

10. 리스트와 Key

리스트(목록) = 같은 아이템을 순서대로 모아놓은 것
리스트를 담는 자료구조 → 배열

Key = 리스트의 아이템들을 구분하기 위한 고유한 값

10.1 여러 개의 컴포넌트 렌더링 하기(예제)

- 배열을 사용하여 반복되는 여러 개의 컴포넌트들을 쉽게 렌더링 가능
- 컴포넌트를 코드상에 하나씩 직접 넣는 것은 비효율적
- 배열의 `map()` 함수를 이용
 - 배열의 각 요소에 어떤 처리를 한 뒤 리턴하는 것으로 새로운 배열 만들기
 - `map()` 함수 안에 있는 엘리먼트에는 무조건 `Key`가 있어야 함

10.2 리스트의 Key

- 리스트에서 아이템들을 구분하기 위한 고유한 문자열
- 어떤 아이템이 추가, 변경, 삭제되었는지 구분하기 위해 사용
 - (참고) 키가 없다면 재렌더링 시 엘리먼트를 비효율적으로 업데이트
- 다양한 키 값의 사용법
 - **데이터의 id를 사용 (주로 사용)**
 - 인덱스를 사용 (데이터에 고유한 id값이 없을 경우에만 사용)
 - 아이템의 순서가 바뀔 수 있는 경우 권장 안함
 - <https://robinpokorny.medium.com/index-as-a-key-is-an-anti-pattern-e0349aece318>

10.3 출석부 출력하기(실습)

- my-app/src/chapter10 실습

11. Form

사용자로부터 입력을 받기 위해 사용하는 것(입력 양식)
대표적으로 <input>

11.1 HTML Form

- submit하면 지정한 URL로 요청을 보내는 기본 폼 동작을 수행

HTML 폼 엘리먼트는 폼 엘리먼트 자체가 내부 상태를 가지기 때문에, React의 다른 DOM 엘리먼트와 다르게 동작합니다. 예를 들어, 순수한 HTML에서 이 폼은 name을 입력받습니다.

```
<form>
  <label>
    Name:
    <input type="text" name="name" />
  </label>
  <input type="submit" value="Submit" />
</form>
```

- 그러나 대부분의 경우, 자바스크립트 코드에서 사용자가 입력한 데이터에 접근하고 처리 하는 것이 편리
- React에서는 이를 위해 제어 컴포넌트(controlled components)

11.2 제어 컴포넌트(예제)

- 사용자가 입력한 값에 접근하고 제어할 수 있도록 해주는 컴포넌트
- 입력값이 state를 통해 리액트의 통제를 받는 입력 폼 엘리먼트

11.3 다양한 Form(예제)

- `<textarea>` 태그
 - 여러 줄에 걸쳐서 텍스트를 입력 받기 위한 HTML 태그
 - `value` 속성으로 입력된 값을 관리
- `<select>` 태그
 - 드롭다운 목록을 보여주기 위한 HTML 태그
 - 여러 옵션 중에서 하나 또는 여러 개를 선택할 수 있는 기능을 제공
 - `value` 속성으로 입력된 값을 관리
- `<input type="file">` 태그
 - 하나 또는 여러 개의 파일을 선택할 수 있게 해주는 HTML 태그
 - 서버로 파일을 업로드하거나 자바스크립트로 파일을 다룰 때 사용
 - 값이 읽기 전용이기 때문에 React에서는 비제어 컴포넌트가 됨

11.4 여러 개의 입력 제어하기(예제)

- 컴포넌트에 여러 개의 state를 선언하여 각각의 입력에 대해 사용하면 됨

11.5 Input Null 값에 대해(참고)

- 값이 비어 있음을 나타낼 때 value 속성에 null값 사용은 안됨
 - 일반적으로 제어 컴포넌트에서는 빈 문자("")를 사용
 - 비제어 컴포넌트에서는 undefined 값을 사용

11.6 사용자 정보 입력받기(실습)

- my-app/src/chapter11 실습

간단한 블로그 만들기

my-blog 실습