

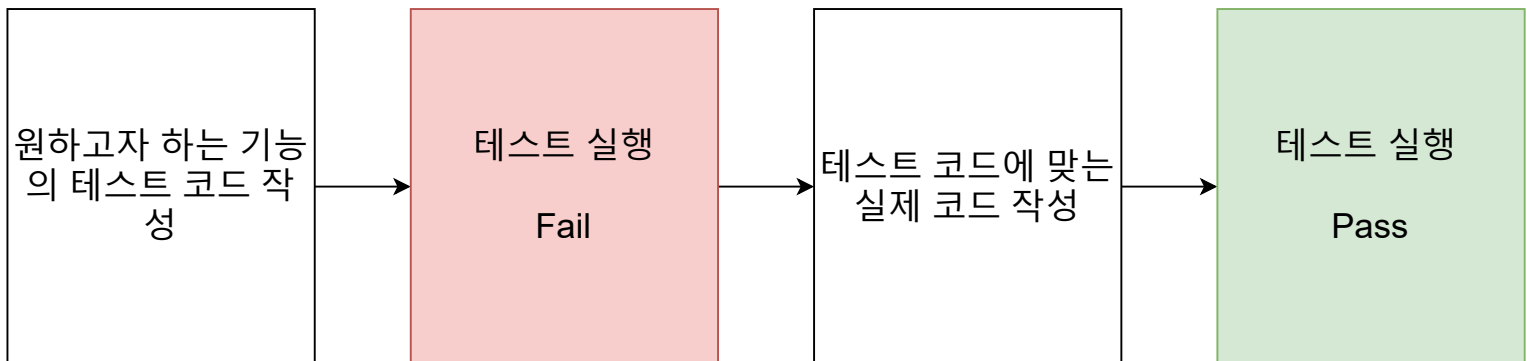
테스트 주도 개발(Test Driven Development)

테스팅을 위한 프로젝트 설정들과 테스팅에 대한 대략적인 설명도 했기 때문에 이제는 간단한 애플리케이션을 만들면서 테스팅을 직접 해볼 것입니다. 하지만 바로 시작하기 전에 앱만드는 순서를 TDD 방식을 이용해서 할 것이기에 TDD가 무엇인지 잠시 보고 가겠습니다.

Test Driven Development 란 무엇인가요?

실제 코드를 작성하기 전에 테스트 코드를 먼저 작성합니다.

테스트 코드를 작성한 후 그 테스트 코드를 Pass 할 수 있는 실제 코드를 작성합니다.

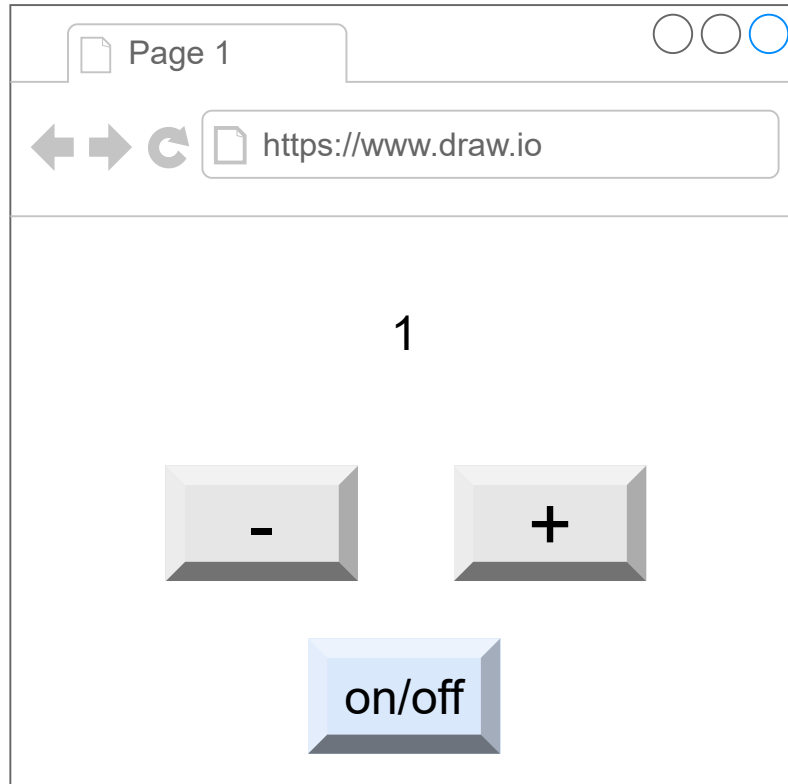


TDD를 하면 좋은 점

1. TDD를 하므로 인해 많은 기능을 테스트하기에 소스 코드에 안정감이 부여된다.
2. 실제 개발하면서 많은 시간이 소요되는 부분은 디버깅 부분이기에 TDD를 사용하면 디버깅 시간이 줄어들고 실제 개발 시간도 줄어듭니다.
3. 소스 코드 하나하나를 더욱 신중하게 짤 수 있기 때문에 깨끗한 코드가 나올 확률이 높습니다.

