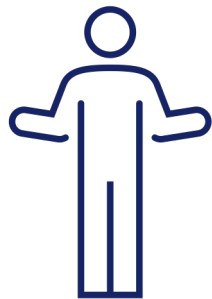
# Static generation med data fetching



# Static generation med data fetching



# Static generation med data fetching

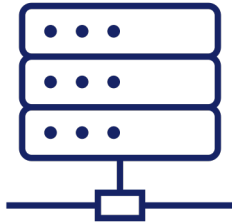


# Static generation med data fetching





Client side rendering



Server side rendering



Static generation

---

Pre-rendering

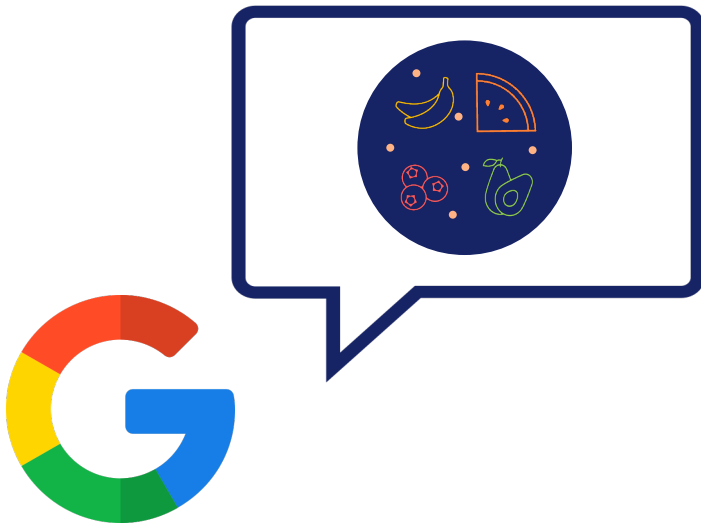
**En liten notis om Search Engine Optimisation (SEO)**

