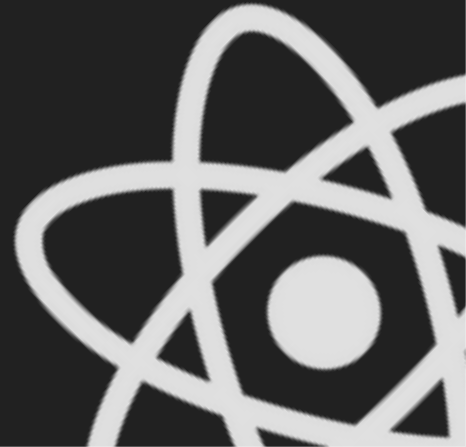


# NEXT .JS

## Getting started with Next.js



# **Ablauf**

**1**

**Kurzübersicht über den Workshop**

**2**

**Theoretischer Grundlagenteil**

**3**

**Praktischer Teil**

**4**

**Zusammenfassung**

# **Kurzübersicht über den Workshop**

# Überblick

- Mit dem Workshop ***“Getting Started with Next.js”*** wird ein erster Einblick in das Next.js Framework gegeben.
- Die Teilnehmer erhalten im Rahmen des Workshops Informationen über die Möglichkeiten, die das Framework bietet.
- Darüber hinaus werden grundlegende Kenntnisse für den praktischen Einsatz von Next.js vermittelt.

# Zielgruppe

Dieser Workshop wendet sich an Entwickler, die den Prozess der Entwicklung von Web-Anwendungen mit der React.js-Bibliothek schneller gestalten und das Rendering auf dem Server reibungslos implementieren möchten.

# Empfohlene Vorkenntnisse

JavaScript und die Grundlagen von React sollen beherrscht werden, einschließlich der folgenden Punkte:

- Erstellen von Komponenten
- JSX verwenden
- Übergabe von Props
- Verwendung von State

# **Theoretischer Grundlagenteil**

# Was ist **NEXT**.js ?

Next JS ist ein React-Webframework für die Frontend-Entwicklung, das von Vercel entwickelt wurde und Funktionalitäten bietet wie:

- Serverseitiges Rendering (SSR: Server Side Rendering)
- Statische Seitengenerierung (SSG: Static Site Generation)



# Was ist ~~NEXT~~.js ?

## ***Serverseitiges Rendering (SSR)***

- Serverseitiges Rendering (SSR) ist die Fähigkeit einer Anwendung, die Webseite auf dem Server anzuzeigen, anstatt sie im Browser zu rendern.
- Die Serverseite sendet eine vollständig gerenderte Seite an den Client; das JavaScript-Bündel des Clients übernimmt die Aufgabe und ermöglicht den Betrieb des SPA-Frameworks. (*SPA: Single Page Application*)

# Vorteile von NEXT.js

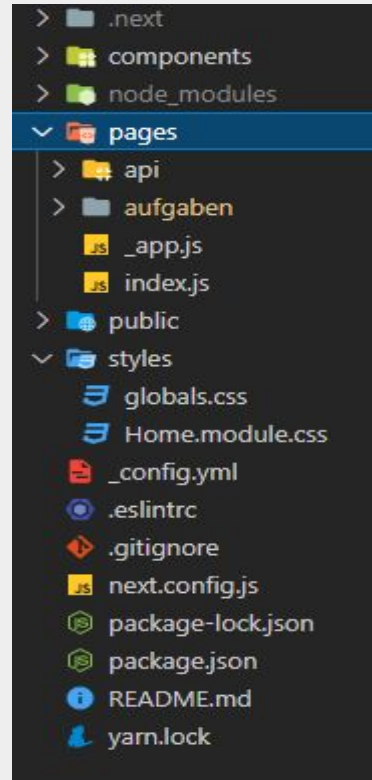
- Verbesserung der Leistung.
- Einfaches Page Routing.
- API-Routen
- Verwendung von TypeScript und Sass
- Einfache Bereitstellung

# **Praktischer Teil**

# Praktischer Teil

## Seiten in Next.js

- Eine Seite in Next.js ist eine React-Komponente, die aus einer Datei im Ordner **“pages”** exportiert wird.
- Seiten werden auf der Grundlage ihres Dateinamens mit einer Route verknüpft.
  - **pages/index.js** ist mit der Route **/** verknüpft.
  - **pages/aufgaben/aufgabe\_2/aufgabe2.js** ist mit der Route **/aufgaben/aufgabe\_2/aufgabe2** verknüpft.



