

5.Ref:DOM에 이름달기

작성일시: 2022년 5월 9일 오전 10:54 참고 도서: 리엑트를 다루는 기술

5.0 Ref란?

리엑트 프로그램 내부에서 DOM에 이름 붙이는 방법을 ref(reference의 줄임말)개념 입니다.

Hook 수업예제 보충

MyRef.js

```
import React from 'react';
import MyBox from '../components/MyBox';

/**
 * React에서 document.getElementById(...)에 해당하는 기능을 사용하는 방법
 */
const MyRef = () => {
  // 컴포넌트 렌더링시 콘솔에 출력된 내역 삭제하기
  React.useEffect(() => console.clear(), []);

  // HTML 태그를 react 안에서 참조할 수 있는 변수를 생성
  const myDname = React.useRef();
  const myLoc = React.useRef();
  const myResult = React.useRef();

  // 컴포넌트에 설정하기 위한 ref
  const myBoxRef = React.useRef();

  return (
    <div>
      <h2>My Ref</h2>

      {/* 미리 준비한 컴포넌트 참조변수와 HTML 태그를 연결 */}
      <div>
        <label htmlFor='dname'>학과명</label>
        <input type='text' ref={myDname} id="dname"/>
      </div>
      <div>
        <label htmlFor='dname'>학과위치</label>
        <input type='text' ref={myLoc} id="loc"/>
      </div>
      <h3>
        입력값: <span ref={myResult}></span>
      </h3>
      <button onClick={e => {
        // 컴포넌트 참조변수를 사용해서 다른 HTML태그에 접근 가능
        // --> "참조변수,current" 해당 HTML을 의미하는 Javascript DOM객체
        // --> myDname.current와 document.querySelector(...),
```

```

document.getElementById(...)등으로 생성한 객체가 동일한 DOM객체이다.
    console.log(myDname);
    console.log(myLoc);

    const dname = myDname.current.value;
    const loc = myLoc.current.value;

    myResult.current.innerHTML = dname + ", " + loc;
  }}>클릭</button>
<hr/>

<h3>컴포넌트에 ref 적용하기</h3>

{/* ref 참조변수를 컴포넌트에 적용한다. */}
<MyBox ref={myBoxRef}/>
<button type='button' onClick={() =>{
  //<MyBox>를 통해 myBoxRef를 주입받는 DOM에 접근하여 제어함.
  myBoxRef.current.style.backgroundColor = '#f00';
}}>Red</button>

<button type='button' onClick={() =>{
  //<MyBox>를 통해 myBoxRef를 주입받는 DOM에 접근하여 제어함.
  myBoxRef.current.style.backgroundColor = '#00f'
}}>Blue</button>
</div>
);
};

export default MyRef;

```

MyBox.js

```

import React from 'react'
// 부모로부터 전달받은 ref 참조변수를 받기 위해 "React.forwardRef" hook에 대한 콜백으
로 컴포넌트를 구현한다.
// 이렇게 구현된 컴포넌트는 props와 부모로부터 전달받은 ref 참조변수를 파라미터로 주입받
는다.

const MyBox = React.forwardRef ((props, ref) => {
  const containerStyle = {
    border: '1px solid black',
    height: '100px',
    width: '100px',
  };
  return (
    <div style={containerStyle} ref={ref}></div>
  );
});

export default MyBox;

```

07_hook_event

Mystate | **DateRange1** | MyEffect | MyRef | MyReducer | DateRange2 | MyMemo | MyCallback | MyWidth

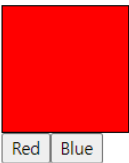
My Ref

학과명

학과위치

입력값:

컴포넌트에 ref 적용하기



07_hook_event

Mystate | **DateRange1** | MyEffect | MyRef | MyReducer | DateRange2 | MyMemo | MyCallback | MyWidth

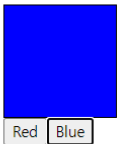
My Ref

학과명

학과위치

입력값:

