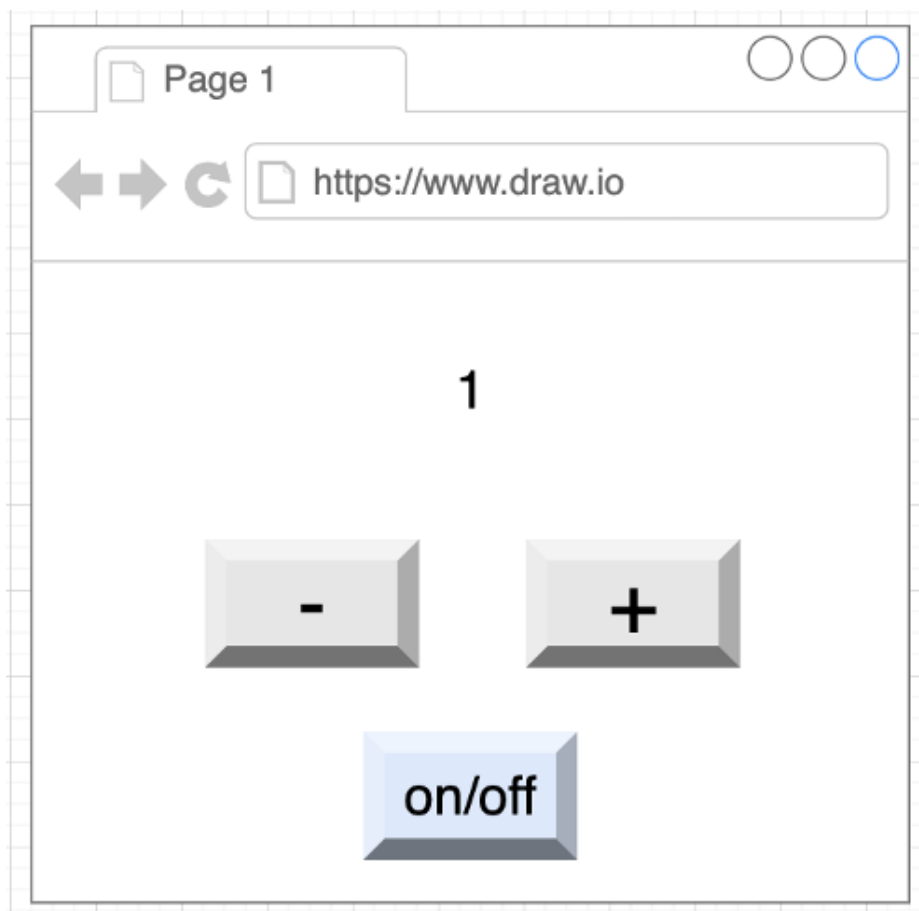


4

4회차 Basic Mission - 노진현

간단한 애플리케이션을 테스트 주도 개발을 사용해서 만들기

만들게 될 애플리케이션은 아래와 같습니다.



+ 버튼을 누르면 1씩 올라가고, - 버튼을 누르면 1이 내려갑니다.
on/off는 누를 때마다 +, - 버튼을 disabled true false로 바꿔서 줍니다.

1. 테스트 세팅

Issue : 테스트 코드 실행 시 React 17 에서 React 18로 버전업 됨에 따라 `React.DOM.render` 가 더 이상 지원되지 않는다는 메시지와 함께 에러가 발생됐다.

```
console.error
Warning: ReactDOM.render is no longer supported in React 18. Use createRoot instead. Until you switch to the new API, your app will behave as if it's running React 17. Learn more: https://reactjs.org/link/switch-to-createroot
```

```
3 |
4 | test("the counter starts at 0", () => {
> 5 |   render(<PlusMinus/>)
    |   ^
```

해결

```
npm install --save-dev @testing-library/react@latest
npm install --save-dev @testing-library/jest-dom@latest
npm install --save-dev @testing-library/user-event@latest
```

참고 : <https://bobbyhadz.com/blog/react-testing-library-reactdom-render-not-supported>

2. 테스트 케이스 작성

1. 카운터 초기값이 0 인지 확인

- `counter`-testid가 부여된 element의 text가 0 인지 확인

```
test("the counter starts at 0", () => {
  render(<PlusMinus/>)

  const counterText = screen.getByTestId("counter")
  expect(counterText).toHaveTextContent("0");
});
```