# Client side rendering

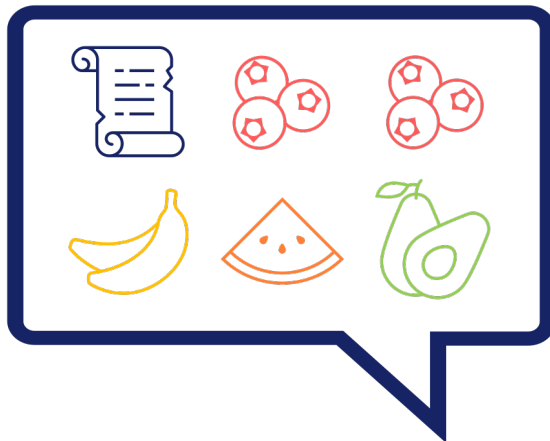




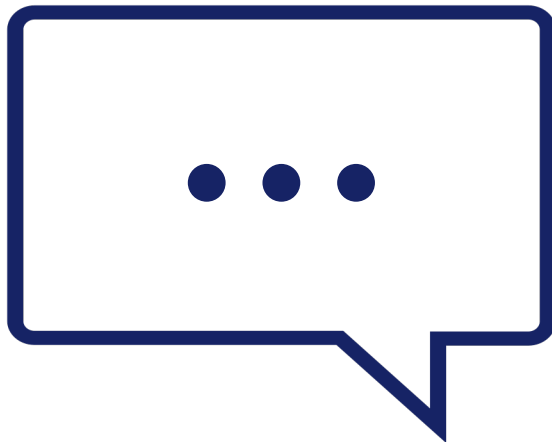
# Client side rendering



# Client side rendering



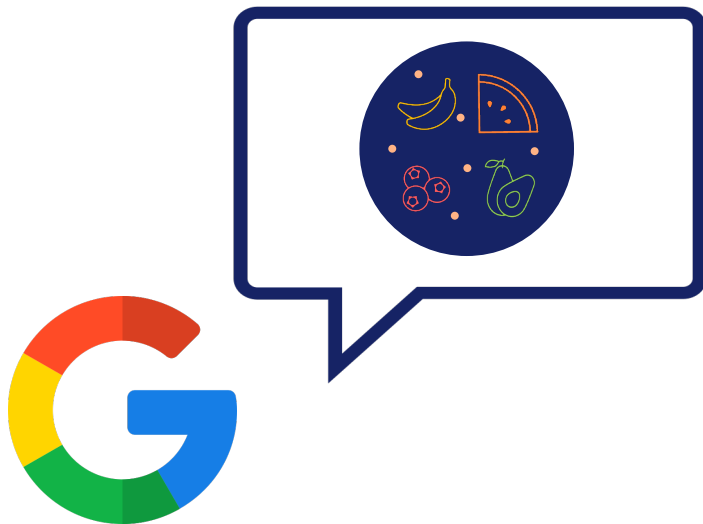
# Client side rendering



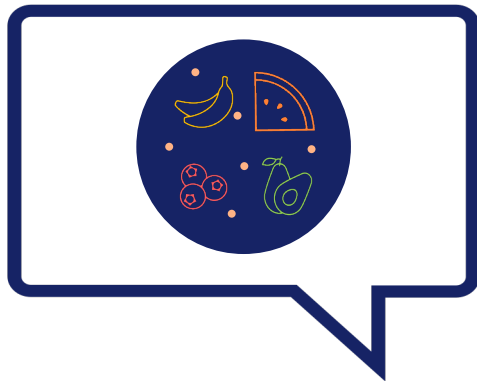
# Server side rendering



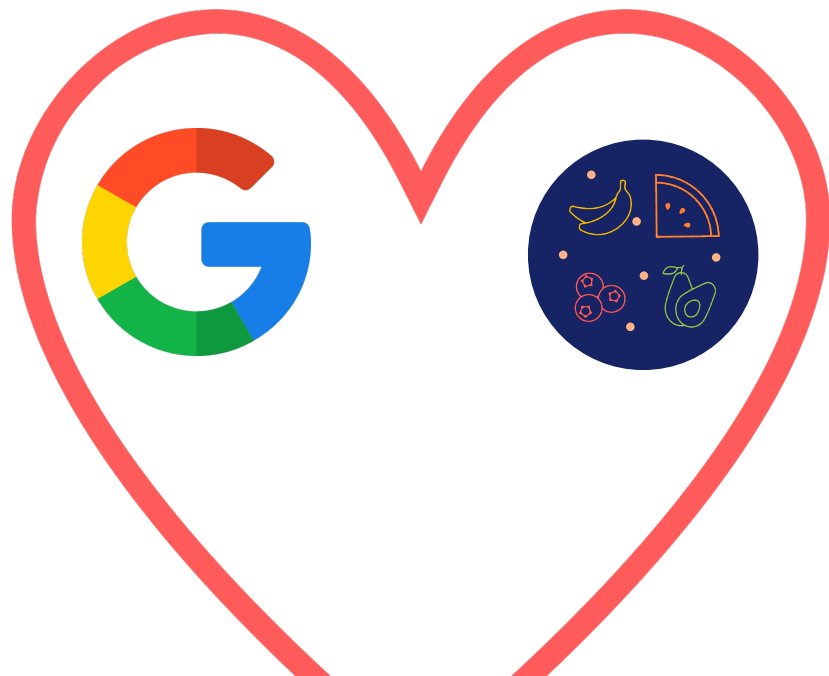
# Server side rendering



# Server side rendering



# Server side rendering



# Dette var mer riktig før!

Google gidder nå å kokkelere litt for å lage ferdig salaten



# Pros



## Client side rendering

Rask når det er mye samhandling

Færre kall mot server

Billigere å «hoste»



## Server side rendering

Rask på å vise brukeren HTML

Færre krav til brukernes enheter

Optimalisere caching



## Static generation

Lynrask ved lasting av statisk innhold

# Når?



Client side rendering

Mye samhandling

SEO ikke er så viktig



Server side rendering

Mindre samhandling

Data fetching

SEO er viktig



Static generation

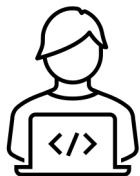
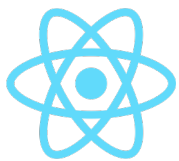
Statisk innhold

# Konklusjon

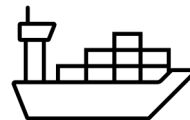
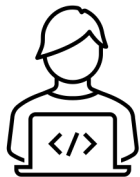
Det beste er nok en kombo av CSR, SSR og SSG!

