npm, npx, yarn 이해하기

NPM (Node Package Manager): 자바스크립트 패키지 관리 도구로, 패키지 설치, 버전 관리, 의존성 관리를 수행했음.

NPX (Node Package Execute): NPM과 함께 제공되는 도구로, 패키지를 전역 설치하지 않고 실행할 수 있도록 해줬음.

Yarn: 페이스북에서 개발한 패키지 매니저로, 병렬 설치, 더 나은 캐싱 및 보안 기능을 제공하여 빠르고 효율적인 패키지 관리를 지원했음.

리액트 초기 설치

프로젝트 생성 방법: create-react-app명령어를 사용하여 리액트 프로젝트를 간편하게 생성했음.

VSCode로 설치하기

비주얼 스튜디오 코드로 리액트 생성하기: VSCode에서 터미널을 열고 create-react-app을 통해 프로젝트를 생성했음.

리액트 사전지식

리액트를 왜 사용할까? 리액트는 컴포넌트 기반 아키텍처를 제공하여 재사용 가능하고 관리하기 쉬운 UI를 구축할 수 있었음.

바벨(Babel)과 웹팩(Webpack): 바벨은 최신 자바스크립트 코드를 구형 브라우저에서도 호환되도록 변환하는 트랜스파일러였음. 웹팩은 모듈 번들러로, 여러 자바스크립트 파일을 하나로 묶어 효율적으로 관리했음.

확장자 이해하기: 리액트 파일은 주로 .js, .jsx, .ts, .tsx확장자를 사용했음. JSX는 자바스크립트 XML의 줄임말로, 자바스크립트 안에서 HTML을 작성할 수 있게 해줬음.

컴포넌트

클래스 컴포넌트와 함수형 컴포넌트: 클래스 컴포넌트는 리액트의 기존 방식으로, 상태와 생명주기 메서드를 갖췄음. 함수형 컴포넌트는 리액트 훅을 사용하여 상태와 생명주기 기능을 구현할 수 있었으며, 더 간결하고 이해하기 쉬웠음.

컴포넌트를 작성하는 방법: 컴포넌트는 자바스크립트 함수 또는 클래스로 작성되며, JSX를 반환했음.

컴포넌트 CSS 스타일링: CSS 파일을 별도로 만들어 컴포넌트에 스타일을 적용할 수 있었음. CSS 모듈, styled-components 등의 기법을 사용하여 컴포넌트별로 고유한 스타일을 적용할 수 있었음.

테일윈드 추가 설정: Tailwind CSS는 유틸리티 퍼스트 CSS 프레임워크로, 빠르고 효율적인 스타일링을 가능하게 했음.

폰트 적용 방법: 웹폰트를 사용하거나 로컬 폰트를 불러와서 스타일을 적용할 수 있었음.

Props: 부모 컴포넌트에서 자식 컴포넌트로 데이터를 전달하는 방법임. Props는 읽기 전용이며, 자식 컴포넌트에서 수정할 수 없었음.

Children: 컴포넌트의 자식 요소를 다루는 방법을 설명했음. 이는 컴포넌트 내부에서 다른 컴포넌트를 포함하는 데 유용했음.

조건부 렌더링: 조건에 따라 컴포넌트를 렌더링할 수 있었음.

반복 렌더링: 배열의 map 메서드를 사용하여 컴포넌트를 반복 렌더링할 수 있었음.

이미지 렌더링: 이미지 파일을 import하여 사용하거나, URL을 통해 이미지를 렌더링할 수 있었음.

로그인 컴포넌트 만들기: 간단한 로그인 컴포넌트를 구현하여 실습했음.

리액트 훅

useState: 함수형 컴포넌트에서 상태를 관리할 수 있게 했음.

useId: 유니크한 ID를 생성할 수 있게 했음.

useRef: DOM 요소에 접근하거나, 컴포넌트의 상태를 저장할 수 있게 했음.

