

Er Next.js egentlig det neste?

Toralf Frich

Hvem er jeg?

Toralf



jobber i

BEKK

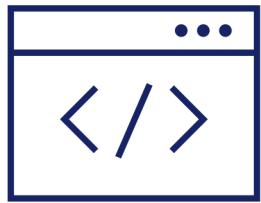
og sitter hos



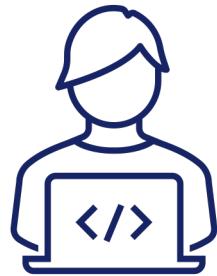
der jeg bruker

NEXT.JS

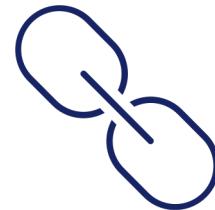
Hva skal vi snakke om i dag?



Rendering



Next.js



Rendering i Next.js

Rendering er veien fra html...

```
<html>  
  <body>  
    <script>
```

...til en nettside som vi kan bruke!

The screenshot shows the homepage of the Entur website. The header features the Entur logo and language selection (Bokmål). The main search bar asks "Hvor vil du reise? På ferie". Below the search bar are fields for "Fra" and "Til". The background is a stylized illustration of a landscape with mountains, a bus, and two people with backpacks. At the bottom, there are three cards: "Kart" (Map), "Tog i Europa" (Trains in Europe), and "Om Entur" (About Entur).

EN_TUR

Bokmål Min side Meny

Hvor vil du reise?
På ferie

Fra Til

Chat

Kart

Se avganger fra stoppesteder, eller finn sparkesykler og bysykler i kartet.

Tog i Europa

Få informasjon om Interrail, togtilbuddet i Europa og avgikk.

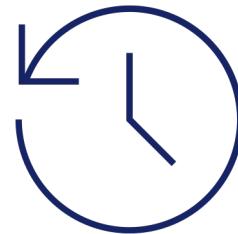
Om Entur

Les om Entur, hva vi gjør og reiseplanleggeren vår.

Aktuelle saker



Klient

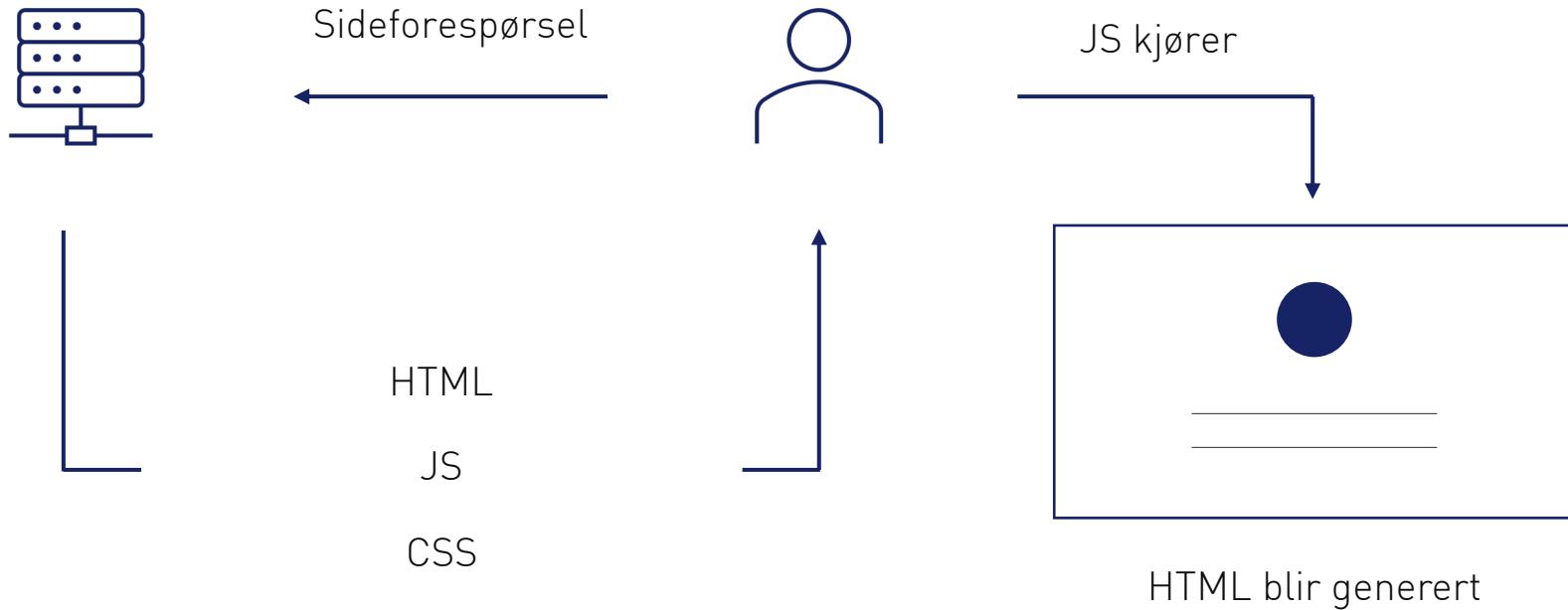


Pre-rendering



Klient

Client side rendering



Client side rendering

Initielt

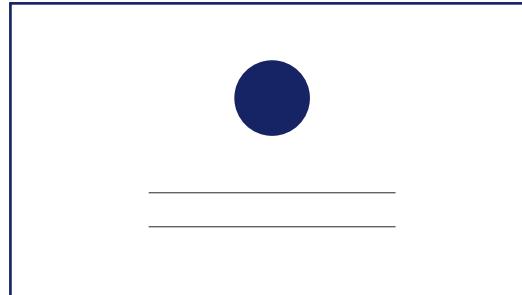
Nettleseren viser ingenting



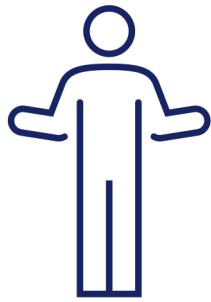
JS kjører
→

Hydrering

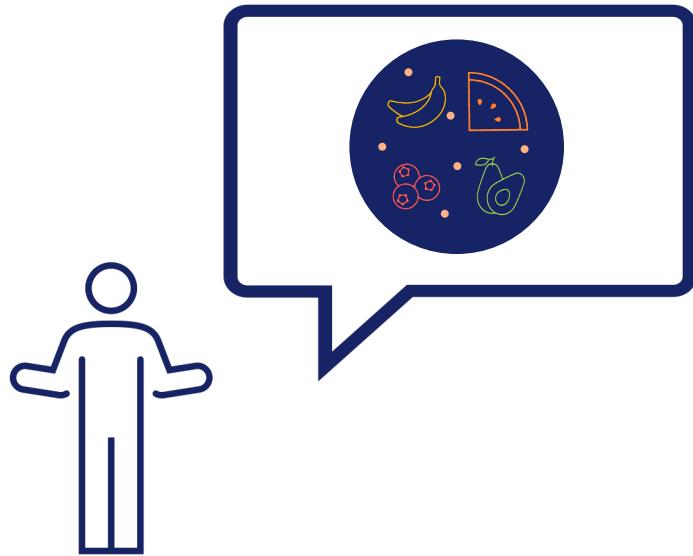
Nettsiden er interaktiv



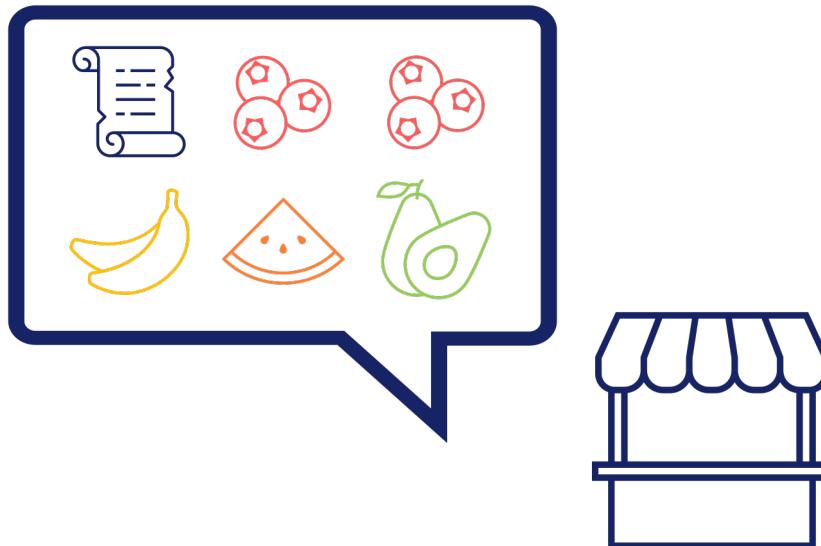
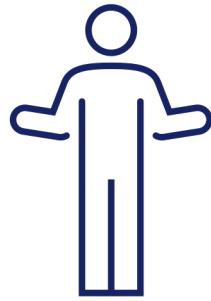
Client side rendering