NEXT.js

Next.js er et rammeverk



~~NEXT~~.JS



...og masse mer!

# Hva følger med?

Routing

API routes

Kodesplitting og bundling

Optimaliseringer

CSR, SSR, SG

Node server

Standardkomponenter

En del funksjoner og hooks

# Routing i Next

/app

/salat

/page.tsx

/tomat

/page.tsx



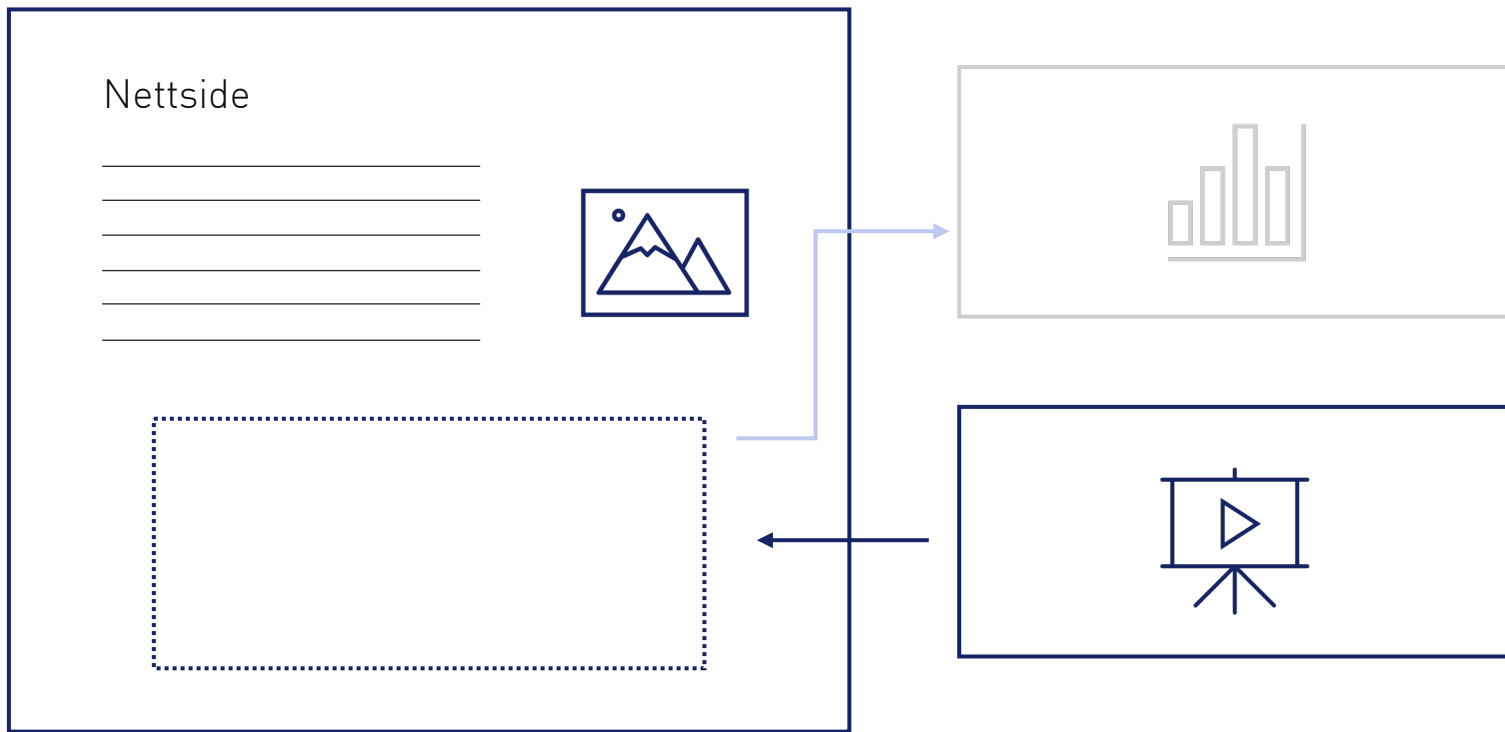
# Code splitting i Next

Inkludert!