간단한 앱 소개

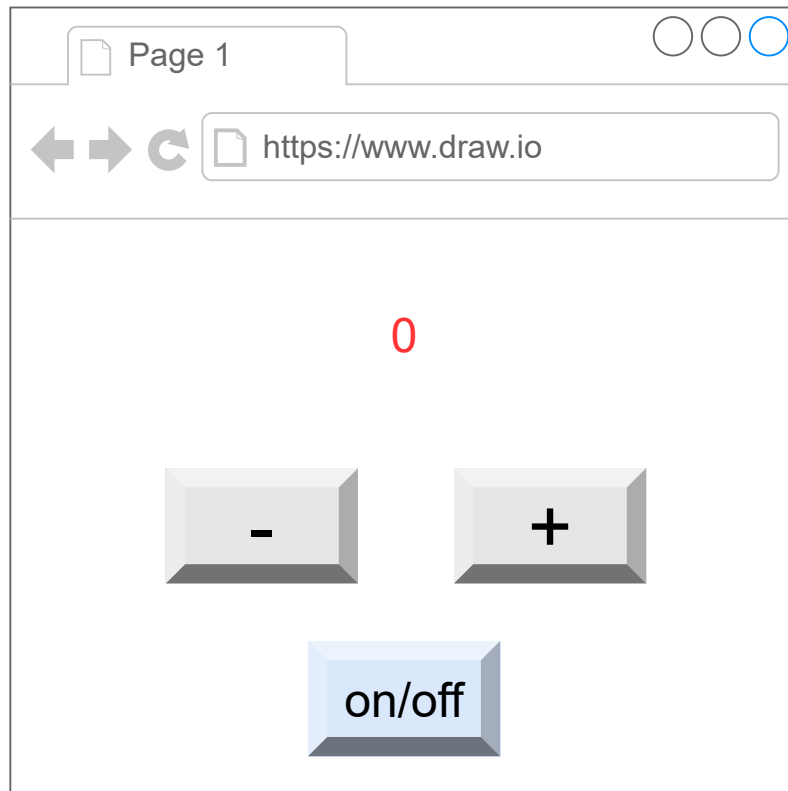
이제부터 앱 두개를 만들겠습니다. 처음으로는 간단한 앱을 만들겠습니다.



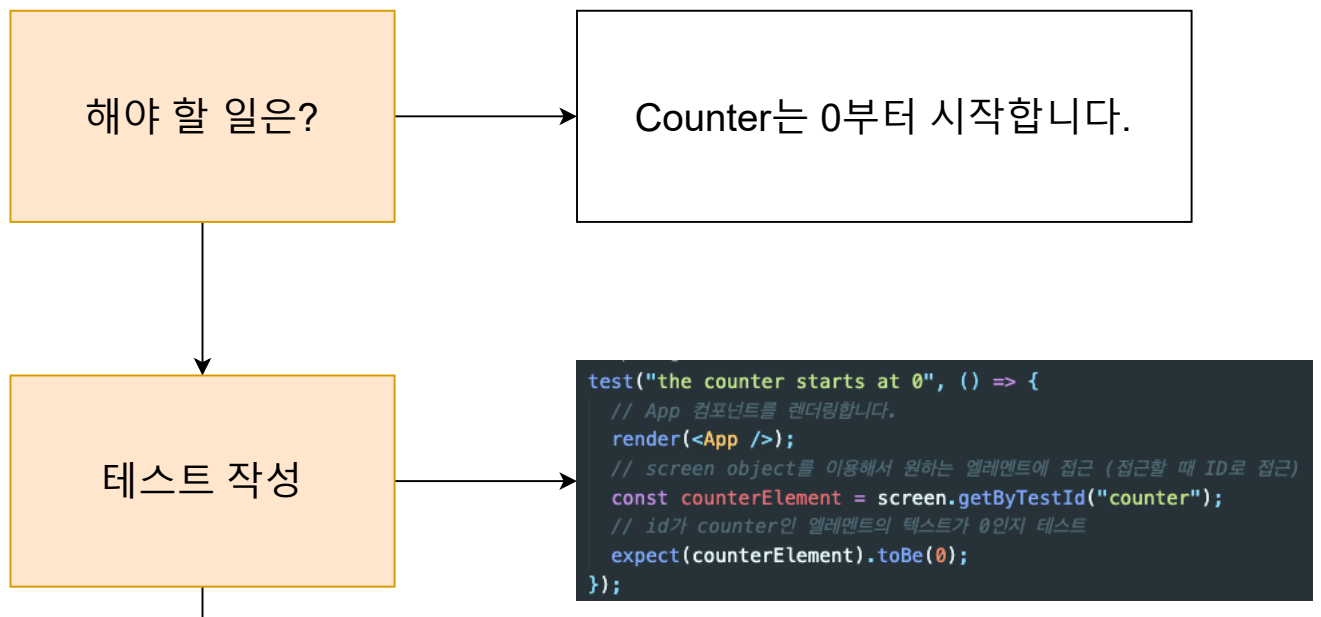
+, -, 그리고 on/off 버튼이 있습니다. 거기서 +를 누르면 숫자가 올라가고 -를 누르면 내려갑니다. 그리고 on/off 버튼(푸른색)을 누르면 +, - 버튼이 작동을 안하고 색깔이 변하는 간단한 앱을 만들어 보겠습니다.

