원티드 프론트엔드 챌린지
8월 1회차

20:00에 시작하겠습니다!

링크

출석체크 : https://forms.gle/4nxh2zfjppb7fe7eA

강사소개

- (전) 의료 인공지능 스타트업 테크리드
- 원티드 챌린지 3회 진행
- 내일 이직(!)

챌린지 내용

- 간단한 배달앱을 활용해서 경험하는 아래 항목들
  - 배달앱이 특성상 값을 넘겨줘야 할 것들이 많아서 선택함
- 변수관리 방법
  - 로컬변수와 전역변수
  - 렌더링 효율을 개선하는 변수관리법
- 테스트코드
  - 테스트 범위
  - 테스트 종류
  - 요즘 많이 하는 방식?
- 최적화
  - React <-> JavaScript
  - Volunteer?

아하모멘트

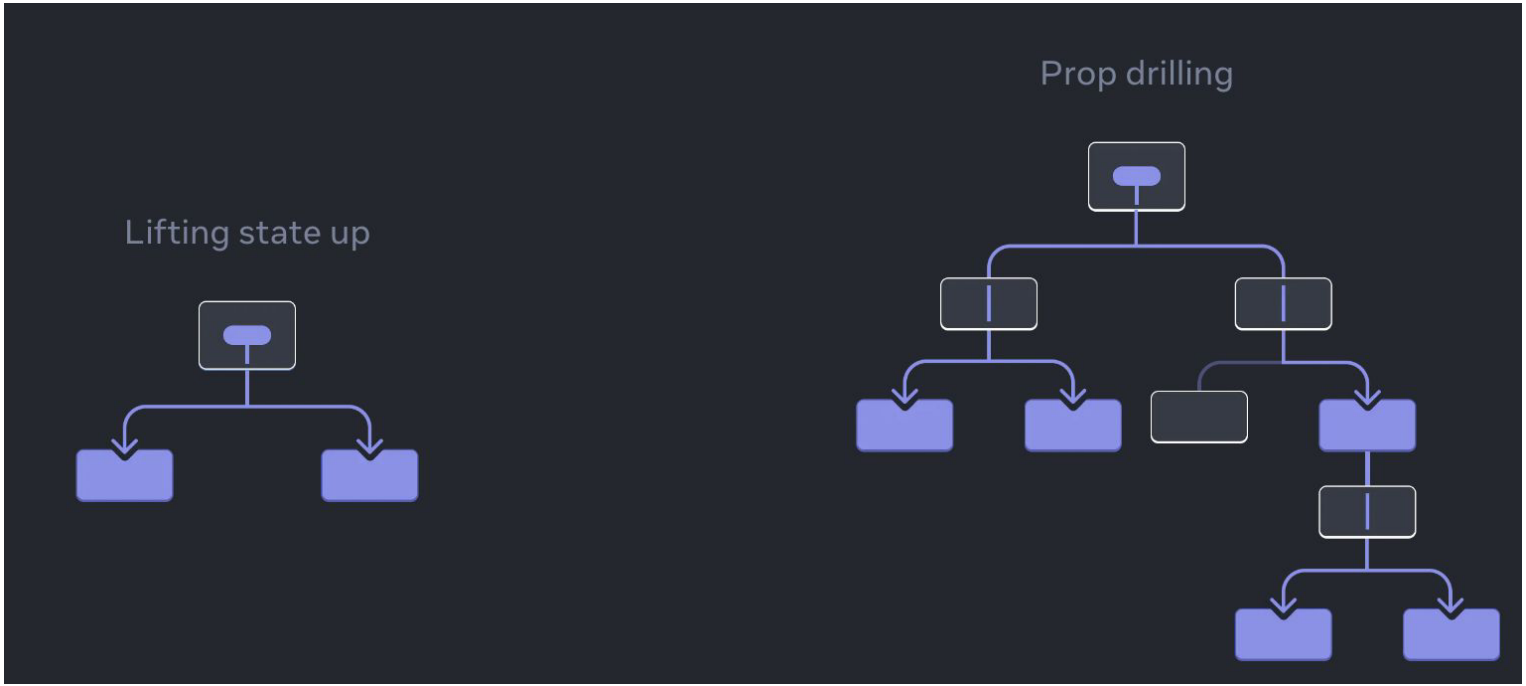
- [아하!모먼트] 최근에 배운 새로운 기술이나 도구가 있나요?
  - 최근 기술 알아보기
- [아하!모먼트] 요즘 관심있는 분야는 어떤것인가요?
  - 기술면접 대비방법
- [아하!모먼트] 개발을 공부하는 나만의 방법이 있나요?
  - 성장하기 위한 노력
- [아하!모먼트] 이 회사에 지원한 이유가 무엇인가요?
  - 회사 선택 기준
- 각종 질의응답
  - Google Form 활용
  - 그동안 진행한 챌린지 내용. 또는 기타 등등 아무거나
  - https://forms.gle/d76uG2mwLb3rEG349