컴포넌트에 ref 적용하기



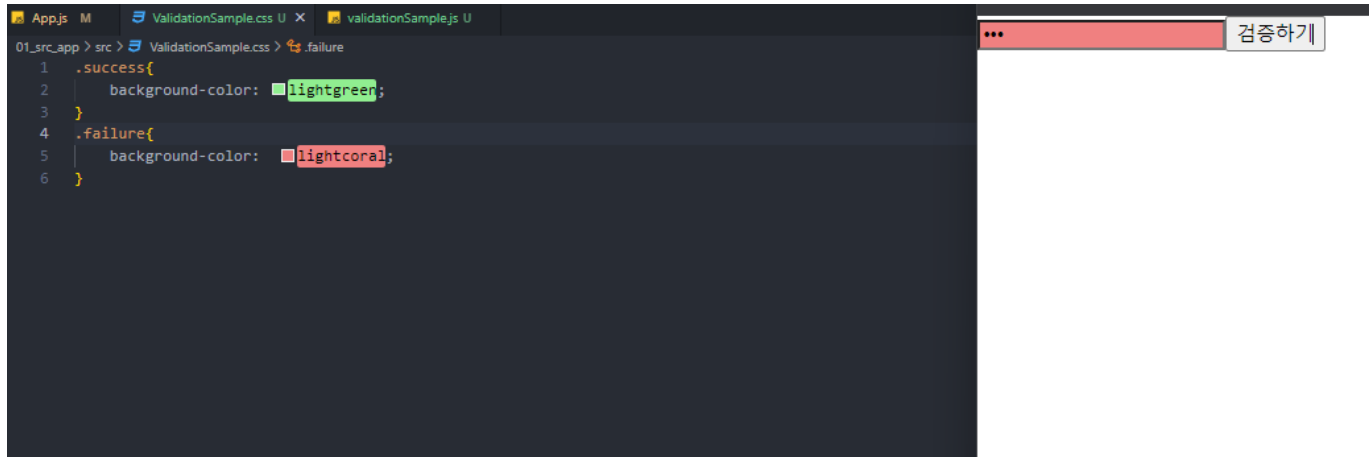
5.1 ref는 어떤 상황에서 사용해야 할까?

- DOM을 직접적으로 건드려야 할 때이다.

5.1.1 예제 컴포넌트 생성

ValidationSample.css

```
background-color: lightgreen;
}
.failure{
  background-color: lightcoral;
}
```



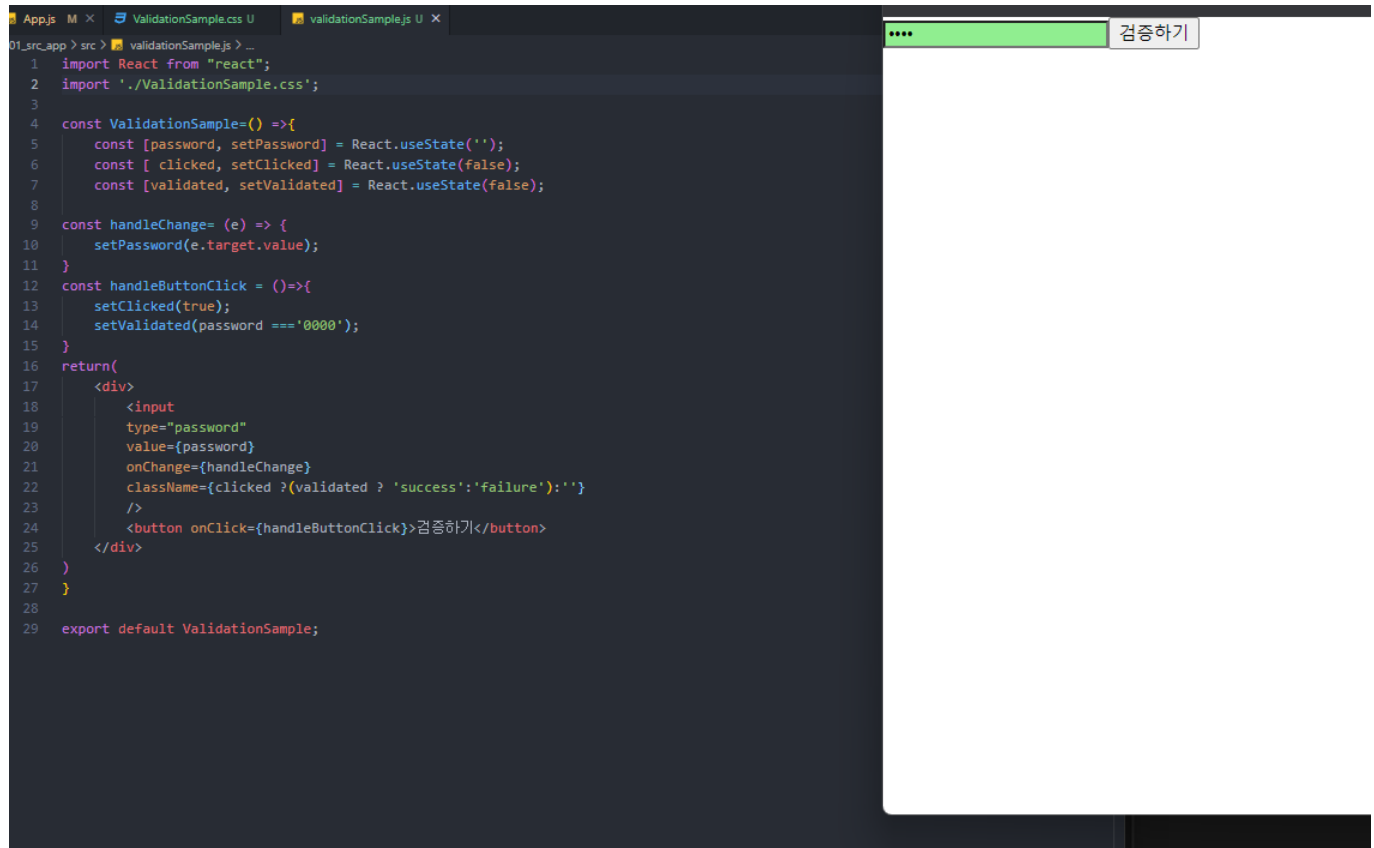
ValidationSample.js

```
import React from "react";
import './ValidationSample.css';

const ValidationSample=() =>{
  const [password, setPassword] = React.useState('');
  const [ clicked, setClicked] = React.useState(false);
  const [validated, setValidated] = React.useState(false);

  const handleChange= (e) => {
    setPassword(e.target.value);
  }
  const handleClick = ()=>{
    setClicked(true); //버튼을 누르면
    setValidated(password === '0000');
    // input 태그의 변경값이 바뀌면 password가 바뀐다.
    // validated = input 태그 이름을 선택한다.
  }
  return(
    <div>
      <input
        type="password"
        value={password}
        onChange={handleChange}
        className={clicked ?(validated ? 'success':'failure'):''}
      />
      { /* 삼항연산자를 통해 이중중첩 if문처럼 사용함 */}
      <button onClick={handleButtonClick}>검증하기</button>
    </div>
  )
}
```

```
export default ValidationSample;
```