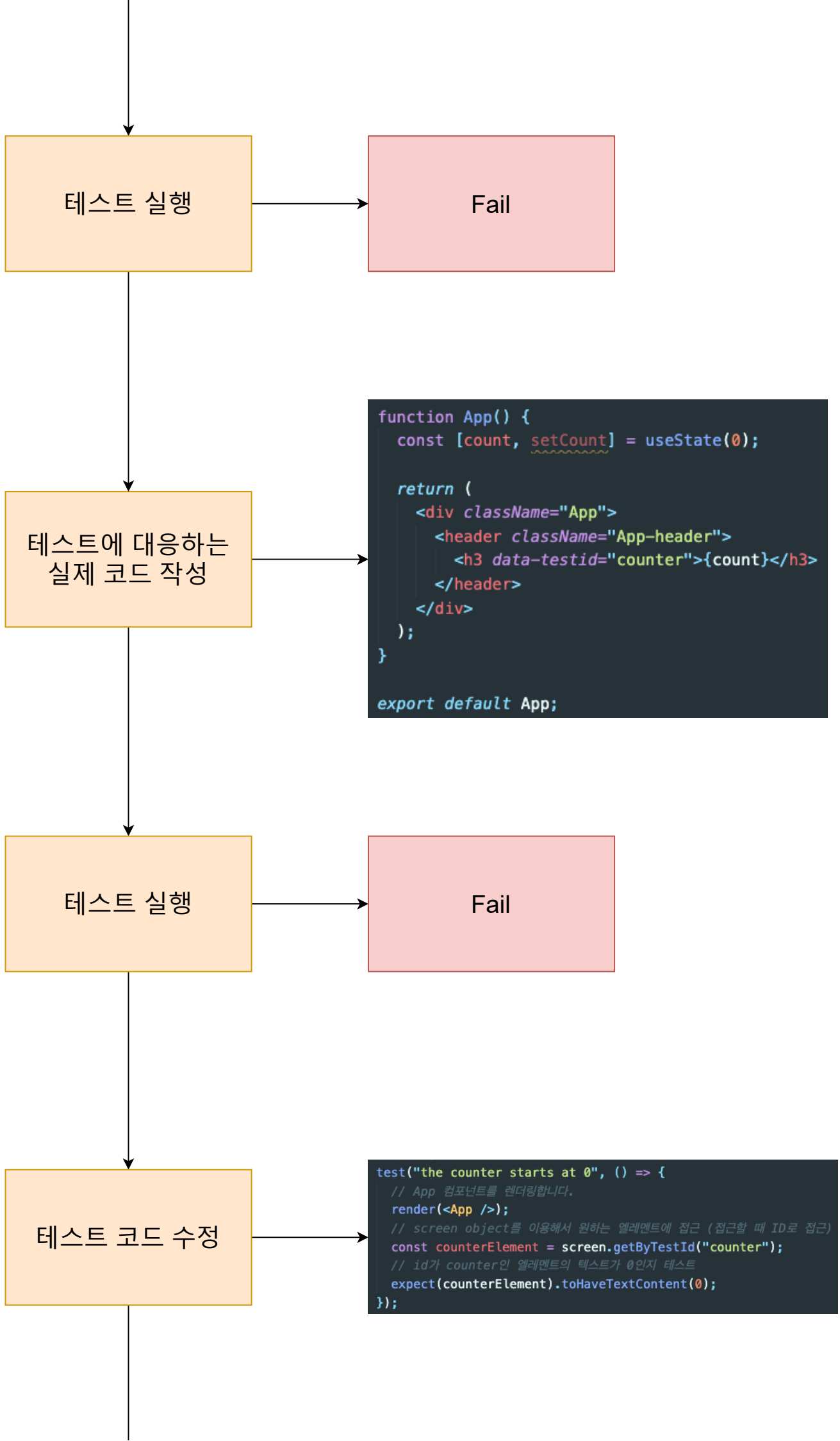
간단한 앱 만들기 시작

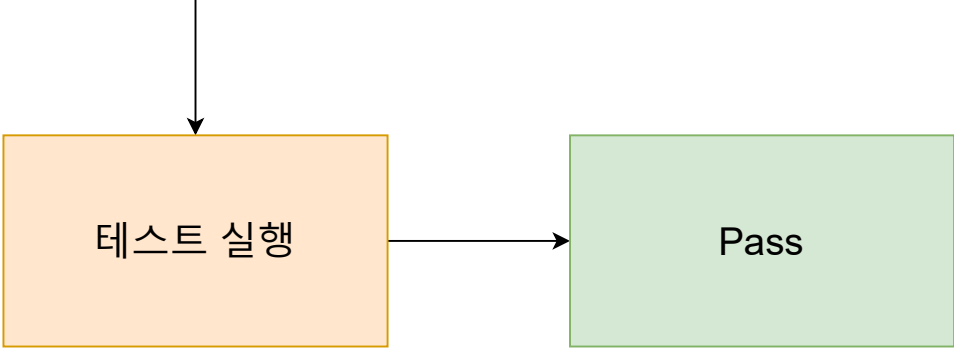
이제는 간단한 앱 만드는 걸 시작해보겠습니다. 