**9장 조건부 렌더링**

1. **조건부 렌더링**

**Greeting.jsx**

import React from "react";

function UserGreeting(props){

  return <h1>다시 오셨군요!</h1>

}

function GuestGreeting(props){

  return <h1>회원가입을 해 주세요!</h1>

}

function Greeting(props){

  const isLoggedIn = props.isLoggedIn;

  if (isLoggedIn) {

    return <UserGreeting />

  }

  return <GuestGreeting />

}

export default Greeting;

export {UserGreeting,GuestGreeting};

**Truthy - true는 아니지만 true로 여겨지는 값**

**true**

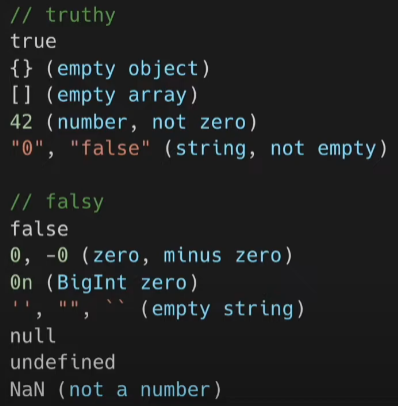
**{ } (empty object)**

**[ ] (empty array)**

**42 (number, not zero)**

**0이 아닌값(\*의 아스키코드값=42)**

**“0”, “false” (string, not empty)**

****

**Falsy - false는 아니지만 false로 여겨지는 값**

**false**

**0, -0 (zero, minus zero)**

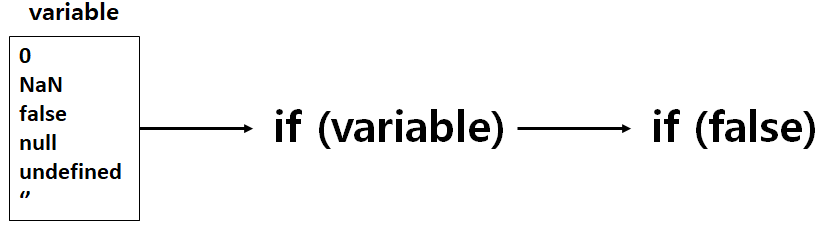
**0n (BigInt zero)**

**‘’,””,``**

**null**

**undefined**

**NaN (not a number)**

****

**falsy 값이 자바스크립트에서 boolean 값을 요구하는 부분에 들어가면 false로 변환된다.**

**조건문이나 반복문에 들어가면 false로 작동한다.**

**falsy에 해당하지 않은 모든 값은 truthy 값으로 보면 된다.**

**이는 boolean 값을 요구하는 부분에서 자바스크립트 내부적으로 변환이 일어날 뿐 true, false 값과 같지 않다.**

import React from "react";

function UserGreeting(props){

  return <h1>다시 오셨군요!</h1>

}

function GuestGreeting(props){

  return <h1>회원가입을 해 주세요!</h1>

}

function Greeting(props){

  const isLoggedIn = props.isLoggedIn;

  if (isLoggedIn) {

    return <UserGreeting />

  }

  return <GuestGreeting />

}

export default Greeting;

export {UserGreeting,GuestGreeting};

**2.엘리먼트 변수**

***엘리먼트 변수 – 엘리먼트를 변수 처럼 다루는 방법***

**컴포넌트로부터 생성된 리액트 엘리먼트를 변수처럼 저장해서 사용하는 방법**

**-별도의 변수로 선언해서 조건부 렌더링을 할 수 있다.**

import React from "react";

function LoginButton(props){

  return (

    <button onClick={props.onClick}>

      로그인

    </button>

  );

}

export default LoginButton;

import React from "react";

function LogoutButton(props){

  return (

    <button onClick={props.onClick}>

      로그아웃

    </button>

  );

}

export default LogoutButton ;

import React,{ useState } from "react";

import LoginButton  from './LoginButton';

import LogoutButton from './LogoutButton';

import Greeting from './Greeting'

function LoginControl(props){

  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const handleLoginClick = () => {

    setIsLoggedIn(true);

  }

  const handleLogoutClick = () => {

    setIsLoggedIn(false);

  }

  let button;

  if (isLoggedIn){

    button = <LogoutButton onClick={handleLogoutClick} />;

  }else{

    button = <LoginButton onClick={handleLoginClick} />;

  }

  return (

    <div>

      <Greeting isLoggedIn={isLoggedIn} />

{button}

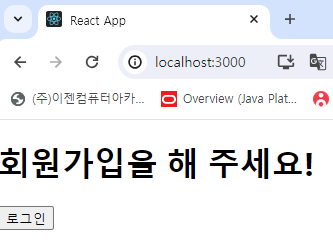
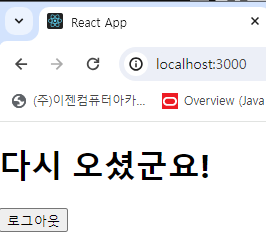
    </div>

  )

}

export default LoginControl;

**로그인의 값에 따라서 button 변수에 컴포넌트를 대입하고 대입된 변수를 return에 넣어 컴포넌트가 렌더링이 되도록 하고 있다. 이 컴포넌트는 리액트 엘리먼트로 생성되어 랜더링됨**

** **

**3.인라인 조건**

***Inline Conditions – && (Short circuit) 으로 구현***

***해당 코드가 필요한 곳에 직접 집어 넣는다.***

**- Inline 조건문을 사용**

**Inline 조건문 – 조건문을 코드 안에 집어넣는 것**

***Inline if***

**true && expression -> expression**

**false && expression -> false**

****

import React , { useState }from "react";

function Mailbox(props){

  const [unreadMessages, setunreadMessages] = useState(["a","b"]);

  //const unreadMessages = props.unreadMessages;

  return (

    <div>

      <h1> 안녕하세요! </h1>

      {unreadMessages.length > 0 &&

        <h2>

          현재 {unreadMessages.length} 개의 읽지 않은 메시지가 있습니다.