5.1.2 App 컴포넌트에서 예제 컴포넌트 렌더링

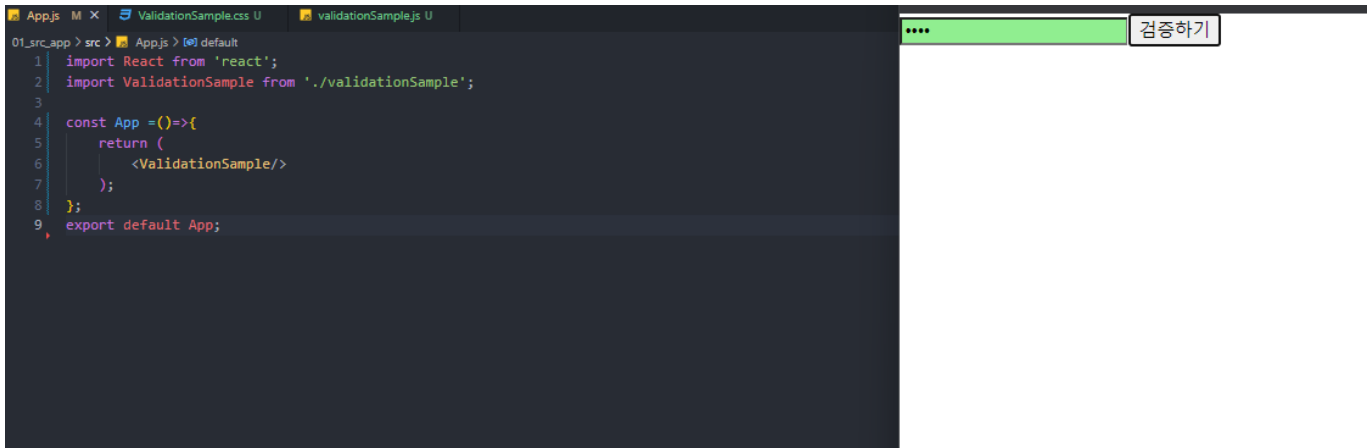
App.js

```

import React from 'react';
import ValidationSample from './validationSample';

const App = ()=>{
  return (
    <ValidationSample/>
  );
};
export default App;

```



5.1.3 DOM을 꼭 사용해야 하는 상황

useState만으로 해결할 수 없는 기능이 있다.

