section4

1. 주요 구성 파일 :

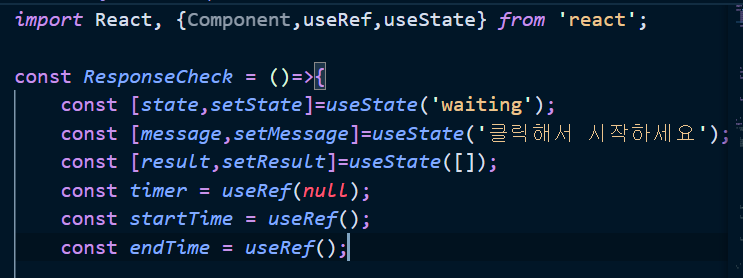
- responseCheck.jsx,

- client.jsx

- webpack.config.js

2. 코드 리뷰

- responseCheck.jsx:



- state 리스트들 :

- state : 현재 상태 : waiting → ready → now 순서로 변경되고, 각 상태마다 UI 가 바뀐다. useState() 메서드로 초기값을 ‘waiting’ 으로 지정.’

- message : 각 state의 변경됨에 따라 UI에 나타내어질 state. UseState() 메서드로 초기값을 ‘클릭해서 시작하세요’ 라고 지정됨.

- result : 배열형태의 state 이다. now 로 state 가 바뀐 이후 얼마나 빨리 클릭하였는지의 기록들을 담아넣는 배열.

- timer : useRef() 를 통해 지정. 통상 useRef 를 DOM 을 지정하지만, setTimer 혹은 setInterval 같은 API를 reference 해준다. setTimer() api를 reference 해주면서 click한 시간을 재는 용도의 state.

- endTime , startTime : Date.now() api를 useRef를 통해 지정. 시작시간 & 끝시간을 기록하는 useRef.

※ useRef 의 반환값은 hooks의 생명주기 내내 유지되며, 호출이 된다해도 re-rendering 이 발생하지 않는다. useRef에서 변경 가능한 객체를 지정해 줄 때, 지정된 변수.current = 대입할 변수

이런식으로 해줌.

- UI 요소를 클릭할 때 click 이벤트로 들어가는 함수.



- state 가 waiting 일 때: state를 ready 바 바꿔준 후 message 를 ‘초록색 되면 click’ 으로 바꿔줌. useRef를 통해 지정된 timer 변수에 setTimeout api 매서드를 reference 해준다. setTImeout 의 callback 함수를 통해 state를 now로 바꾸고 msessage 를 ‘지금 click’으로 바꾼다.

※useRef() 로 초기화된 변수는 변수명.current 를 통해 지정할 DOM객체 혹은 api함수 객체를 지정한다.

※ 상태를 바꿔주는 setState() 매서드들은 비동기 함수이다. 그리고 매번 실행 될 때마다 react hook이 재실행되면서 return으로 반환되는 dom이 다시 랜더링이 된다. 물론 비동기 이기 때문에 브라우져에서 상태값을 바꿔주는 다른 setState() 함수가 있는지 확인 후, 존재한다면 전체 setState() 함수가 실행들이 완료가 된 후에 다시 rendering 을 한다.

- state 가 ready 일 때 : click 할 때 state가 ready 라면 clearInterval() 을 통해 timer가 취소되고 state 가 다시 waiting 으로 바뀐다.

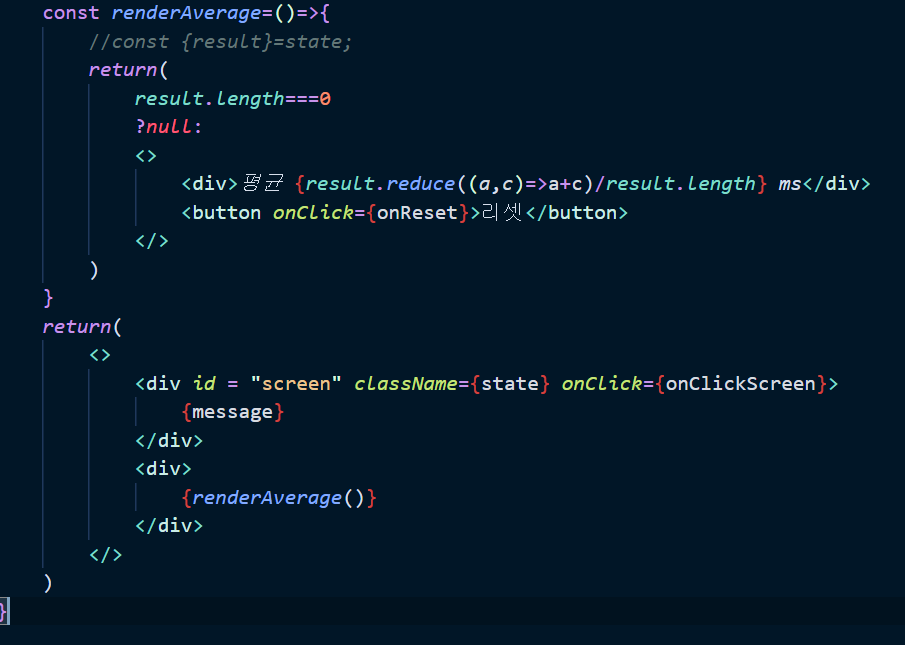
- state 가 now 일 때 : now 일 때 click 하면 endTime 이 계산되며, endTime 에서 startTime을 뺀 값에 result 상태값 (배열 형태의) 이 갱신된다. 즉 새로운 값이 result 배열에 push 되는데, 이때 result 에 직접 push 하는 것이 아닌, setResult() 함수를 사용하여, 내부의 콜백을 통해 갱신한다.

※ setResult((prevResult : 배열이 갱신되기전 상태 )=>{

return [...prevResult:갱신되기전 배열의 주소값, obj : 배열에 새로 추가할 요소]

})

이런 방식으로 갱신됨...



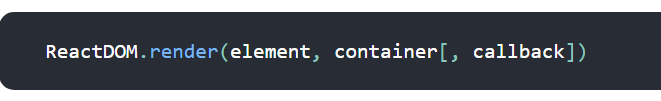
- return 을 통해 위와 같은 dom 을 반환해준다. 그리고 renderAverage(0 함수를 통해 역대 평균 기록을 나타나게 해준다.

-client.jsx



- ReactDom.render 매서드를 통해 html 파일의 root 의 id를 가진 dom에 responseCheck.jsx 파일에서 반환된 dom 형태를 render해줌.

※ ReactDOM.render() :



- container DOM에 렌더링 하고 컴포넌트에 대한 참조를 반환함. 이전에 container 내부에 렌더링 되었다면 해당 엘리먼트는 업데이트하고 최신의 React 엘리먼트를 반영하는 데 필요한 DOM만 변경. React DOM 의 diffing 알고리즘을 사용하여 효율적으로 업데이트 함.