지역변수 vs 전역변수

1. state를 어떻게 관리할까?
    a. 리액트에 한정짓지 않고 프로그래밍 관점에서 비교한다면?
        i. Scope
        ii. Lifespan
        iii. Namepsace collision
    b. var vs let vs const
2. 해당 변수를 어디에서 사용하는가?
3. 전역변수 설정하는 기준?

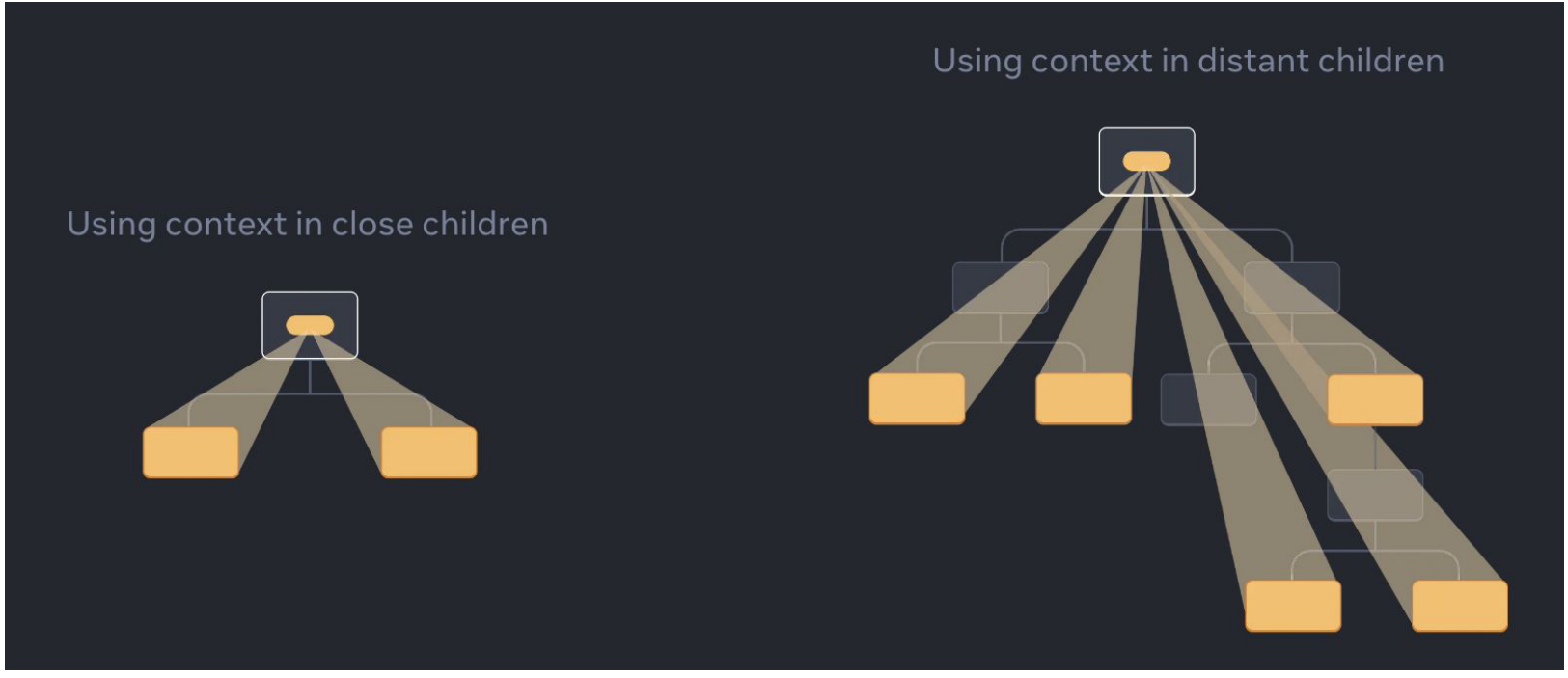
전역 상태 관리의 필요성

1. Container - Presenter 방식



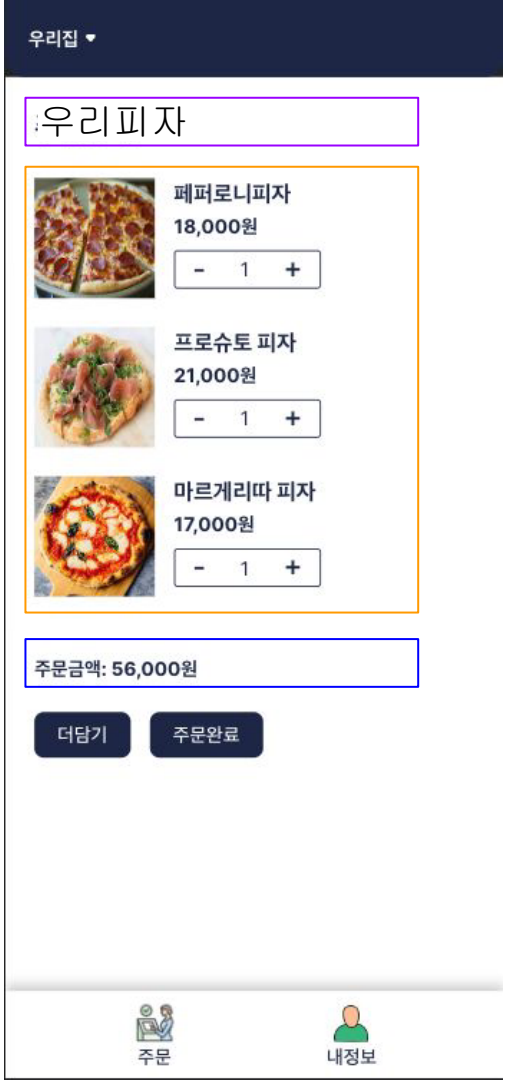
전역 상태 관리의 필요성

1. Flux && Redux



전역변수 툴 비교 - Context API

1. 특징
    a. `State` 를 `Provide` 하는 방식
    b. React 내장 기능이라 별도 라이브러리 설치가 필요하지 않음
    c. HTTP Request도 Context에서 모두 관리
        i. `context`라는 단어에 맞게 해당 관심사와 관련된 모든 것을 context가 처리
2. 단점
    a. 비지니스 로직이 복잡해지는 경우 Provider 다수 생성 필요
    b. Provider 1개의 경우에도 설정할 것들이 많음