1. 특정 input에 포커스 주기
2. 스크롤 박스 조작하기
- 3. Canvas요소에 그림 그리기등

Canvas라는 태그는 HTML 안에 Javascript를 통해 그림을 그리기가 가능하고, Time 속성을 가지고 움직일 수 있다. ⇒ 2D게임을 만들 수 있다.

5.2.2 React.useRef를 통한 ref설정

ref를 만드는 또 다른 방법은 리액트에 내장되어있는 React.useRef()라는 함수를 사용.

React.useRef 사용예시

```
import React from "react";
const RefSample=() => {
  const input = React.useRef();
  const handleFocus = () => {
    input.current.focus();
  }
  return (
    <div>
      <input ref={input}/>
    </div>
  );
}
export default RefSample;
```

5.2.3 적용

5.2.3.2 버튼 onClick 이벤트 코드 수정

버튼 클릭 이벤트에 focus를 강제로 넣어준다.

validationSample.js-handleButtonClick 메서드

```
import React from "react";
import './ValidationSample.css';

const ValidationSample=() =>{
  const [password, setPassword] = React.useState('');
  const [ clicked, setClicked] = React.useState(false);
  const [validated, setValidated] = React.useState(false);
  const input = React.useRef();

  const handleChange= (e) => {
    setPassword(e.target.value);
  }
  const handleClick = ()=>{
    setClicked(true);
    setValidated(password === '0000');
    input.current.focus();
  }
  return(
    <div>
      <input
        type="password"
        value={password}
        onChange={handleChange}
        className={clicked ?(validated ? 'success':'failure'):''}
        ref={input}
      />
      <button onClick={handleButtonClick}>검증하기</button>
    </div>
  )
}

export default ValidationSample;
```

```

1 import React from "react";
2 import './ValidationSample.css';
3
4 const ValidationSample = () => {
5   const [password, setPassword] = React.useState('');
6   const [clicked, setClicked] = React.useState(false);
7   const [validated, setValidated] = React.useState(false);
8   const input = React.useRef();
9
10  const handleChange = (e) => {
11    setPassword(e.target.value);
12  }
13
14  const handleClick = () => {
15    setClicked(true);
16    setValidated(password === '0000');
17    input.current.focus();
18  }
19
20  return (
21    <div>
22      <input
23        type="password"
24        value={password}
25        onChange={handleChange}
26        className={clicked ? (validated ? 'success' : 'failure') : ''}
27        ref={input}
28      />
29      <button onClick={handleClick}>검증하기</button>
30    </div>
31  )
32 }
33
34 export default ValidationSample;

```

5.3 컴포넌트에 ref 달기

컴포넌트에도 ref를 달 수 있다. 컴포넌트 내부에 있는 DOM을 컴포넌트 외부에서 사용할 때 쓴다.

5.3.1 사용법

```

<MyComponent
  ref={참조변수 이름 명시}
/>

```

5.3.2 컴포넌트 초기 설정

5.3.2.1 컴포넌트 파일 생성

```

import React from "react";

const ScrollBox = React.forwardRef((props, ref) => {
  const style = {
    border: '1px solid black',
    height: '300px',
    width: '300px',
    overflow: 'auto',
    position: 'relative'
  };

  const innerStyle = {
    width: '100%',
    height: '650px',
    background: 'linear-gradient(white,black)'
  };

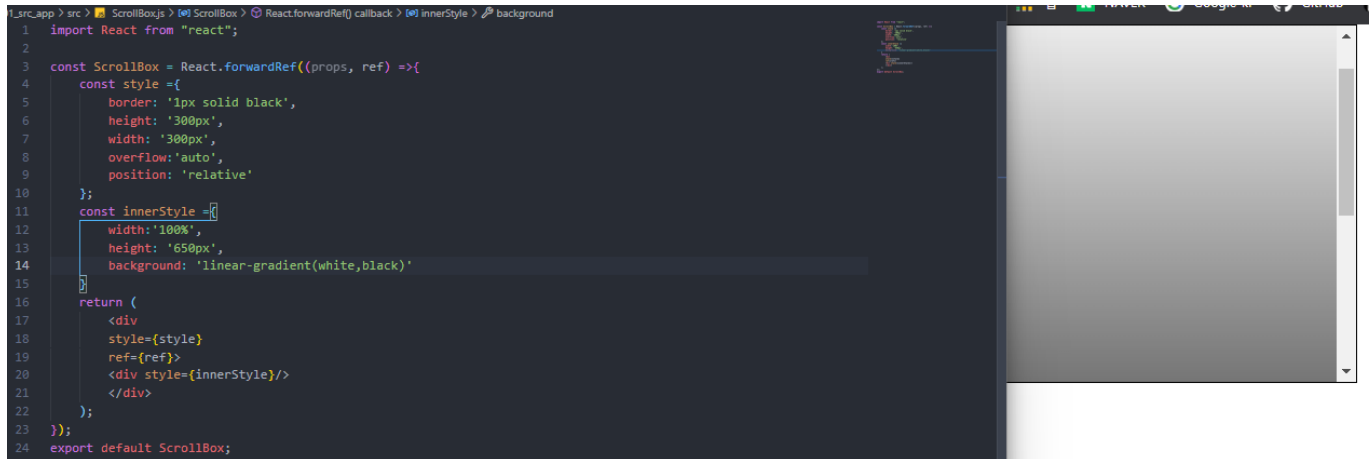
```



```

    }
    return (
      <div
        style={style}
        ref={ref}>
        <div style={innerStyle}/>
      </div>
    );
  });
export default ScrollBox;