Filer under en route blir til bundles

Muliggjør «lazy-loading»

# Partial rendering i Next.js

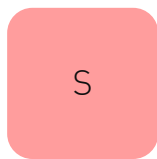


# Node server i Next.js

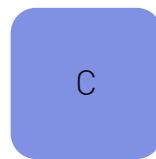
Rendering med

~~NEXT~~.JS

# Server Components og Client Components

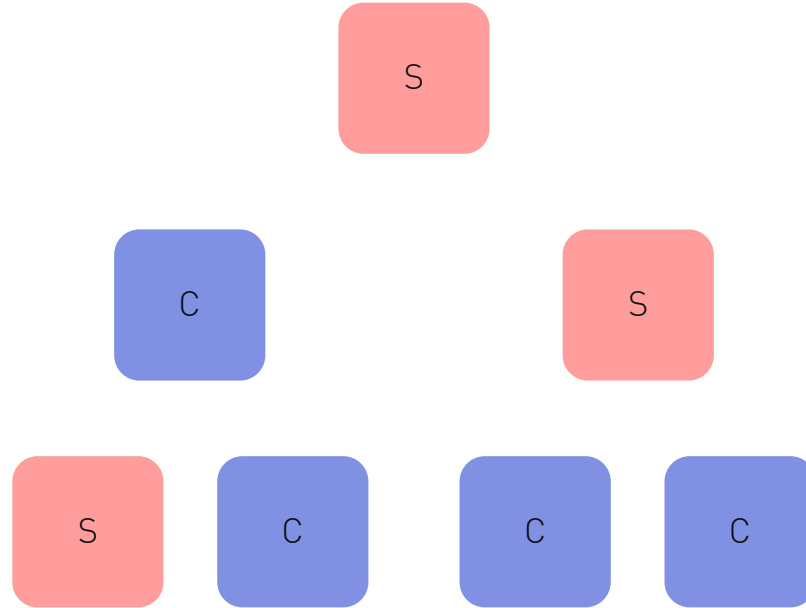


Server



Klient

# Server Components og Client Components



# Server Components og Client Components

Klient

```
'use client';

import { useState } from 'react';

export default const Counter = () => {
  const [count, setCount] = useState(0);

  return (
    <div>
      <p>You clicked {count} times</p>
      <button onClick={() => setCount(count + 1)}>Click me</button>
    </div>
  );
}
```

# Server Components og Client Components

## Server

```
import Salat from "./Salat";
import Tomat from "./Tomat";

export default async function Home() {
  const dataRes = await fetch("http://localhost:8080/product.json");
  const data = (await dataRes.json()) as {
    navn: string;
    beskrivelse: string;
  };

  return (
    <div>
      <Salat name={data.navn} beskrivelse={data.beskrivelse} />
      <div>
        <Tomat name={data.navn} beskrivelse={data.description} />
      </div>
    </div>
  );
}
```