테스트 주도 개발을 할 것이기 때문에 먼저 테스트 코드 부터 작성해보겠습니다.



Counter 생성

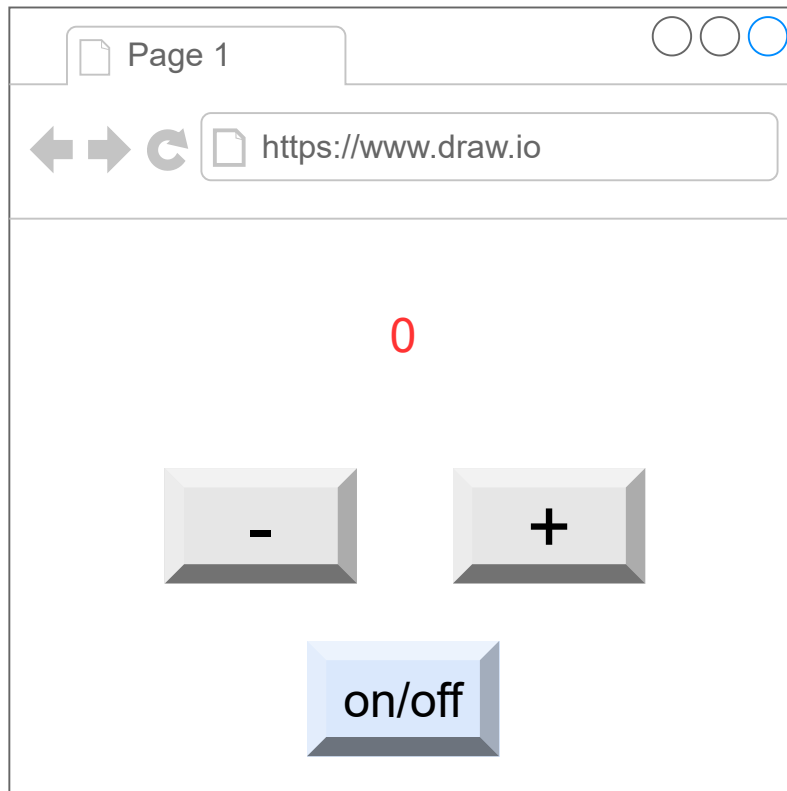






플러스, 마이너스 버튼 생성

카운터를 올리고 내릴 수 있는 버튼을 생성해보겠습니다.