1. OrderContext
    a. restaurantName
    b. newOrder
    c. totalPrice

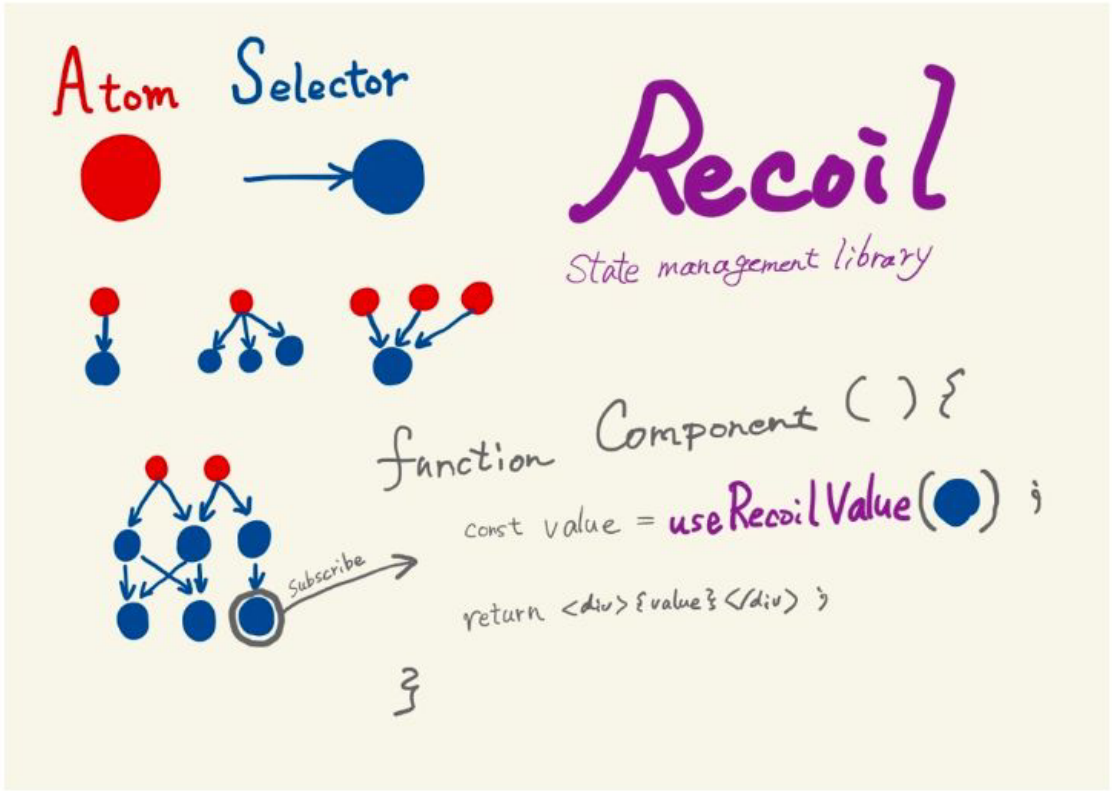
전역변수 툴 비교 - Redux

1. 특징
    a. FLUX 패턴을 적용함
    b. Store 하나에서 변수를 관리하기 때문에 여러 Provider를 선언하지 않아도 됨
    c. Thunk, saga 등의 Middleware를 통한 비동기처리
2. 단점
    a. 설정할 것이 엄청 많음
    b. 개인적으로는 Context API 2.0이 아닌가 생각…
    c. 요즘 잘 안씀

전역변수 툴 비교 - Recoil

1. atoms and selectors라는 개념
    a. 내가 필요한 값만 `subscribe`하는 느낌
    b. atom은 일반적인 state와 유사한 개념
    c. selector는 atom을 Manipulate 해야하는 경우 사용됨
2. 장점
    a. 구조가 간단해서 적용하기 쉬움
    b. ContextAPI의 rendering 비효율을 개선함
        i. Concurrent mode 지원
3. 단점
    a. atom의 수가 엄청나게 증가할 수도 있음

전역변수 툴 비교 - Recoil



1. Context API의 렌더링 문제를 해결함

Server State - React Query

1. 비즈니스 로직들을 대부분 백엔드에서 관리
    a. 많은 변수들이 서버에서 불러온 값을 관리하기 위해 사용
2. 서버에서 불러온 값들을 클라이언트가 관리하지 말자
3. 코드를 다시 보면…

React Query 최적화

1. Cache된 데이터의 상태
    a. Fresh
    b. Stale
    c. Inactive
2. staleTime vs cacheTime
    a. useRestaurantList를 통해서 확인

전역변수를 사용하지 않을 수는 없을까?

1. Toss SLASH 23
    a. 페이지 routing을 하지 않고 component로 관리
    b. 관련성 있는 코드의 “응집도"를 높이는 방식

전역변수를 사용하지 않을 수는 없을까?

1. Toss SLASH 23
   a. 페이지 routing을 하지 않고 component로 관리
   b. 관련성 있는 코드의 “응집도"를 높이는 방식
2. 항상 이런 방식을 택할수는 없음
   a. 개발팀의 컨벤션 중요함
   b. 컨벤션을 지키는 코드가 가독성에 유리

Recap

1. 지역변수 vs 전역변수
2. Context API vs Recoil
3. React-Query의 장점

[아하!모먼트] 최근에 배운 새로운 기술이나 도구가 있나요?

1. 새로운(?) 기술들이 쏟아지고 있음
    a. Next.js, VITE, ChatGPT
2. 정보를 습득하기 위한 노력
    a. 요즘IT
    b. 코드너리
    c. 커리어리
    d. 프론트엔드 번역해주시는 분들 구독
3. ChatGPT 사용법