# Streaming i Next.js

Nettside

---

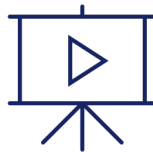
---

---

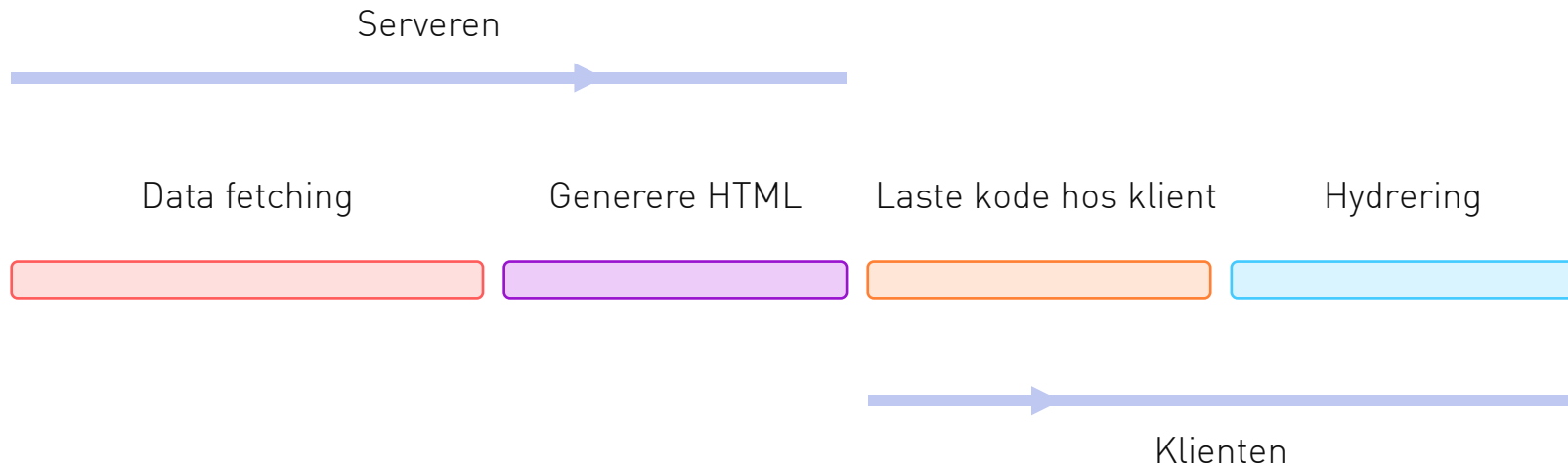
---

---

---

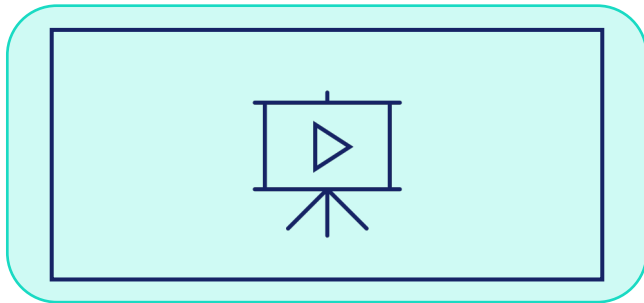
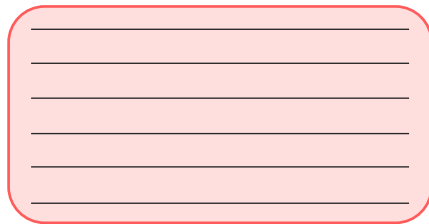