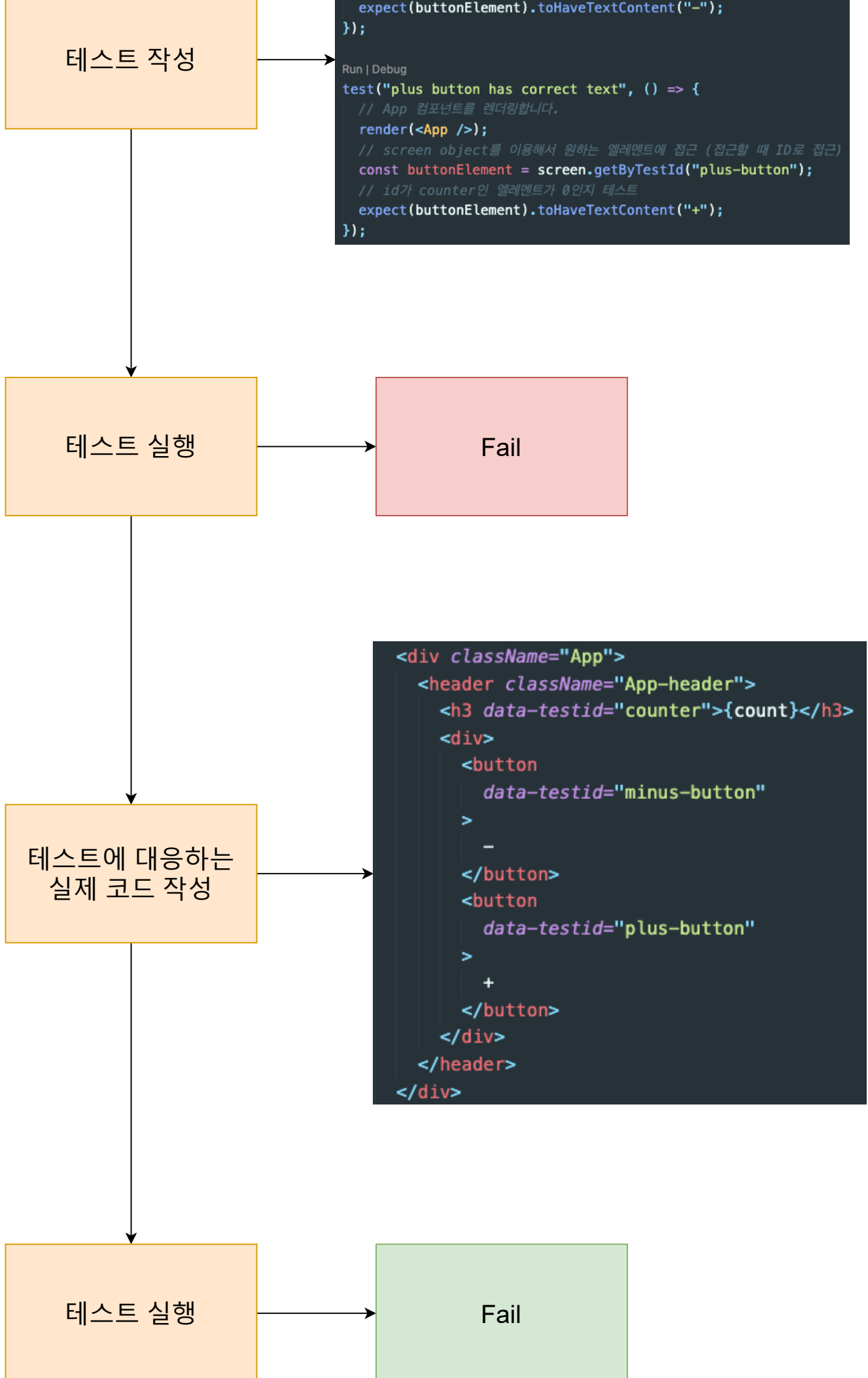


버튼 생성

해야 할 일은?