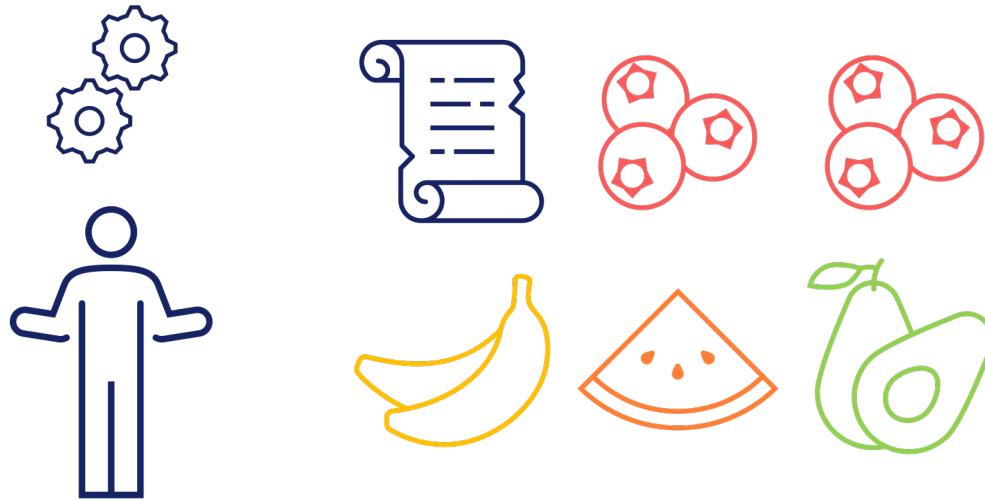
Client side rendering



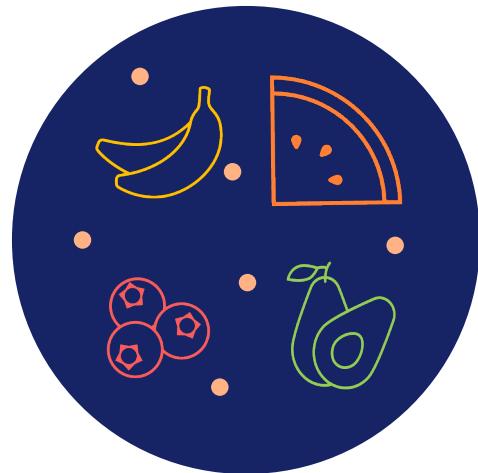
Client side rendering



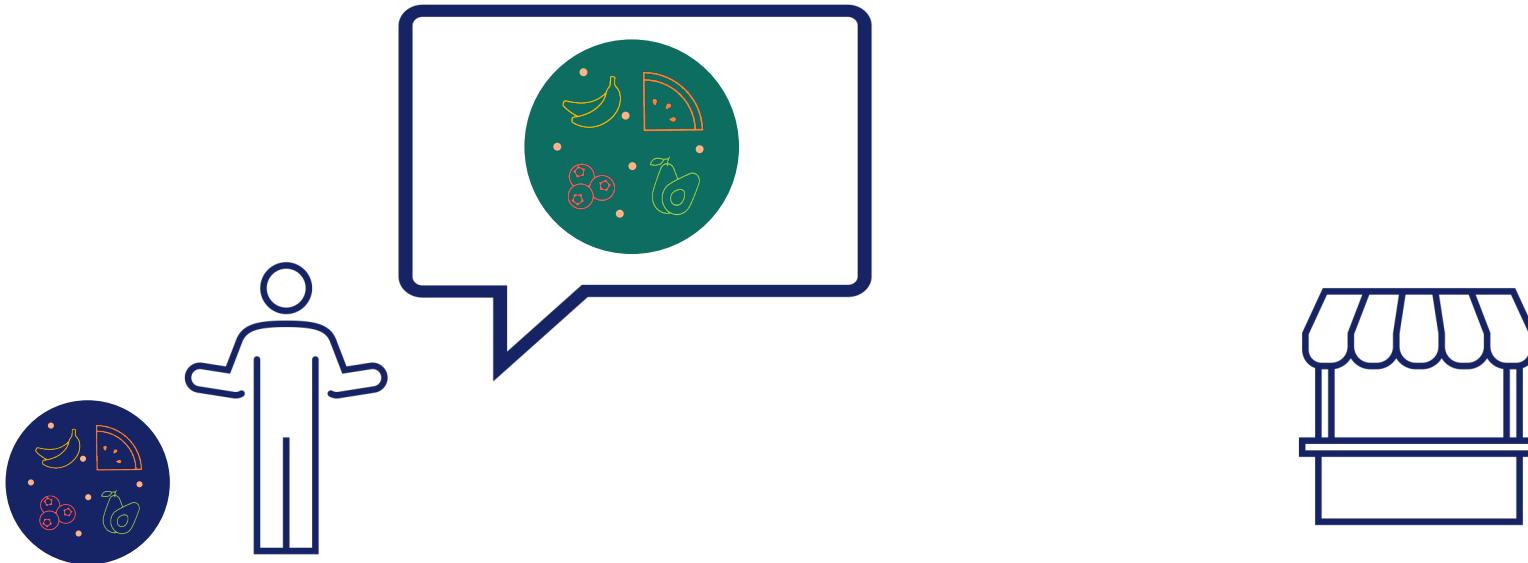
Client side rendering



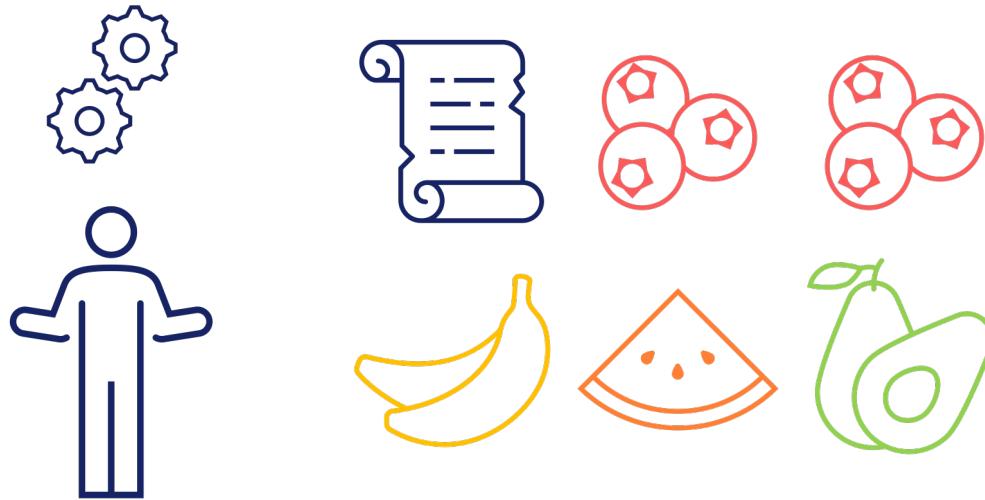
Client side rendering



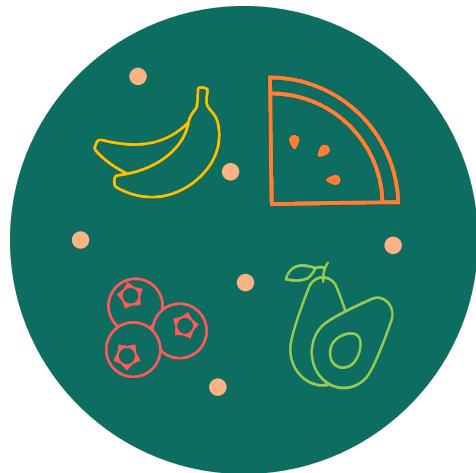
Client side rendering



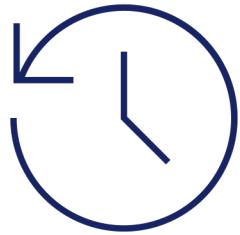
Client side rendering



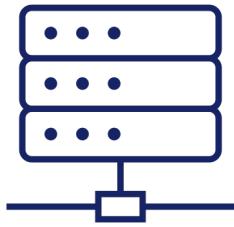
Client side rendering



entur.no



Pre-rendering

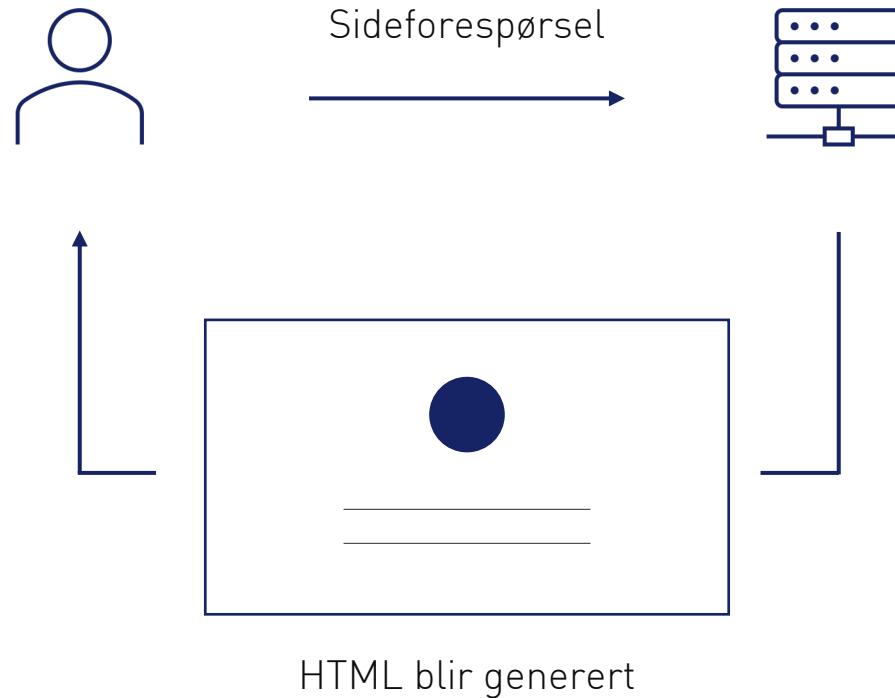


