

19. 복잡한 이벤트와 상태 디자인하기

- 동작 설명해보기
- state와 event handler로 나누기
- 비슷한 것 묶기

1. 드롭다운 컴포넌트

- 동작을 하나씩 설명해보기

: 드롭다운 클릭 : 옵션이 나타남 : 하나의 옵션을 클릭함

: 옵션들이 사라짐 : 클릭한 옵션이 박스에 표시됨

2. state와 event handler로 나누기

(1) state : 사용자가 화면에서 변화를 느끼는 것들

(2) event handler : 사용자가 어떤 행동을 취하는 것들

- 드롭다운 클릭 `event handler`
- 옵션이 나타남 `state`
- 하나의 옵션을 클릭함 `event handler`
- 옵션들이 사라짐 `state`
- 클릭한 옵션이 박스에 표시됨 `state`
- 비슷한 것끼리 묶기 : (옵션이 나타남 + 옵션들이 사라짐)

옵션이 열림 `true`, 옵션이 사라짐 `false`

옵션이 선택됨 : `object`, 옵션이 선택되지 않음 : `null`

`event handler` : 옵션을 클릭함, 드롭다운을 클릭함

- 초기 상태

```
import React from "react";
import Dropdown from "../Components/Dropdown"

function App() {
  const options = [
    { label: "빨강", value: "red" },
    { label: "녹색", value: "green" },
    { label: "파랑", value: "blue" },
  ];

  return <Dropdown options={options} />
}

export default App;
```

```
import React, { useState } from "react";

const Dropdown = ({ options }) => {
  return <div>선택하기</div>
};

export default Dropdown;
```

🥕 state `isOpen` : (옵션이 열림 : true), (옵션이 사라짐 : false)

🥕 state `selectedOption` : (옵션이 선택됨 : object), (옵션이 선택되지 않음 : null)

🥕 event handler `handleSelect` : (옵션을 클릭함)

🥕 event handler `handleClick` : (드롭다운을 클릭함)

```
import React, { useState } from "react";

const Dropdown = ({ options }) => {
  const [isOpen, setIsOpen] = useState(false);

  const renderedOptions = options.map((option) => {
    return (
      <div key={option.value}>{option.label}</div>
    )
  });

  return (
    <>
      <div>선택하기</div>
      <{renderedOptions}</>
    </>
  )
};

export default Dropdown;
```

```
import React, { useState } from "react";

const Dropdown = ({ options }) => {
  const [isOpen, setIsOpen] = useState(false);
  const [selectedOption, setSelectedOption] = useState(null);

  const renderedOptions = options.map((option) => {
    return (
      <div key={option.value}>{option.label}</div>
    )
  });

  return (
    <>
      <div>선택하기</div>
      <{renderedOptions}</>
    </>
  )
};

export default Dropdown;
```

```

    )
  });

  return (
    <>
      <div>{!selectedOption ? "선택하기" : selectedOption.label}</div>
      {isOpen && <>{renderedOptions}</>}
    </>
  )
};

export default Dropdown;

```

```

import React, { useState } from "react";

const Dropdown = ({ options }) => {
  const [isOpen, setIsOpen] = useState(false);
  const [selectedOption, setSelectedOption] = useState(null);

  const handleClick = () => {
    setIsOpen((prevState) => !prevState);
  };

  const handleSelect = (option) => {
    setSelectedOption(option);
    setIsOpen(false);
  }

  const renderedOptions = options.map((option) => {
    return (
      <div key={option.value} onClick={() => handleSelect(option)}>
        {option.label}
      </div>
    )
  });

  return (
    <>
      <div onClick={!selectedOption ? "선택하기" : selectedOption.label}>
</div>
      {isOpen && <>{renderedOptions}</>}
    </>
  )
};

export default Dropdown;

```