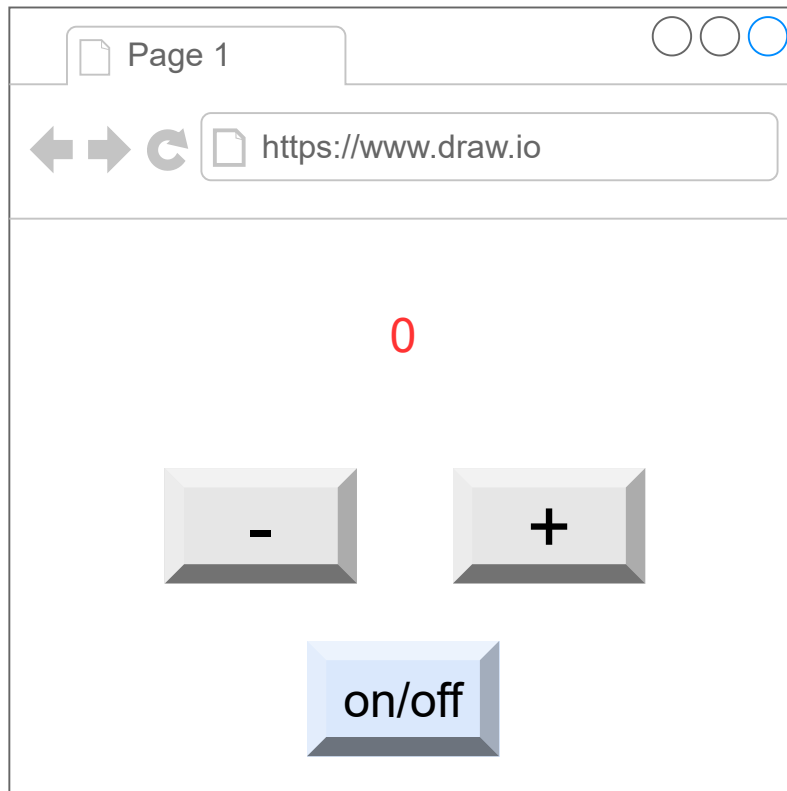
+, - 버튼 두개를 생성합니다.

```
test("minus button has correct text", () => {  
  // App 컴포넌트를 렌더링합니다.  
  render(<App />);  
  // screen object를 이용해서 원하는 요소에 접근 (접근할 때 ID로 접근)  
  const buttonElement = screen.getByTestId("minus-button");  
  // id가 counter인 요소가 0인지 테스트
```



플러스, 마이너스 버튼 기능 넣기(fire event)

카운터를 올리고 내릴 수 있는 버튼의 기능을 넣어서 카운터를 변화시켜 주겠습니다.



FireEvent API

유저가 발생시키는 액션(이벤트)에 대한 테스트를 해야하는 경우 사용합니다.

<https://testing-library.com/docs/dom-testing-library/api-events/>

버튼 생성

해당 html 요소

+ 버튼을 누르면

해야 할 일은?

카운터가 1로 변하게 됩니다.

테스트 작성

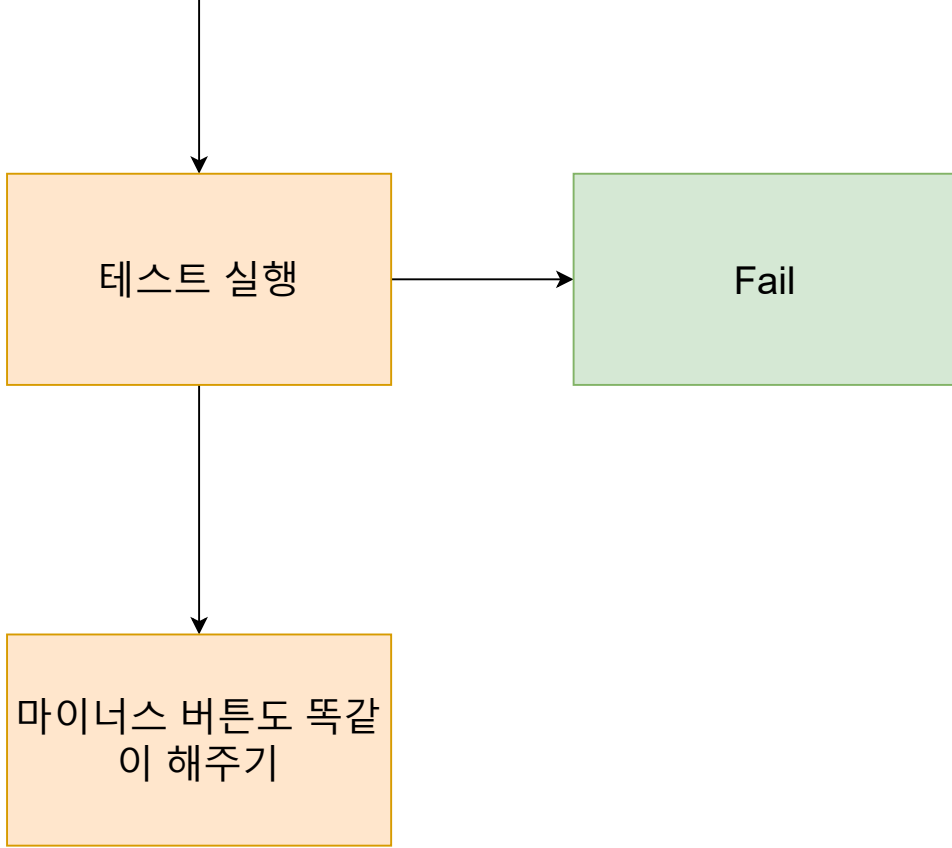
```
test("When the + button is pressed, the counter changes to 1", () => {  
  // App 컴포넌트를 렌더링합니다.  
  render(<App />);  
  // screen object를 이용해서 원하는 엘리먼트에 접근 (접근할 때 ID로 접근)  
  const buttonElement = screen.getByTestId("plus-button");  
  // click plus button  
  fireEvent.click(buttonElement);  
  // 카운터가 0에서 +1 돼서 1이 됩니다.  
  const counterElement = screen.getByTestId("counter");  
  expect(counterElement).toHaveTextContent(1);  
});
```

