

# Context API 짚 하기





## 문제] 다음의 조건에 만족하도록 React를 작성 하시오.

### 조건

- ① Context API를 이용하여 전역 상태로 찜 목록 관리하기
- ② WishlistContext.jsx 파일을 생성하여 전역 상태를 관리하도록 작성한다.
- ③ Context 안에는 다음 기능을 포함한다:
  - wishlist: 현재 찜 목록 배열 상태
  - addToWishlist(item): 찜 추가 함수
  - removeFromWishlist(id): 찜 제거 함수
  - isInWishlist(id): 이미 찜된 항목인지 확인하는 함수
- ④ src 폴더 안에 wishlist 폴더를 생성한다. => 그 안에 다음 파일들을 작성한다:

```
src/  
├── wishlist/  
│   ├── WishlistContext.jsx    // 전역 상태 관리  
│   ├── ProductList.jsx       // 상품 목록 출력  
│   ├── WishlistPage.jsx      // 찜 목록만 따로 보기  
│   └── Wishlist.css           // 스타일 작성
```



## 문제] 다음의 조건에 만족하도록 React를 작성 하시오.

### 조건

- ① 상품 목록 출력 (ProductList.jsx)
- ② useState()를 이용하여 샘플 상품 데이터를 배열로 관리한다.
- ③ 아래는 임시 상품 데이터이다. (id, name, image, price 등 최소 3개 이상)  

```
const products = [  
  { id: 1, name: '노트북', price: 1500000 },  
  { id: 2, name: '마우스', price: 30000 },  
  { id: 3, name: '키보드', price: 80000 },  
];
```
- ① 각 상품 옆에 “찜하기 ♥” 버튼을 구현한다.
- ② 찜 버튼 클릭 시 Context의 addToWishlist() 또는 removeFromWishlist()를 호출한다.
- ③ 찜 된 상품일 경우 버튼 색상이나 아이콘이 다르게 표시되도록 한다.



## 문제] 다음의 조건에 만족하도록 React를 작성 하시오.

### 조건

- ① 찜 목록 보기 (WishlistPage.jsx)
- ② 전역 상태의 wishlist를 불러와 목록으로 출력한다.
- ③ 찜 목록이 없을 경우 "찜한 상품이 없습니다." 메시지를 표시한다.
- ④ 각 항목 옆에는 "삭제" 버튼을 두어 removeFromWishlist() 기능이 동작하도록 한다.
- ⑤ CSS 파일을 각 컴포넌트에서 import './Wishlist.css' 로 불러온다.
- ⑥ 새로고침 시 찜 목록이 초기화되어도 무방하다.
- ⑦ useState, useContext, BrowserRouter, Route, Link를 반드시 사용해야 한다.



## <기초 출력 화면>



### 찜하기 예제

찜 목록 보기



#### 상품 목록

- 노트북 - 1,500,000원 ♡ 찜하기
- 마우스 - 30,000원 ♡ 찜하기
- 키보드 - 80,000원 ♡ 찜하기

현재 찜한 상품 수: 0개

## <상품 찜 한 상태>



### 찜하기 예제

찜 목록 보기



#### 상품 목록

- 노트북 - 1,500,000원 ♡ 찜 해제
- 마우스 - 30,000원 ♡ 찜하기
- 키보드 - 80,000원 ♡ 찜 해제

현재 찜한 상품 수: 2개



## <찜 목록 보기 상태>



# 찜하기 예제

[상품 보기](#)

### ❤ 찜한 상품 목록

- 노트북 - 1,500,000원 [삭제](#)
- 키보드 - 80,000원 [삭제](#)

## <찜 한 상품 삭제 하기>



# 찜하기 예제

[상품 보기](#)

### ❤ 찜한 상품 목록

- 노트북 - 1,500,000원 [삭제](#)