Server side rendering



Static generation

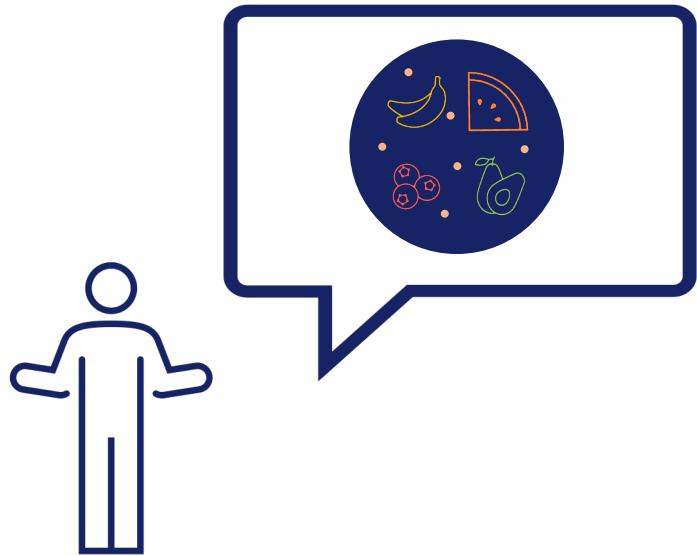
Server side rendering



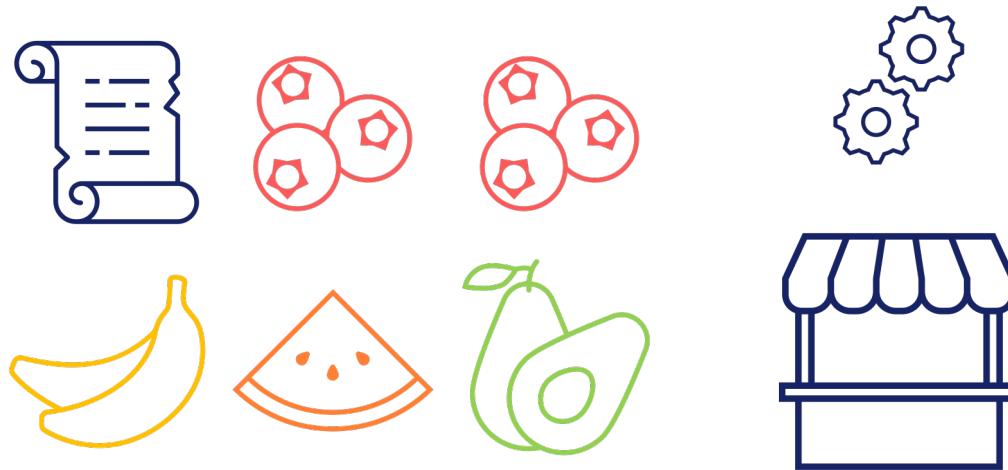
Server side rendering



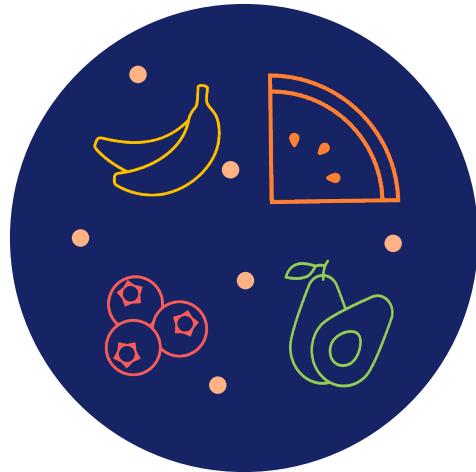
Server side rendering



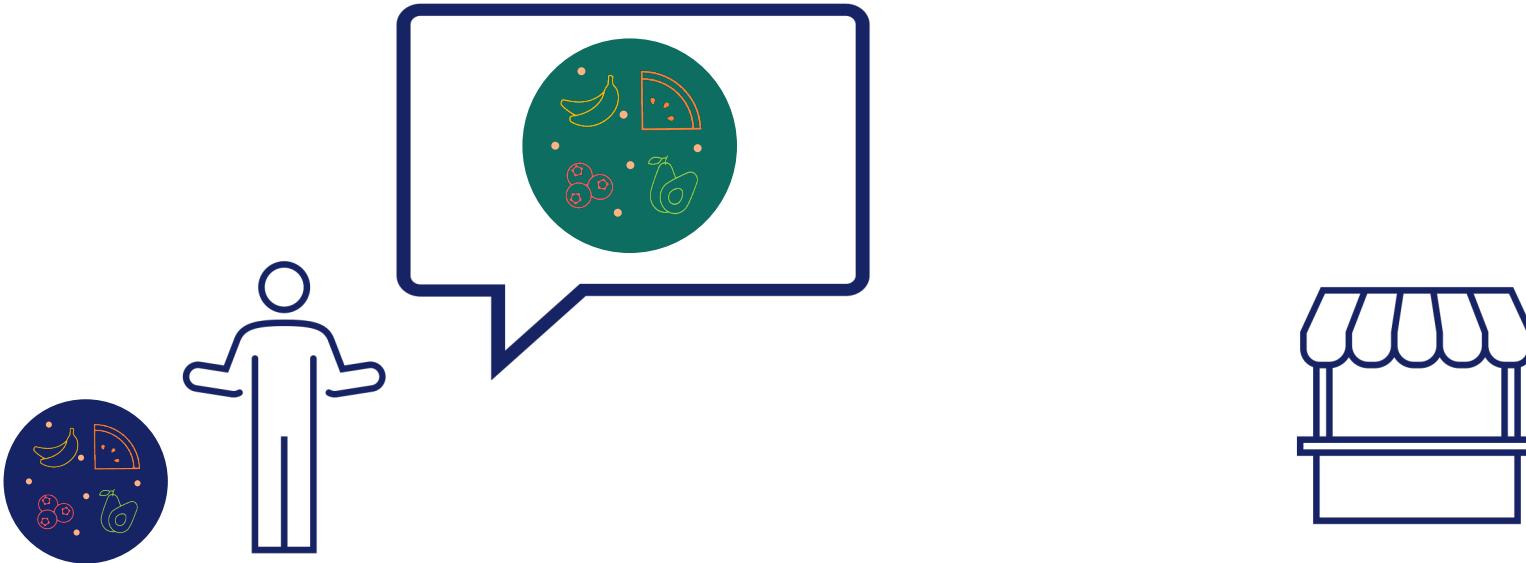
Server side rendering



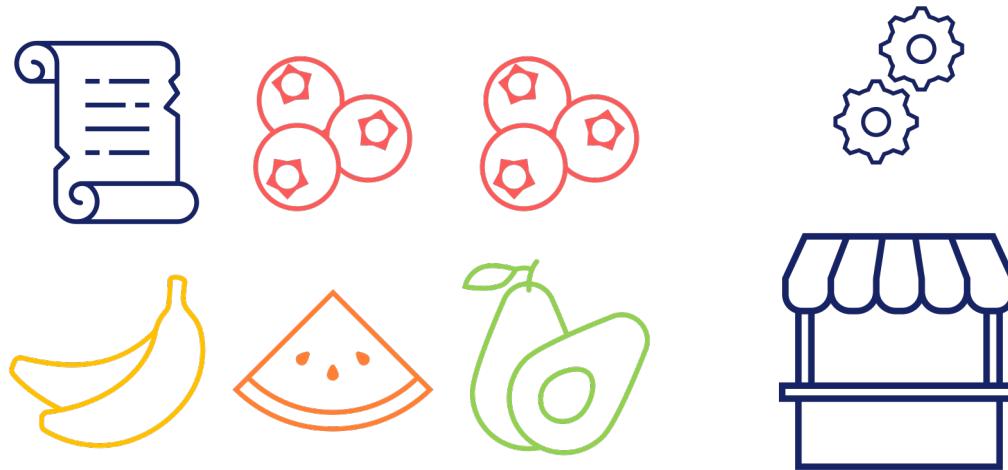
Server side rendering



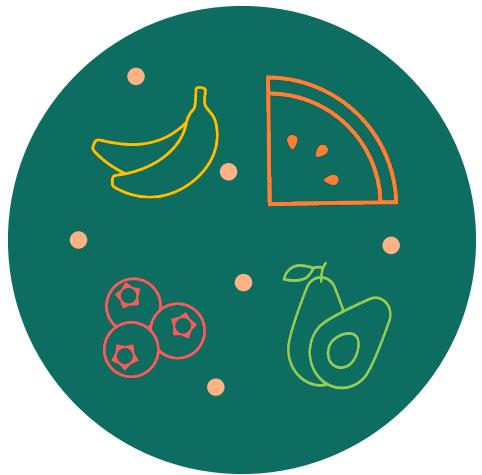
Server side rendering

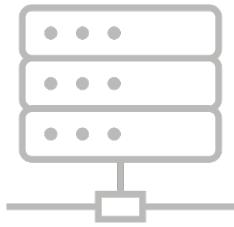


Server side rendering



Server side rendering