2. 마이너스 버튼의 텍스트가 올바른지 확인 (-)

- `minus` title이 부여된 버튼의 text가 - 인지 확인

```
test("minus button has correct text", () => {
  render(<PlusMinus/>)
  const minusButton = screen.getByTitle("minus");
  expect(minusButton).toHaveTextContent("-");
});
```

3. 플러스 버튼의 텍스트가 올바른지 확인 (+)

- `plus` title이 부여된 버튼의 text가 `+` 인지 확인

```
test("plus button has correct text", () => {
  render(<PlusMinus/>)
  const plusButton = screen.getByTitle("plus")
  expect(plusButton).toHaveTextContent("+");
});
```

4. 플러스 버튼 클릭 시, 카운터가 `1`로 변하는지 확인

- `fireEvent` 를 이용해서 click 이벤트를 발생시켜 기대하는 동작 확인

```
test("When the + button is pressed, the counter changes to 1", ()=> {
  render(<PlusMinus/>);

  const plusButton = screen.getByRole("button", {
    name : "+"
  });
  const counterText = screen.getByTestId("counter");

  fireEvent.click(plusButton);
  expect(counterText).toHaveTextContent("1");
});
```

5. 마이너스 버튼 클릭 시, 카운터가 `-1`로 변하는지 확인

```
test("When the - button is pressed, the counter changes to -1", () => {
  render(<PlusMinus/>);

  const minusButton = screen.getByRole("button", {
    name : "-"
  });
  const counterText = screen.getByTestId("counter");

  fireEvent.click(minusButton);
  expect(counterText).toHaveTextContent("-1");
});
```

6. 온/오프 버튼이 `파란색`으로 구성되어있는지 확인

- on/off text를 가진 button의 background-color가 blue 인지 확인

```
test("on/off button has blue color", ()=>{
  render(<PlusMinus/>);

  const onoffButton = screen.getByRole("button", {
    name : "on/off"
  });
});
```

```
expect(onoffButton).toHaveStyle("background-color: blue")
});
```

7. 온/오프 버튼 클릭 시마다 플러스/마이너스 버튼이 활성/비활성 상태로 변하는지 확인

- 초기값, 1번 클릭 시, 2번 클릭 시의 상황으로 구분하여 +/-버튼 활성화 여부 확인

```
test("Prevent the -,+ button from being pressed when the on/off button is clicked", () => {
  render(<PlusMinus/>);

  const onoffButton = screen.getByRole("button", {
    name : "on/off"
  });
  const plusButton = screen.getByRole("button", {
    name : "+"
  });
  const minusButton = screen.getByRole("button", {
    name : "-"
  });
  // on/off 버튼 클릭 전(초기값) : +/- 버튼 활성화
  expect(plusButton).not.toHaveAttribute('disabled');
  expect(minusButton).not.toHaveAttribute('disabled');

  // on/off 버튼 클릭 (1회차)
  fireEvent.click(onoffButton);

  // on/off 버튼 1회 클릭 후 : +/- 비활성화
  expect(plusButton).toHaveAttribute('disabled');
  expect(minusButton).toHaveAttribute('disabled');

  // on/off 버튼 클릭 (2회차)
  fireEvent.click(onoffButton);

  // on/off 버튼 2회 클릭 후 : +/- 버튼 활성화
  expect(plusButton).not.toHaveAttribute('disabled');
  expect(minusButton).not.toHaveAttribute('disabled');
});
```