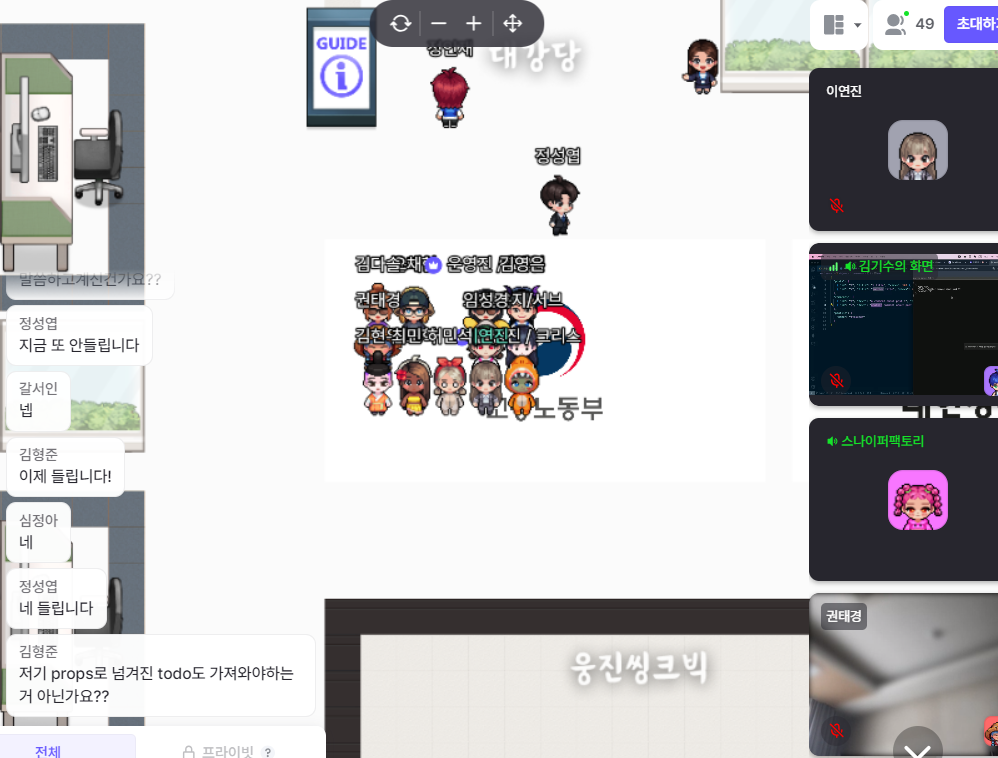
useCallback: 함수형 컴포넌트에서 함수를 메모이제이션할 수 있게 했음.

useMemo: 계산된 값을 메모이제이션할 수 있게 했음.

React.memo: 컴포넌트를 메모이제이션하여 불필요한 렌더링을 방지할 수 있게 했음.

useReducer: 복잡한 상태 관리를 위해 리듀서를 사용할 수 있게 했음.

Todo List 예제: useReducer와 메모이제이션 훅을 사용하여 Todo List를 구현 실습 진행함.



\* 대강당에 각자 의자에 맞춰서 있는거도 귀엽지만 와글와글 모인거 꽤나 귀엽다

리액트의 기본 개념과 컴포넌트 작성 방법을 학습했음. 리액트 훅을 사용하여 함수형 컴포넌트에서 상태를 관리하는 방법도 익혔음. 컴포넌트의 스타일링과 이벤트 처리 방법에 대해서도 학습했음.

리액트 훅의 사용법이 처음에는 다소 생소하게 느껴졌음. 예제를 통해 반복 학습하면서 점차 이해할 수 있었으나, 여전히 어려움이 있었음. 스타일링에서 CSS 모듈과 styled-components의 사용법이 혼란스러웠음. 여러 문서를 참고하면서 문제를 해결할 수 있었지만, 여전히 익숙하지 않음.

리액트 훅을 더 깊이 있게 학습하여 복잡한 상태 관리와 최적화 기법을 익혀야겠다고 느꼈음. styled-components를 활용한 스타일링을 더 연습하여 컴포넌트별로 고유한 스타일을 적용할 수 있도록 해야겠음. 추가적으로, 공식 문서와 다양한 리소스를 통해 리액트의 여러 기능을 체계적으로 학습할 필요가 있음.

리액트의 기본 개념과 컴포넌트 작성 방법을 익히는 데 어려움이 있었음. 실습을 성공적으로 마치지 못한 부분도 있어 아쉬움이 남음. 수업 외로 교재와 강의 등을 통해 리액트를 더 깊이 있게 학습할 필요성을 느낌. 실습한 내용을 복습하고, TODO 리스트를 다시 구현해보면서 부족한 부분을 보완할 계획임. 앞으로는 꾸준히 학습하여 리액트에 대한 이해를 높이고, 다양한 프로젝트에 적용해볼 목표를 가짐.



#유데미 #udemy #웅진씽크빅 #스나이퍼팩토리 #인사이드아웃 #미래내일일경험 #프로젝트캠프 #부트캠프 #Next.js #프론트엔드개발자양성과정 #개발자교육과정