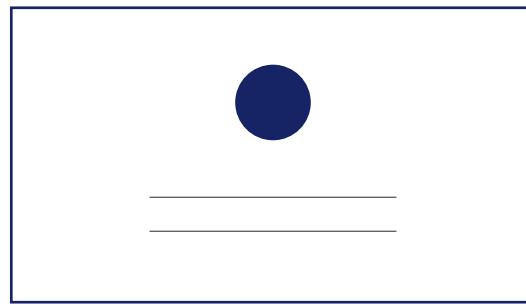
Server side rendering



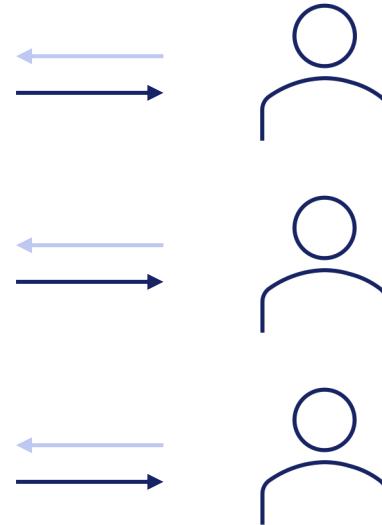
Static generation

Static generation

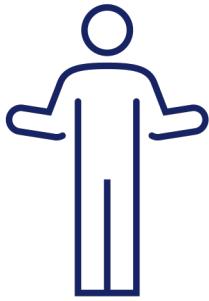
Bygging av applikasjon



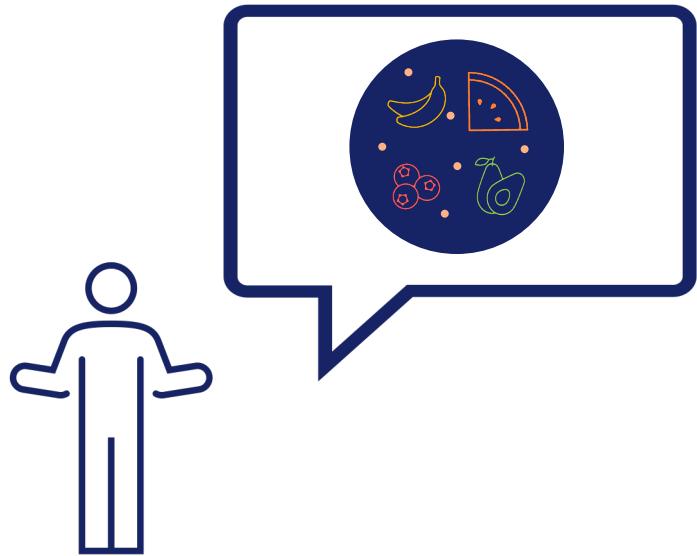
HTML blir generert



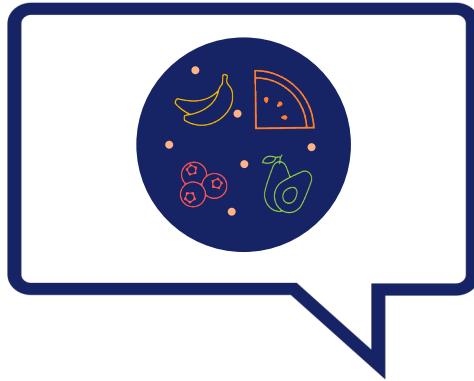
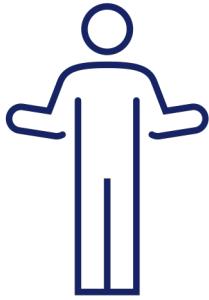
Static generation



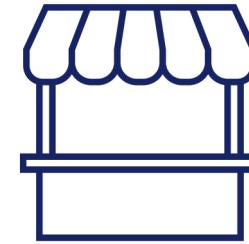
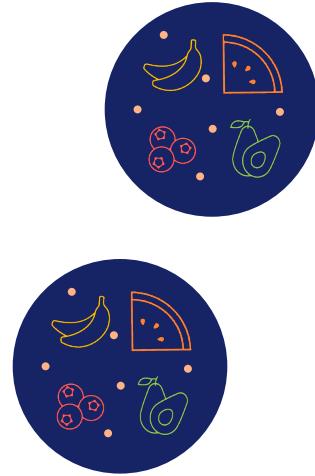
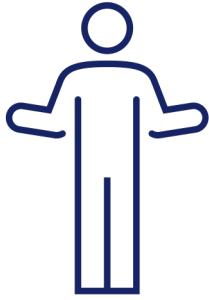
Static generation



