

리액트로 프론트엔드 레벨업

Alex Kwon

본격 컴포넌트 다루기 9/9



- 실습: Pokedex 만들기
- Component lifecycle
- API 데이터 호출하기
- Pure Component
- Styling
- Proptypes

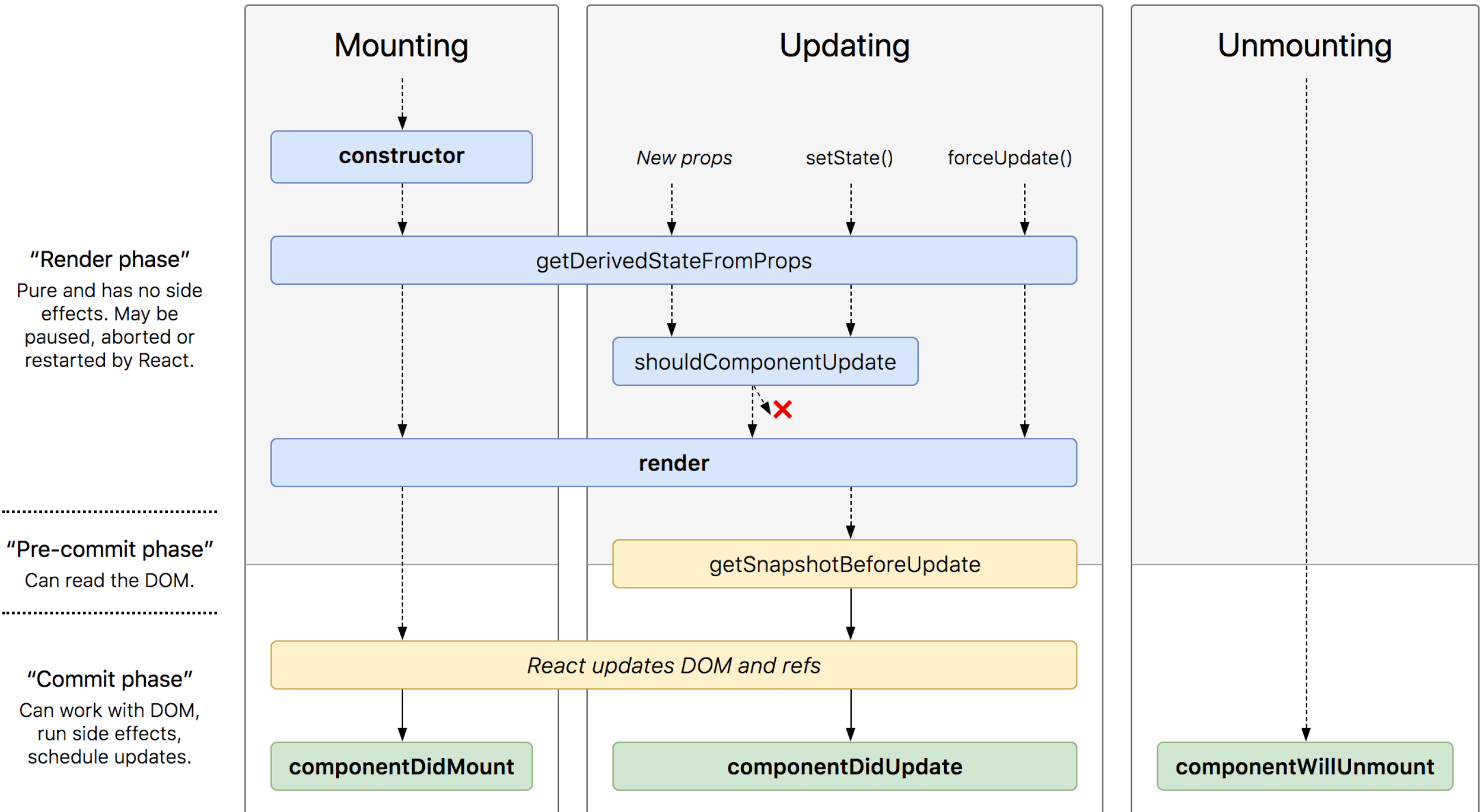
실습: Pokedex 만들기

- <https://pokeapi.co/>
- CRA로 새 프로젝트 구성
- 디렉토리 정리
- <App/>, <Main/>, <Info/> 컴포넌트 생성

Component Lifecycle

- 컴포넌트의 props 또는 state가 변할 때 작동
- 각 시점별 로직을 작성하기 위함
- 종류
 - `componentDidMount()`
 - `componentDidUpdate()`
 - `componentWillUnmount()`

Component Lifecycle 순서



API 데이터 호출하기

- REST API
- JSON 형식의 데이터를 호출
- fetch, axios, \$.ajax, superagent, ...

API 데이터 호출하기 - JSON 데이터

```
▼ {code: 0, description: "정상 처리 되었습니다.", hasNext: false,...}
  code: 0
  count: 18
  ▼ data: [{id: "_popular", label: "인기", comicCount: 1}, {id: "romance", label: "로맨스", comicCount: 113},...]
    ▶ 0: {id: "_popular", label: "인기", comicCount: 1}
    ▼ 1: {id: "romance", label: "로맨스", comicCount: 113}
      comicCount: 113
      id: "romance"
      label: "로맨스"
    ▶ 2: {id: "bl", label: "BL", comicCount: 30}
    ▶ 3: {id: "drama", label: "드라마", comicCount: 118}
    ▶ 4: {id: "fantasy", label: "판타지", comicCount: 91}
    ▶ 5: {id: "gag", label: "개그", comicCount: 78}
    ▶ 6: {id: "action", label: "액션", comicCount: 53}
    ▶ 7: {id: "school", label: "학원", comicCount: 82}
    ▶ 8: {id: "mystery", label: "미스터리", comicCount: 50}
    ▶ 9: {id: "horror", label: "호러", comicCount: 8}
    ▶ 10: {id: "food", label: "음식", comicCount: 4}
    ▶ 11: {id: "day", label: "일상", comicCount: 73}
    ▶ 12: {id: "sf", label: "SF", comicCount: 26}
    ▶ 13: {id: "martial", label: "무협", comicCount: 2}
    ▶ 14: {id: "gl", label: "백합", comicCount: 21}
    ▶ 15: {id: "historical", label: "시대극", comicCount: 17}
    ▶ 16: {id: "sports", label: "스포츠", comicCount: 16}
    ▶ 17: {id: "gore", label: "스릴러", comicCount: 5}
  description: "정상 처리 되었습니다."
  hasNext: false
```

- JavaScript Object Notation - JS 객체 형식
- 트리 구조

Pure Component

- functional 컴포넌트 => 입력 & 출력
- state 및 lifecycle method가 없음

- css, sass/scss, less, ...
- styled-components

- 컴포넌트 별로 props의 type을 정의
- 프로젝트가 커질 때, 협업 시 유용. 문서화, 에러방지.
- typescript 로 대체 가능!



다음에
또!
같이!
만나요!