# Streaming i Next.js

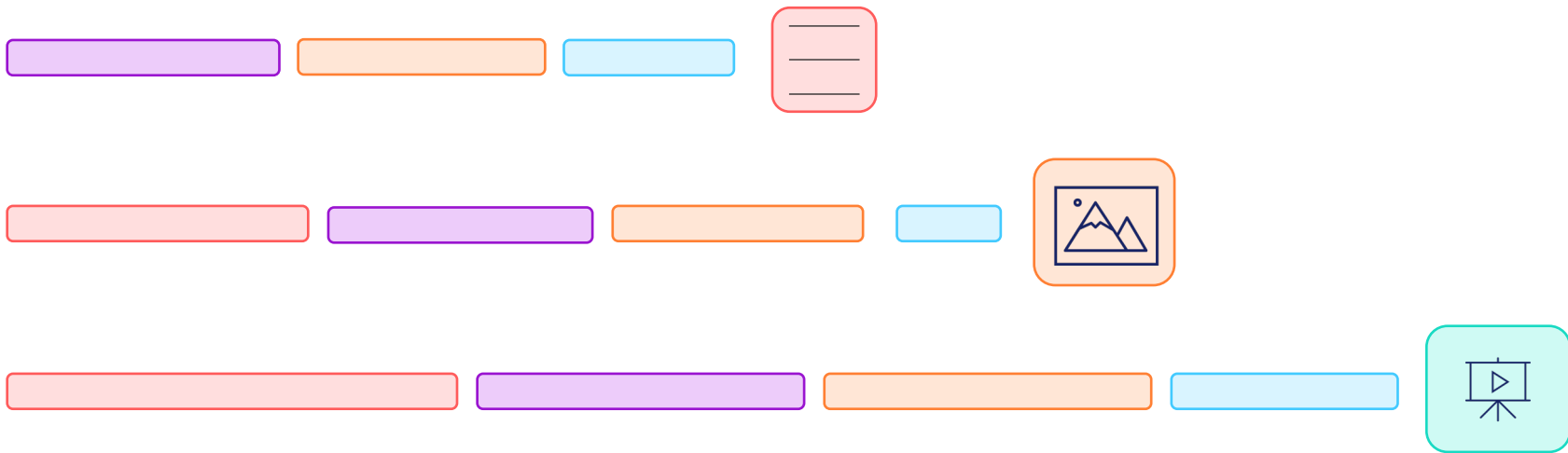


# Streaming i Next.js

Nettside



# Streaming i Next.js



# Streaming i Next.js

Med React Suspense



```
import { Suspense } from "react";
import { PostFeed, Weather } from "../Components";

export default const Posts = () => {
  return (
    <section>
      <Suspense fallback={<p>Loading feed...</p>}>
        <PostFeed />
      </Suspense>
      <Suspense fallback={<p>Loading weather...</p>}>
        <Weather />
      </Suspense>
    </section>
  );
}
```

# Hva er bra med Next.js?

Veldig raskt

Høy utviklingshastighet

Gir deg greier og optimaliseringer du vanligvis måtte funnet ut av selv

Lett å deploye

# Hva kan være kjipt med Next.js?

Kan fort miste mange av fordelene hvis man tukler med standardting

Ikke alltid like lett å tilpasse



**Andrew Clark**

@acdlite



If you use React, you should be using a React framework. If your existing app doesn't use a framework, you should incrementally migrate to one. If you're creating a new React project, you should use a framework from the beginning.

8:52 PM · Jan 23, 2023 · **368.8K** Views

---

**104** Retweets   **60** Quote Tweets   **1,088** Likes





**Andrew Clark** @acdlite · 12h



If you choose to forgo a framework, what you're really choosing is to build your own bespoke framework, one that's likely much worse than what you'd get off the shelf. Even if you think your bespoke framework is better than the alternatives today, will that be true in a year?



2



7



112



29.6K



**Andrew Clark** @acdlite · 12h



It takes lots of time and energy to keep up with the state of the art. Are you ready to become a framework maintainer? Or are there better things you should be doing with your time?



6



1



81



27.4K





**Andrew Clark** @acd lite · 12h



Replying to @acd lite

Your React framework of choice should have built-in solutions for data fetching, routing, and server rendering. Frameworks don't treat these as independent concerns — they provide deeply integrated solutions that are easy to use and result in excellent performance out of the box.



2



3



110



31.3K



**Andrew Clark** @acd lite · 12h



Which framework you choose is an important decision, but it's much less important than choosing to go with a framework in the first place. Personally, I'd go with Next.js, Remix, or Expo.



7



5



135



30.5K





**lil uzi perf**  @ken\_wheeler · 9h



Replying to @acd lite

What if you don't need SSR



10



1



33



15.5K



**Andrew Clark** @acd lite · 9h



Replying to @ken\_wheeler

Same answer



**Sebastian Markbåge** @sebmarkbage · 10h

Replying to @sebmarkbage @riadbenguella and 2 others

There's so much a framework does beyond the highlighted features. Much of the work just goes into making sure other tools work together and paving over ecosystem incompatibilities. With enough eyes on it, you don't even notice that it would've been an issue.



11



17.7K



# Konklusjon

Det finnes mange rammeverk

Kanskje ikke så farlig hvilket man velger

Men på sikt, så er det rammeverk som gjelder

# Er Next.js det neste for deg?

Sjekk det ut på <https://nextjs.org/> da vel 📈

- Takk for meg 🙌