테스트 실행

Fail

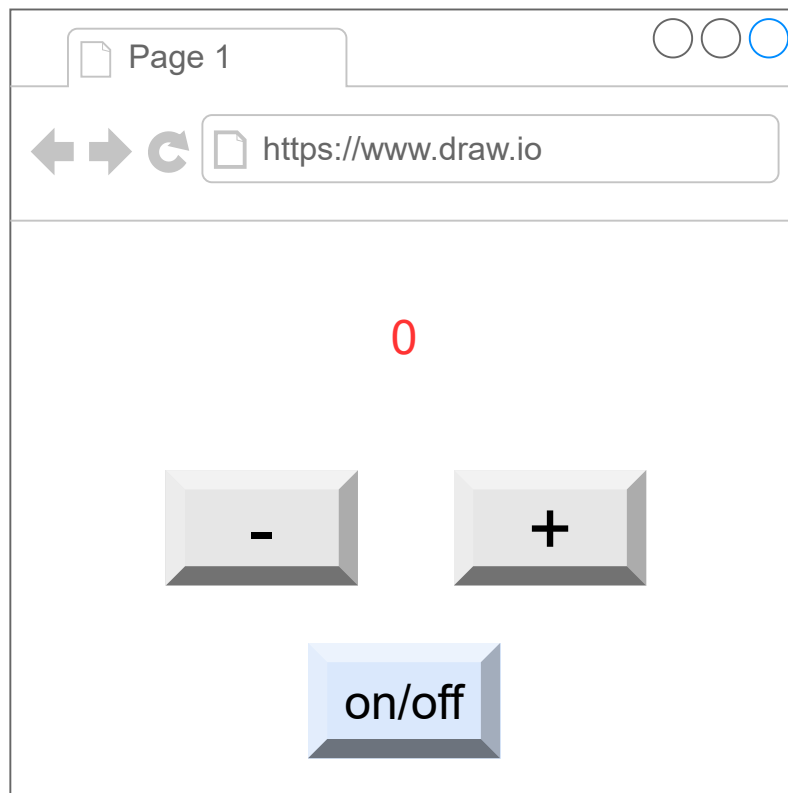
테스트에 대응하는
실제 코드 작성

```
<button  
  data-testid="plus-button"  
  onClick={() => setCount((count) => count + 1)}  
>  
  +  
</button>
```



on/off 버튼 만들기(toHaveStyle)

on/off 버튼을 만들어보겠습니다.



on/off 버튼 생성

해야 할 일은?

on/off 버튼을 만드는데 이 버튼은 파란색으로 스타일을 주겠습니다.