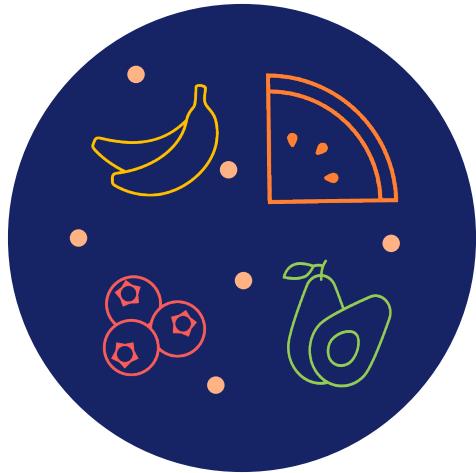
Static generation



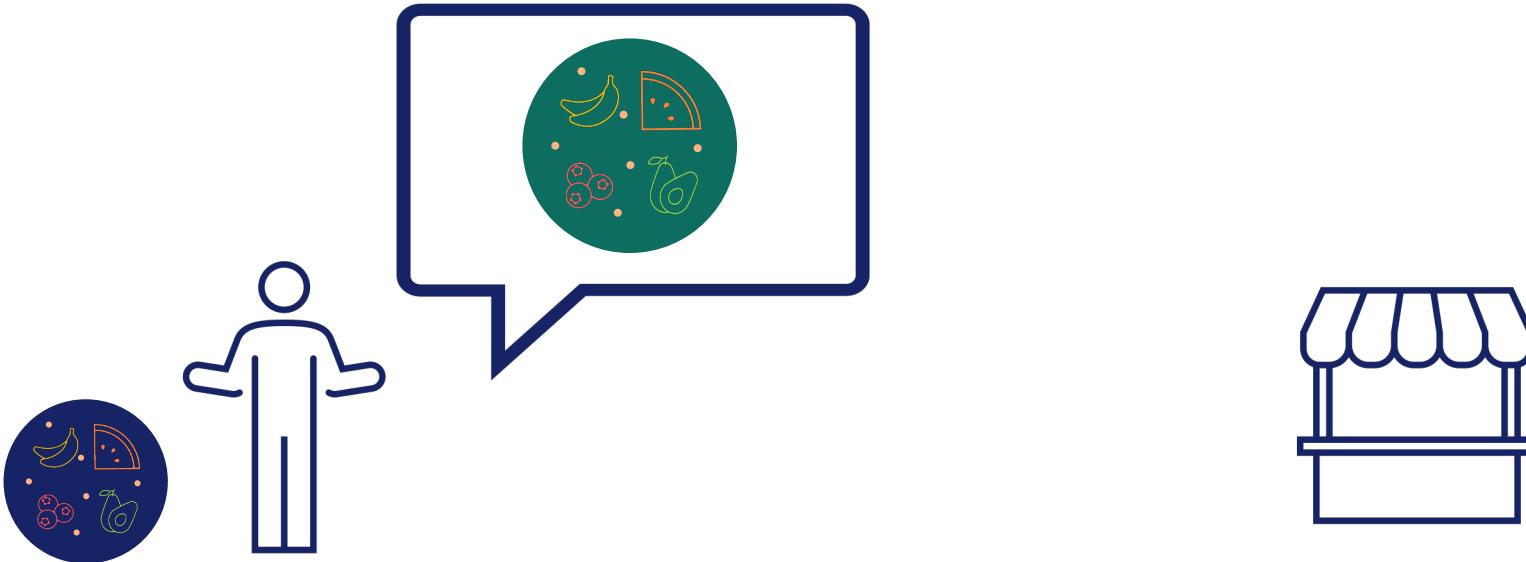
Static generation



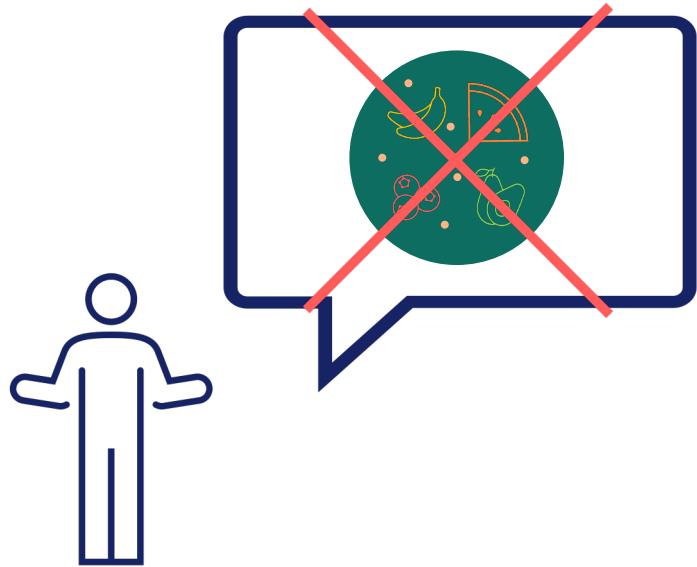
Static generation



Static generation



Static generation

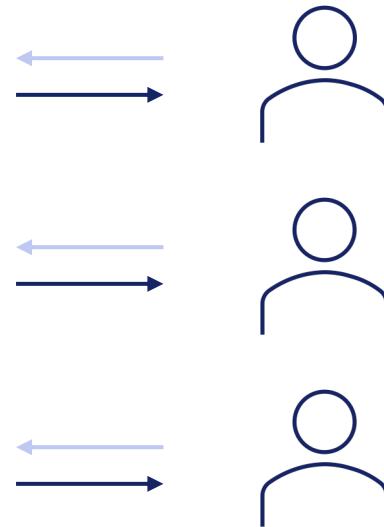
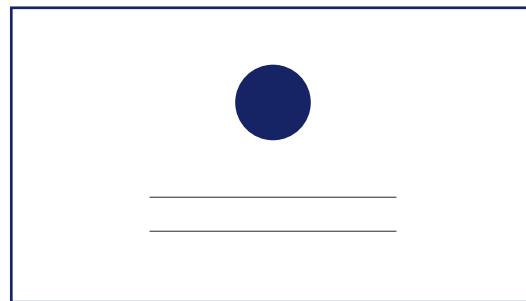


Static generation med data fetching

Bygging av applikasjon



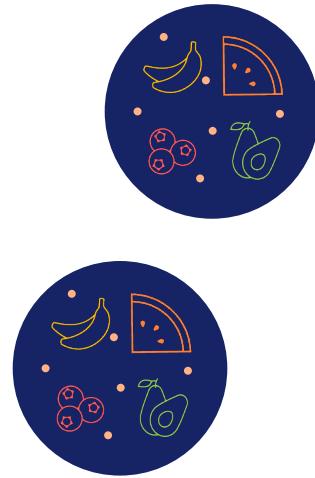
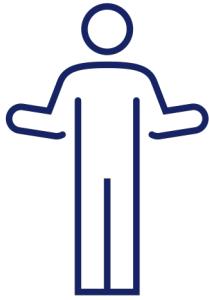
HTML blir generert