▼ 코드 전문

```
import { render, screen, fireEvent } from '@testing-library/react';
import PlusMinus from "../PlusMinus"

test("the counter starts at 0", () => {
  render(<PlusMinus/>)

  const counterText = screen.getByTestId("counter")
  expect(counterText).toHaveTextContent("0");
});

test("minus button has correct text", () => {
  render(<PlusMinus/>)
  const minusButton = screen.getByTitle("minus");
  expect(minusButton).toHaveTextContent("-");
});
```

```

})

test("plus button has correct text", () => {
  render(<PlusMinus/>)
  const plusButton = screen.getByTitle("plus")
  expect(plusButton).toHaveTextContent("+");
});

test("When the + button is pressed, the counter changes to 1", ()=> {
  render(<PlusMinus/>);

  const plusButton = screen.getByRole("button", {
    name : "+"
  });
  const counterText = screen.getByTestId("counter");

  fireEvent.click(plusButton);
  expect(counterText).toHaveTextContent("1");
});

test("When the - button is pressed, the counter changes to -1", () => {
  render(<PlusMinus/>);

  const minusButton = screen.getByRole("button", {
    name : "-"
  });
  const counterText = screen.getByTestId("counter");

  fireEvent.click(minusButton);
  expect(counterText).toHaveTextContent("-1");
});

test("on/off button has blue color", ()=>{
  render(<PlusMinus/>);

  const onoffButton = screen.getByRole("button", {
    name : "on/off"
  });

  expect(onoffButton).toHaveStyle("background-color: blue")
});

test("Prevent the -,+ button from being pressed when the on/off button is clicked", () => {
  render(<PlusMinus/>);

  const onoffButton = screen.getByRole("button", {
    name : "on/off"
  });
  const plusButton = screen.getByRole("button", {
    name : "+"
  });
  const minusButton = screen.getByRole("button", {
    name : "-"
  });
  // on/off 버튼 클릭 전(초기값) : +/- 버튼 활성화
  expect(plusButton).not.toHaveAttribute('disabled');
  expect(minusButton).not.toHaveAttribute('disabled');

  // on/off 버튼 클릭 (1회차)
  fireEvent.click(onoffButton);

  // on/off 버튼 1회 클릭 후 : +/- 비활성화

```

```

expect(plusButton).toHaveAttribute('disabled');
expect(minusButton).toHaveAttribute('disabled');

// on/off 버튼 클릭 (2회차)
fireEvent.click(onoffButton);

// on/off 버튼 2회 클릭 후 : +/- 버튼 활성화
expect(plusButton).not.toHaveAttribute('disabled');
expect(minusButton).not.toHaveAttribute('disabled');
})

```

3. 테스트를 통과할 수 있는 코드 작성

App.js

```

import './App.css';
import PlusMinus from './pages/PlusMinus';

function App() {
  return (
    <div style={{textAlign: 'center'}}>
      <PlusMinus/>
    </div>
  );
}

export default App;

```

PlusMinus.js

- test에서 작성한 기대에 맞게 코드 구성
- counter의 값(`countValue`)과 +/- 버튼 disabled 여부를 `useState`로 관리
- +/- 버튼과 on/off 버튼에서 기대되는 `onClick` 함수 작성

```

import React, { useState } from "react";
import './PlusMinus.css';

const PlusMinus = () => {
  const [countValue, setCountValue] = useState(0);
  const [isDisabled, setIsDisabled] = useState(false);

  const onClickPlus = () => {
    setCountValue(countValue + 1);
  };

  const onClickMinus = () => {
    setCountValue(countValue - 1);
  };
}

```

```

const onClickOnOff = () => {
  setIsDisabled(!isDisabled);
};

return (
  <div>
    <h1 data-testid="counter">{countValue}</h1>
    <button title="plus" onClick={onClickPlus} disabled={isDisabled}>
      +
    </button>
    <button title="minus" onClick={onClickMinus} disabled={isDisabled}>
      -
    </button>
    <br />
    <button
      onClick={onClickOnOff}
      style={{ backgroundColor: "blue", borderColor: "blue" }}
    >
      on/off
    </button>
  </div>
);
};

export default PlusMinus;

```

PlusMinus.css

```

button {
  margin: 1rem;
  width: 20%;
  padding: 20px;
  font-size: 2rem;
  font-weight: 700;
  border-width: 10px;
  border-color: whitesmoke;

  -webkit-appearance: none;
  -moz-appearance: none;
  appearance: none;
  cursor: pointer;
}

```

4. 테스트 결과 확인

```
PROBLEMS 1 OUTPUT TERMINAL DEBUG CONSOLE

PASS src/App.test.js
PASS src/pages/test/PlusMinus.test.js

Test Suites: 2 passed, 2 total
Tests:       8 passed, 8 total
Snapshots:   0 total
Time:        1.666 s, estimated 2 s
Ran all test suites related to changed files.

Watch Usage: Press w to show more.
```

- All Pass
- 테스트코드는 PlusMinus.test.js에 작성. App.test.js는 관계없음