<https://github.com/testing-library/jest-dom>

```
test("on/off button has blue color", () => {  
  // App 컴포넌트를 렌더링합니다.  
  render(<App />);  
})
```

테스트 작성

```
render(<App />);  
// screen object를 이용해서 원하는 엘리먼트에 접근 (접근할 때 ID로 접근)  
const buttonElement = screen.getByTestId("on/off-button");  
// on/off 버튼 색깔을 블루색으로 ...  
expect(buttonElement).toHaveStyle({ backgroundColor: "blue" });  
});
```

테스트 실행

Fail

테스트에 대응하는
실제 코드 작성

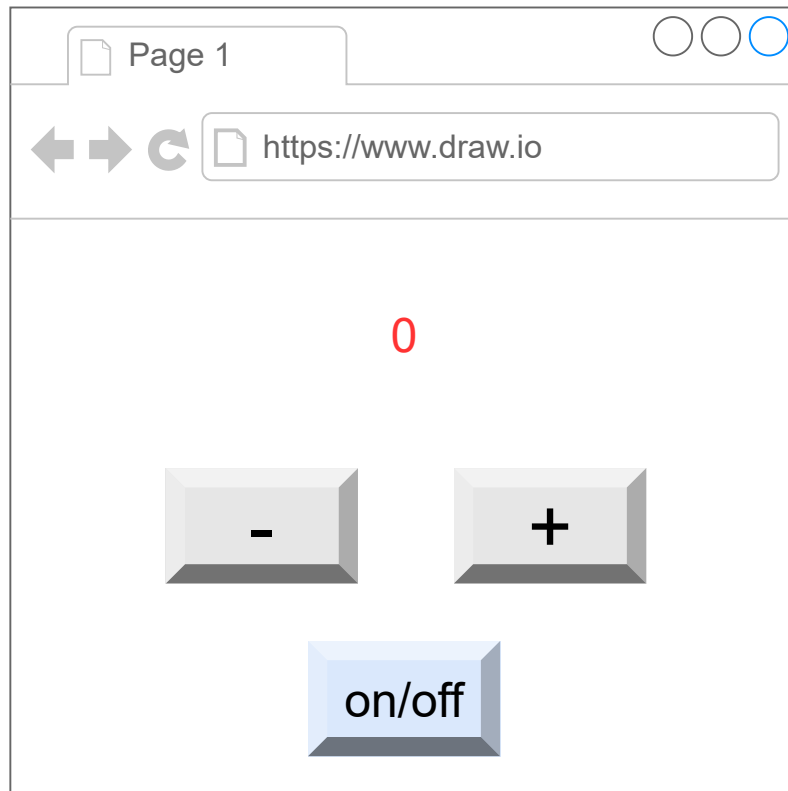
```
<div>  
  <button  
    style={{ backgroundColor: "blue" }}  
    data-testid="on/off-button"  
  >  
    on/off  
  </button>  
</div>
```

테스트 실행

Fail

on/off 버튼 클릭 시 버튼 disabled

on/off 버튼을 클릭 시 -,+ 버튼을 disabled 시켜보겠습니다.



on/off 버튼 생성

해야 할 일은?

on/off 버튼을 클릭할 때 -,+ 버튼을
못누르게 막기

```
test("Prevent the -,+ button from being pressed when the on/off button is clicked", () => {  
  // App 컴포넌트를 렌더링합니다.  
  render(<App />);  
  // screen_object를 이용해서 원하는 엘리먼트에 접근 (접근할 때 ID로 접근)
```

테스트 작성

```
// screen object를 이용해서 원하는 엘리먼트에 접근 (접근할 때 ID로 접근)  
const onOffButtonElement = screen.getByTestId("on/off-button");  
// click onOffButtonElement button  
fireEvent.click(onOffButtonElement);  
// screen object를 이용해서 원하는 엘리먼트에 접근 (접근할 때 ID로 접근)  
const plusButtonElement = screen.getByTestId("plus-button");  
expect(plusButtonElement).toBeDisabled();  
});
```

테스트 실행

Fail

테스트에 대응하는
실제 코드 작성

```
function App() {  
  const [count, setCount] = useState(0);  
  const [disabled, setDisabled] = useState(false);  
  
  return (  
    <div className="App">  
      <header className="App-header">  
        <h3 data-testid="counter">{count}</h3>  
        <div>  
          <button  
            data-testid="minus-button"  
            disabled={disabled}  
            onClick={() => setCount((count) => count - 1)}  
          >  
            -  
          </button>{" "  
          <button  
            data-testid="plus-button"  
            disabled={disabled}  
            onClick={() => setCount((count) => count + 1)}  
          >  
            +  
          </button>  
        </div>  
        <div>  
          <button  
            style={{ backgroundColor: "blue" }}  
            data-testid="on/off-button"  
            onClick={() => setDisabled((prev) => !prev)}  
          >  
            on/off  
          </button>  
        </div>  
      </div>  
    )  
  );  
}
```

```
        </div>  
      </header>  
    </div>  
  );  
}
```