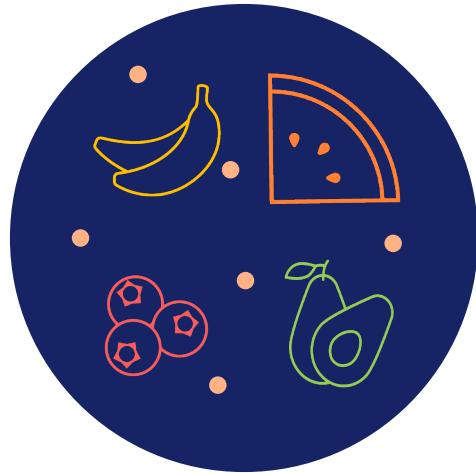
Static generation med data fetching

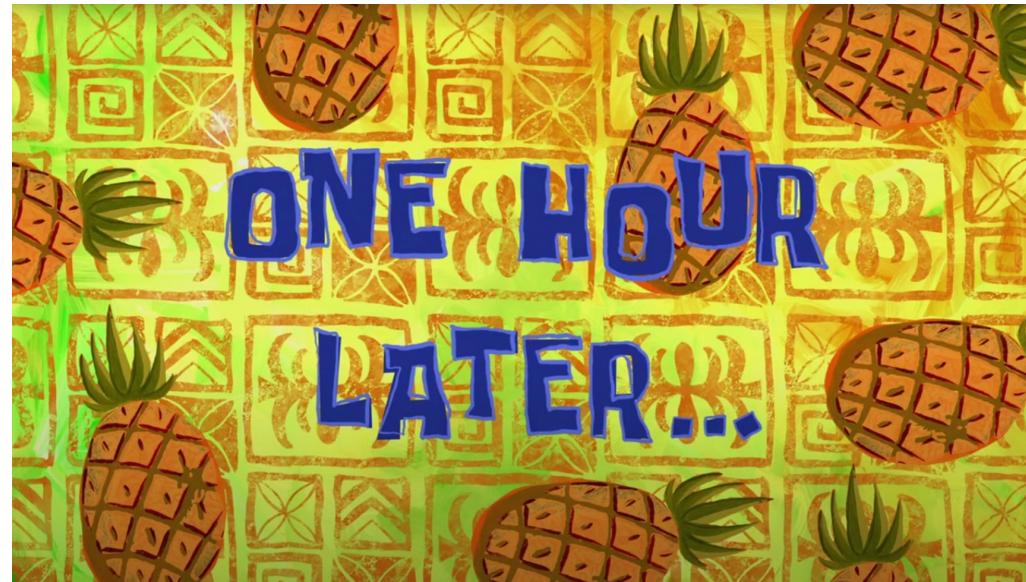


Static generation med data fetching



Static generation med data fetching

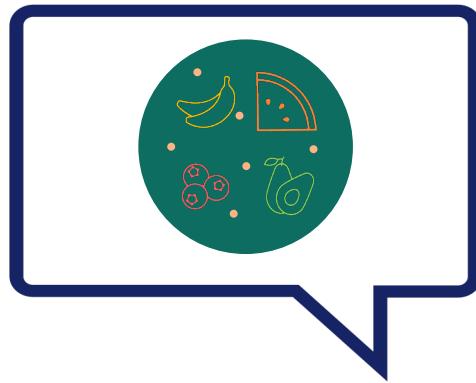
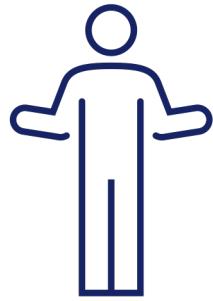




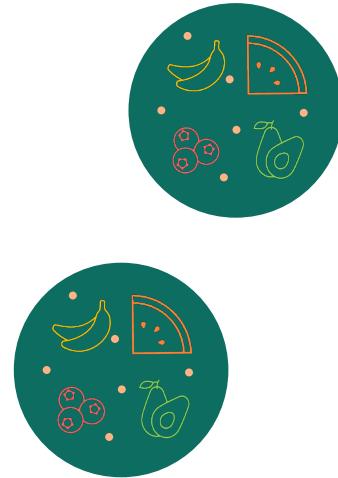
Static generation med data fetching



Static generation med data fetching



Static generation med data fetching

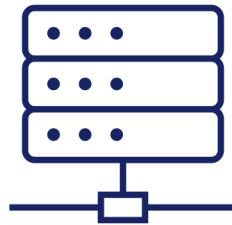


Static generation med data fetching





Client side rendering



Server side rendering



Static generation

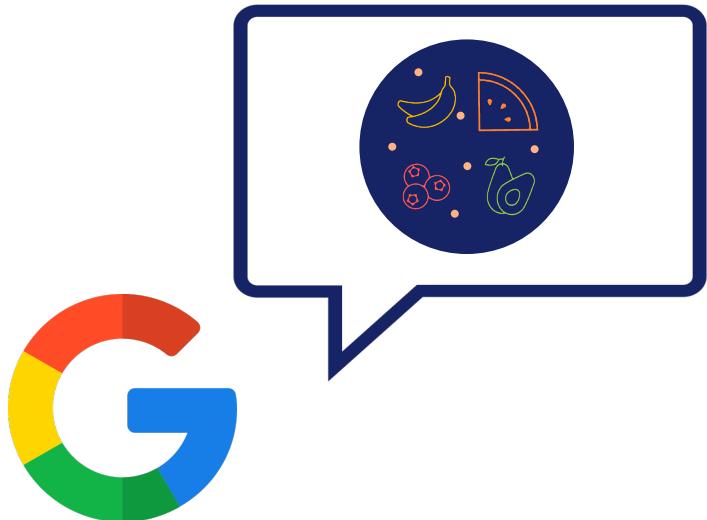
Pre-rendering

En liten notis om Search Engine Optimisation (SEO)

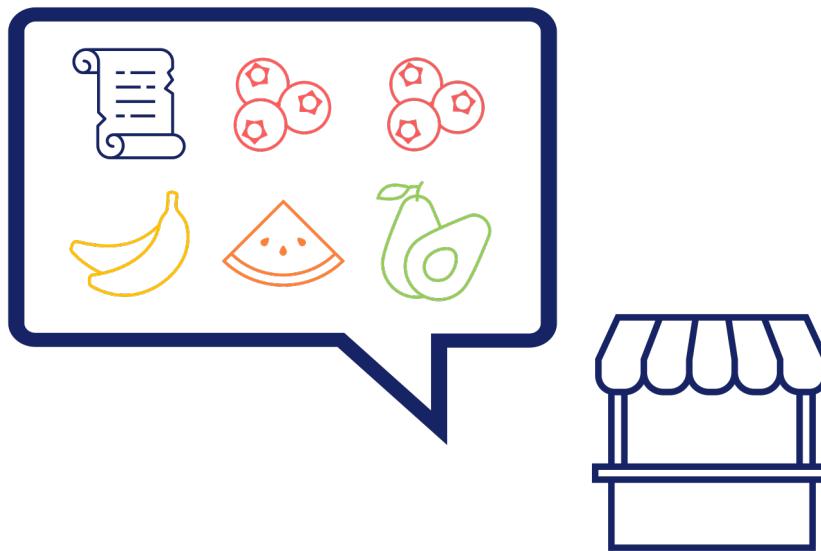
Client side rendering



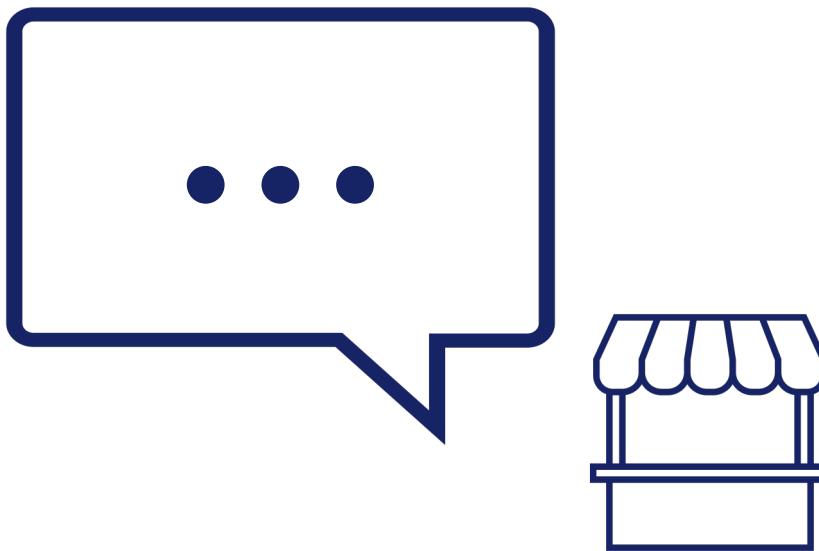
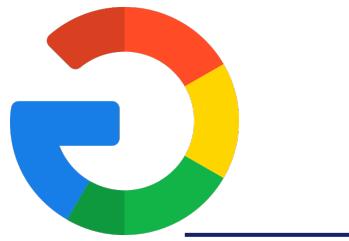
Client side rendering