# Praktischer Teil

## Links Komponenten

- Bei der Verlinkung zwischen Seiten auf Websites verwenden wir normalerweise den `<a>` HTML-Tag.
- In Next.js verwenden wir die Link-Komponente von **next/link**, um das `<a>`-Tag zu umhüllen. `<Link>` ermöglicht die clientseitige Navigation zu einer anderen Seite in der Anwendung.

```
import Link from 'next/link'

export default function FirstPost() {
  return (
    <>
      <h1>First Post</h1>
      <h2>
        <Link href="/">
          <a>Back to home</a>
        </Link>
      </h2>
    </>
  )
}
```

# Praktischer Teil

## Bild Komponenten

- **next/image** ist eine Erweiterung des HTML-Elements `<img>`, die für das moderne Web entwickelt wurde.
- Next.js unterstützt standardmäßig auch die Bildoptimierung. Dies ermöglicht die Größenanpassung, Optimierung und Bereitstellung von Bildern in modernen Formaten, wenn der Browser dies unterstützt. Dadurch wird vermieden, dass große Bilder an Geräte mit einem kleineren Viewport ausgeliefert werden. Außerdem kann Next.js automatisch zukünftige Bildformate übernehmen und sie an Browser ausliefern, die diese Formate unterstützen.

```
import Image from 'next/image'

const YourComponent = () => (
  <Image
    src="/images/profile.jpg" //
    height={144} // Desired size
    width={144} // Desired size v
    alt="Your Name"
  />
)
```

# Praktischer Teil

## CSS Styling

- Next.js-Seiten verwenden eine Bibliothek namens styled-jsx. Dabei handelt es sich um eine "CSS-in-JS"-Bibliothek, mit der Sie CSS innerhalb einer React-Komponente schreiben können, wobei die CSS-Stile skaliert werden (andere Komponenten sind davon nicht betroffen).
- Next.js hat integrierte Unterstützung für styled-jsx, aber Sie können auch andere beliebte CSS-in-JS-Bibliotheken wie styled-components oder emotion verwenden.

```
<div>
  <h3>Hier ist die Aufgabe 1</h3>
  <style jsx>{`
    h3 {
      font-size: 3rem;
      color: green;
    }
  `}</style>
</div>
```

# Praktischer Teil

## Layout Komponente

- Eine Layout-Komponente ist eine Komponente, die von allen Seiten gemeinsam genutzt wird.

```
export default function Layout({ children }) {  
  return <div>{children}</div>  
}
```

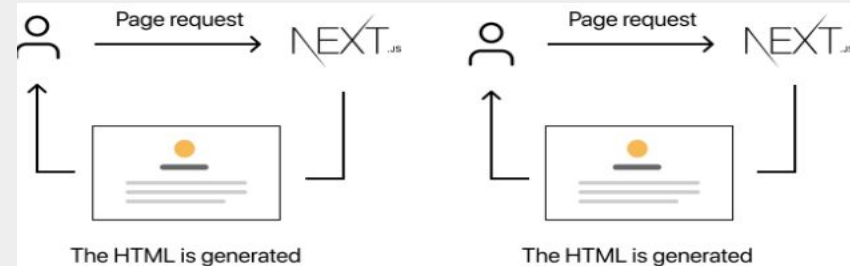
```
import Head from 'next/head'  
import Link from 'next/link'  
import Layout from '../components/layout'  
  
export default function FirstPost() {  
  return (  
    <Layout>  
      <Head>  
        <title>First Post</title>  
      </Head>  
      <h1>First Post</h1>  
      <h2>  
        <Link href="/">  
          <a>Back to home</a>  
        </Link>  
      </h2>  
    </Layout>  
  )  
}
```



# Praktischer Teil

## Arten von Pre-rendering

- Next.js verfügt über zwei Formen des Pre-Rendering: Statische Generierung und Serverseitiges Rendering. Der Unterschied besteht darin, wann das HTML für eine Seite generiert wird.
- **Statische Generierung** ist die Pre-Rendering-Methode, die das HTML zur Erstellungszeit generiert. Das vorgerenderte HTML wird dann bei jeder Anfrage wiederverwendet.
- **Serverseitiges Rendering** ist die Pre-Rendering-Methode, die den HTML-Code bei jeder Anfrage generiert.



# Praktischer Teil

## Data Fetching

- Es gibt drei Next.js-Funktionen, die zum Abrufen von Daten für das Pre-Rendering verwendet werden können:
  - ***getStaticProps (Statische Generierung)*** : Daten zum Zeitpunkt der Erstellung abrufen.
  - ***getStaticPaths (Statische Generierung)*** : Dynamische Routen für das Rendering von Seiten auf der Grundlage von Daten angeben.
  - ***getServerSideProps (Serverseitiges Rendering)*** : Daten bei jeder Anfrage abrufen.

# Praktischer Teil

## Dynamic Routing

- Bei komplexen Anwendungen reicht es nicht immer aus, Routen mit Hilfe vordefinierter Pfade zu definieren.
- In Next.js können Klammern zu einer Seite hinzugefügt werden ([param]), um eine dynamische Route zu erstellen.
- 
- Nehmen wir die folgende Seite ***pages/post/[pid].js***:
  - Jede Route wie `"/post/1"`, `"/post/abc"` usw. wird mit `"pages/post/[pid].js"` abgeglichen.
  - Der übereinstimmende Pfadparameter wird als Abfrageparameter an die Seite gesendet und mit den anderen Abfrageparametern zusammengeführt.

# Praktischer Teil

## Aufgaben:

[https://ramichr.github.io/WT\\_Workshop-Getting\\_started\\_with\\_Next.js/](https://ramichr.github.io/WT_Workshop-Getting_started_with_Next.js/)

# **Zusammenfassung**

# Zusammenfassung

Daraus können wir ersehen, dass Next.js eine verbesserte Version von React ist, nicht nur, weil es mehr Vorteile als React hat, sondern auch wegen seiner einfachen Anwendung.