```



5.3.2.2 App 컴포넌트에서 스크롤 박스 컴포넌트 렌더링

```

import React from 'react';
import ScrollBox from './ScrollBox';

const App = ()=>{
  return (
    <div>
      <ScrollBox/>
    </div>
  );
};
export default App;

```

5.3.4 컴포넌트에 ref달고 내부 메서드 사용

```

import React from 'react';
import ScrollBox from './ScrollBox';

const App = ()=>{
  const scrollBoxRef = React.useRef();

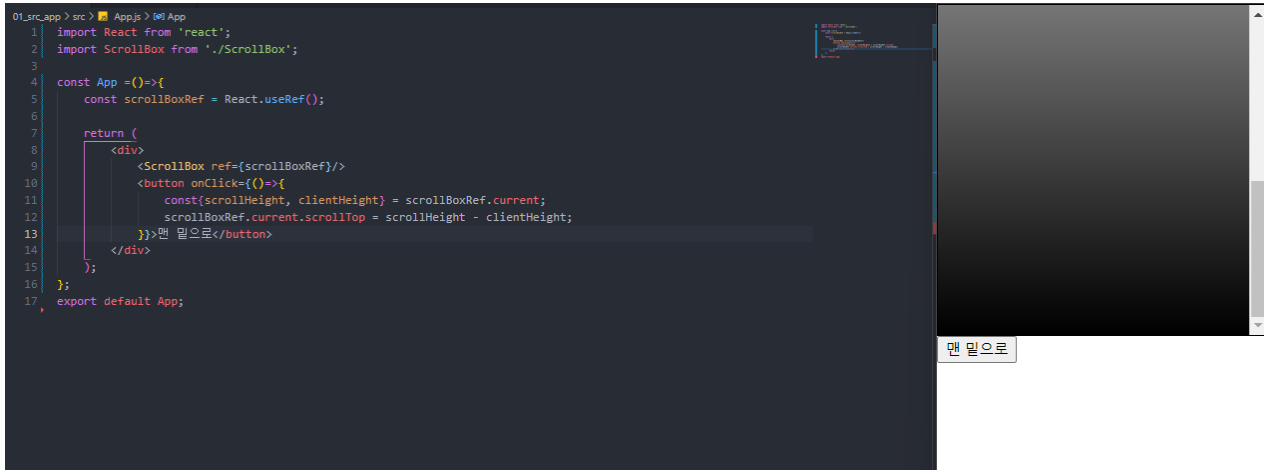
  return (
    <div>

```

```

    <ScrollBar ref={scrollBoxRef}/>
    <button onClick={()=>{
      const {scrollHeight, clientHeight} = scrollBoxRef.current; //부모div
      scrollBoxRef.current.scrollTop = scrollHeight - clientHeight;
      //스크롤이 맨 끝으로 이동
    }}>맨 밑으로</button>
  </div>
);
};
export default App;

```



5.4 정리

1. 컴포넌트 내부에서 DOM에 직접 접근해야 할 때는 ref를 사용한다. ref를 사용하지 않고도 구현이 가능한지 먼저 확인이 필요함.
2. 서로 다른 컴포넌트끼리 데이터를 교류할 때 ref를 사용한다면 할 수는 있지만, 잘못된 코드이다.
3. 컴포넌트끼리 데이터를 교류할 때는 언제나 데이터를 부모 ↔ 자식 흐름을 교류해야함.