Client side rendering



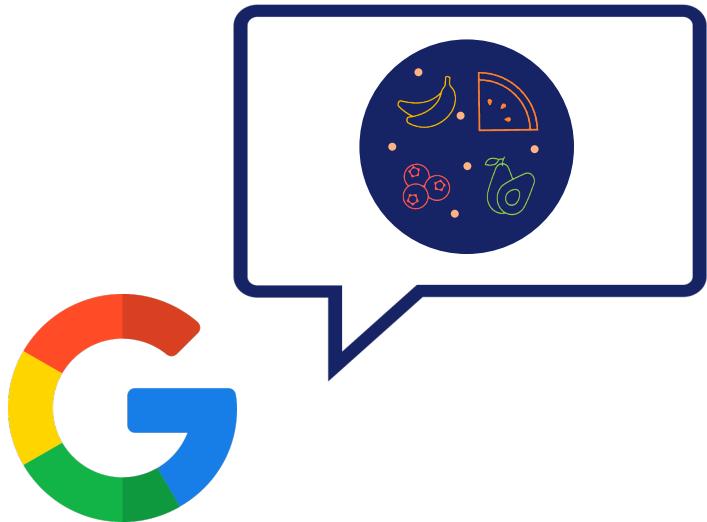
Client side rendering



Server side rendering



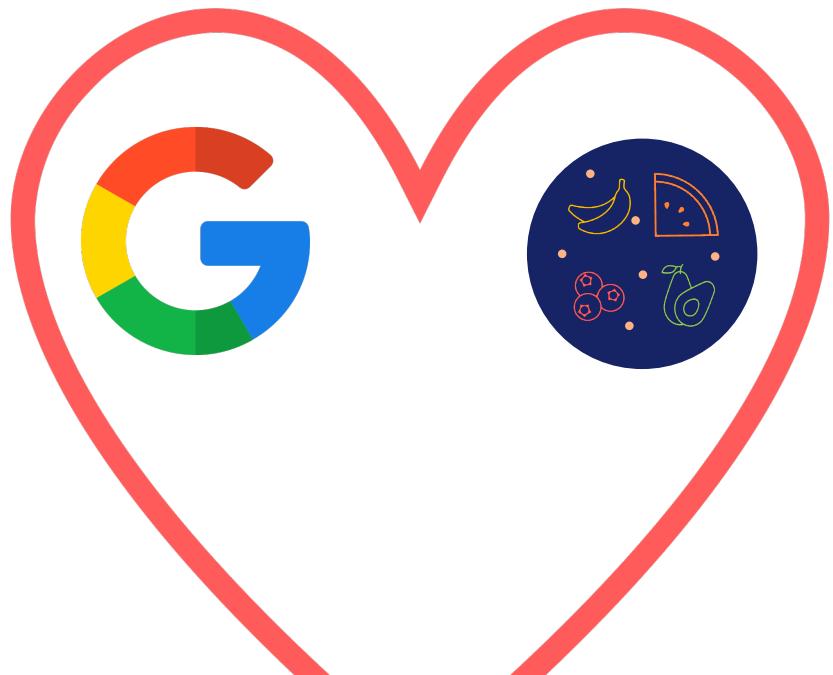
Server side rendering



Server side rendering



Server side rendering



Dette var mer riktig før!

Google gidder nå å kokkelere litt for å lage ferdig salaten

Pros

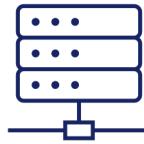


Client side rendering

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



Static generation

Statisk innhold

SEO er viktig

Konklusjon

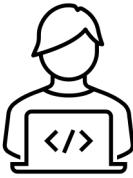
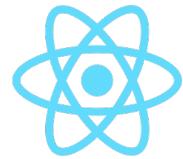
Det beste er nok en kombo av CSR, SSR og SSG!



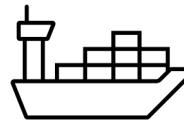
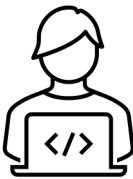
NEXT.JS

A large, bold, black 'X' is positioned over the word 'NEXT.JS'. The 'X' is oriented diagonally, with its top-left arm pointing towards the bottom-left and its top-right arm pointing towards the bottom-right. The word 'NEXT.JS' is written in a clean, sans-serif font, centered below the 'X'. The letters 'N', 'E', 'X', 'T' are all connected by a single horizontal stroke, and the '.JS' is placed to the right of the 'T'.

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

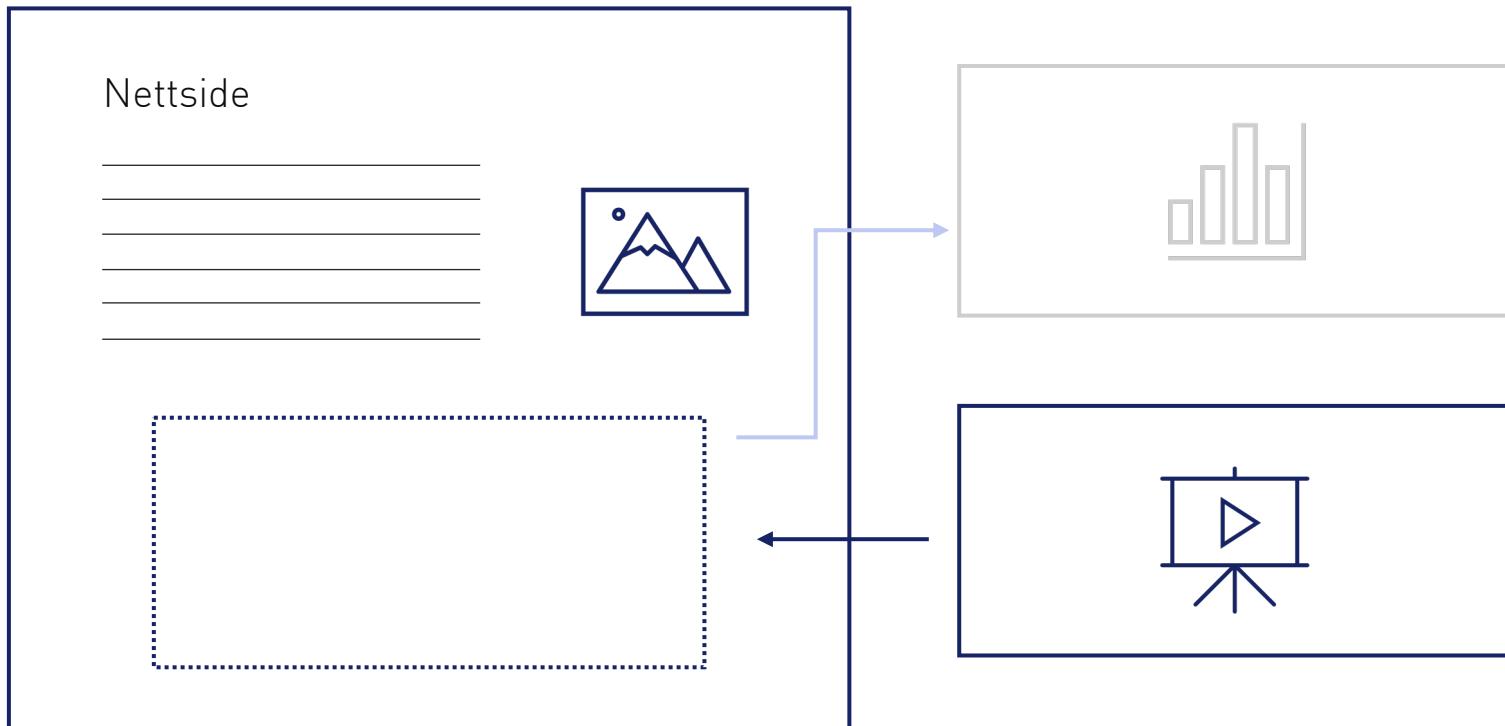
Code splitting i Next

Inkludert!

Filer under en route blir til bundles

Muliggjør «lazy-loading»

Partial rendering i Next.js



Node server i Next.js

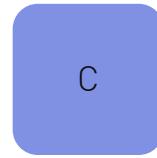
Rendering med



Server Components og Client Components

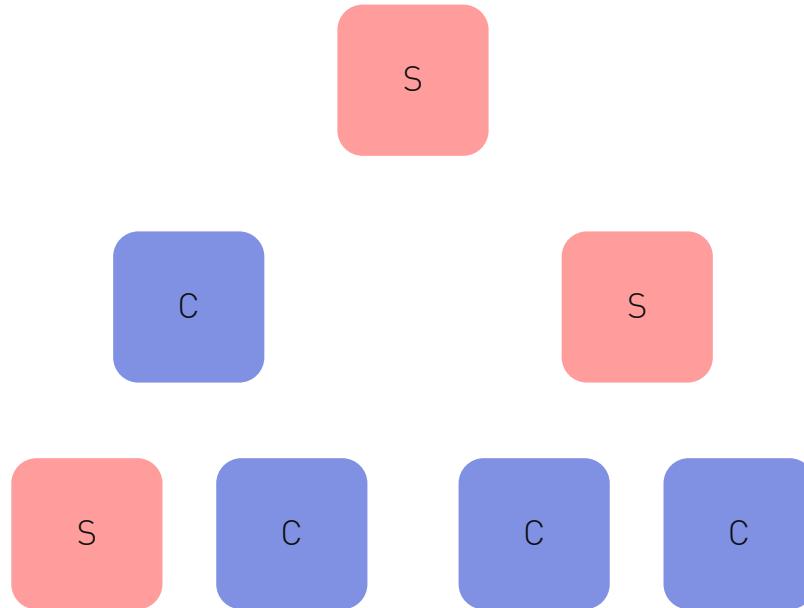


Server



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

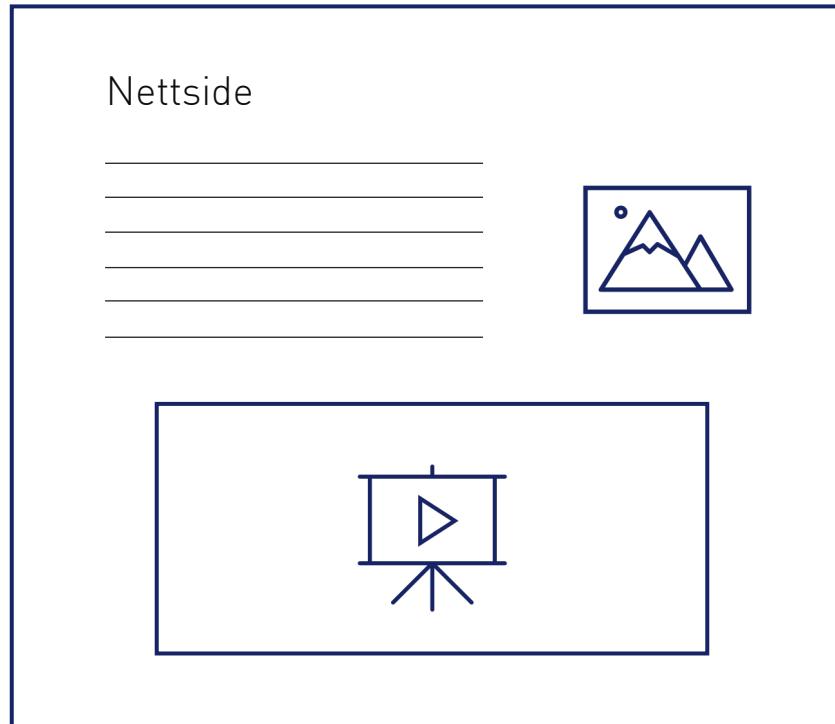
```
● ● ●

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

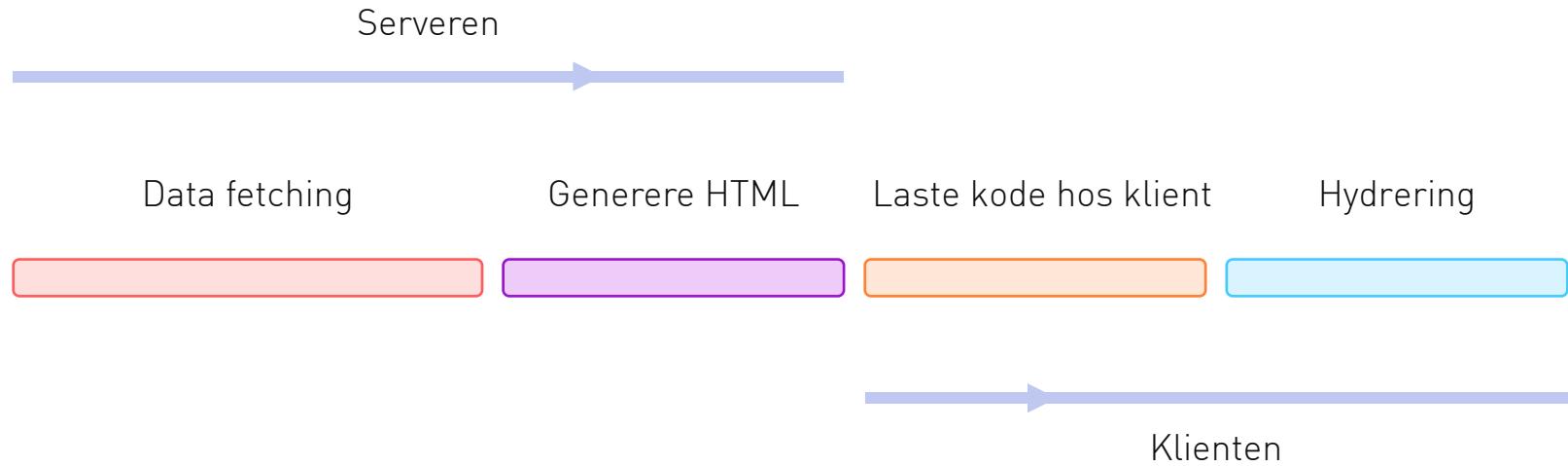
export default async function Home() {
  const dataRes = await fetch("http://localhost:8080/product.json");
  const data = (await dataRes.json()) as {
    navn: string;
    beskrivelse: string;
  };

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

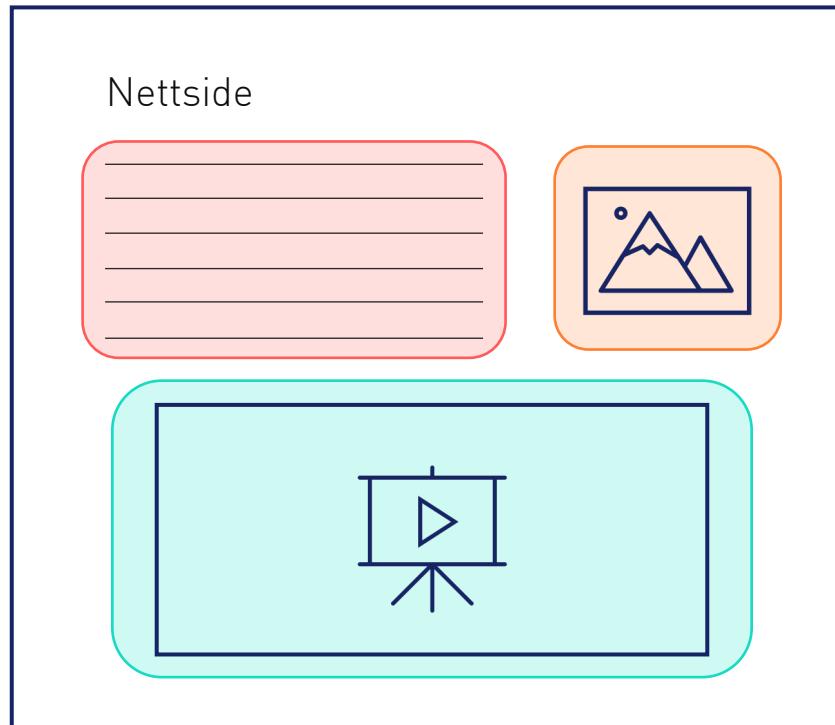
Streaming i Next.js



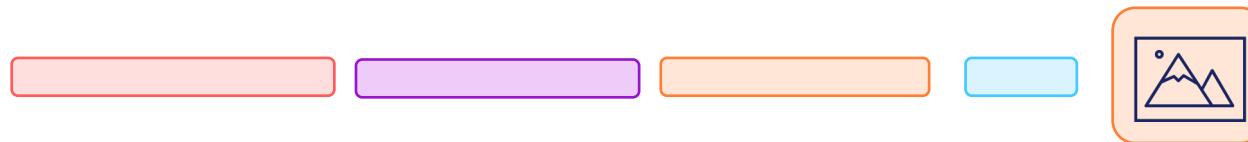
Streaming i Next.js



Streaming i Next.js



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 kjøpt 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 @acdlite · 12h

...

Replying to @acdlite

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 @acdlite · 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 @acdlite

What if you don't need ssr



10



1



33



15.5K



...



Andrew Clark @acdlite · 9h

Replying to @ken_wheeler

Same answer

...



Sebastian Markbåge @sebmarkbage · 10h

Replying to @sebmarkbage @riadbenguela 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 