        </h2>

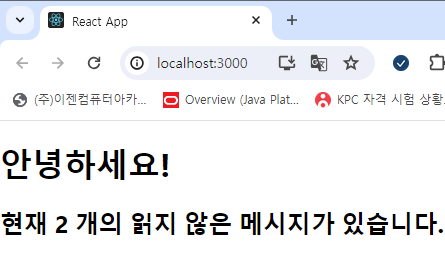
      }

    </div>

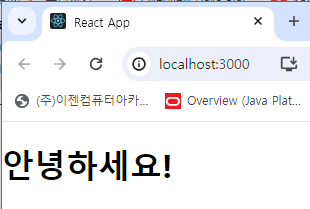
  );

}

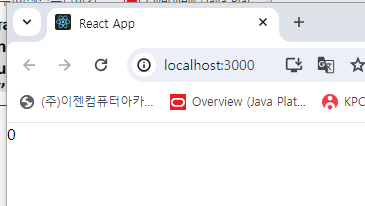
export default Mailbox ;

****

const [unreadMessages, setunreadMessages] = useState([]); 로 설정했을 때 결과

****

import React , { useState }from "react";



function Counter(props){

  const count = 0;

  return (

    <div>

      {count && <h1> 현재 카운트 : {count}</h1>}

    </div>

  );

}

export default Counter ;

***Inline if-else – 삼항연산자 ((조건식)? true : false)로 구현***

import React , { useState }from "react";

function UserStatus(props){

  const [isLoggidIn, setIsLoggidIn] = useState(false);

  return (

    <div>

      이 사용자는 현재

      <b>

        {isLoggidIn ? '로그인' : '로그인하지 않은'}

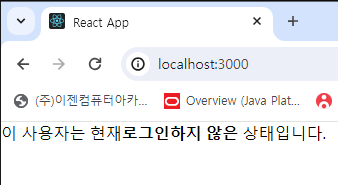
      </b> 상태입니다.

    </div>

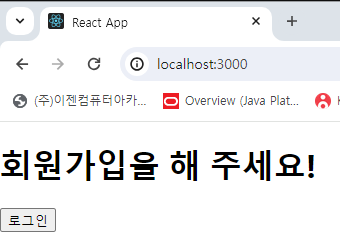
  );

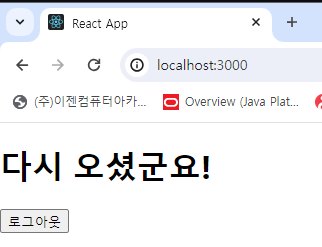
}

export default UserStatus ;

******

import React,{ useState } from "react";





import LoginButton  from './LoginButton';

import LogoutButton from './LogoutButton';

import Greeting from './Greeting'

function LoginControl2(props){

  const [isLoggedIn, setIsLoggedIn] = useState(false);

  const handleLoginClick = () => {

    setIsLoggedIn(true);

  }

  const handleLogoutClick = () => {

    setIsLoggedIn(false);

  }

  return (

    <div>

      <Greeting isLoggedIn={isLoggedIn} />

      {isLoggedIn

        ? <LogoutButton onClick={handleLogoutClick} />

        : <LoginButton onClick={handleLoginClick} />

      }

    </div>

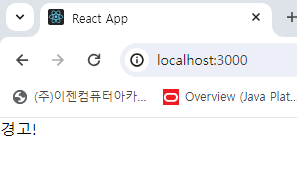
  )

}

export default LoginControl2;

**4.컴포넌트 렌더링 막기**

import React , { useState }from "react";



function WarningBanner(props){

  const [warning, setWarning] = useState(true);

  if(!warning){

    return null;

  }

  return (

    <div> 경고!</div>

  );

}

export default WarningBanner ;

null을 리턴하여 렌더링되지 않게 한다.

import React from "react";

function WarningBanner(props){

  if(!props.warning){

    return null;

  }

  return (

    <div> 경고!</div>

  );

}

export default WarningBanner ;

import React , { useState }from "react";

import WarningBanner from './WarningBanner';

function MainPage(props){

  const [showWarning, setShowWarning] = useState(false);

  const handleToggleClick = () => {

    setShowWarning(prevShowWarning => !prevShowWarning);

  }

  return (

    <div>

      <WarningBanner warning={showWarning} />

      <button onClick={handleToggleClick}>

        {showWarning ? '감추기' : '보이기'}

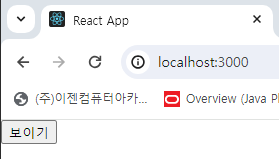
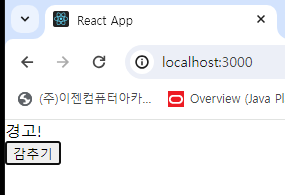
      </button>

    </div>

  );

}

export default MainPage ;

** **

**사용자의 로그인 여부를 나타내는 툴바 컴포넌트**

**환영 메시지를 표시하거나 감추고, 로그인/ 로그아웃 버튼을 보여 주는 역할**

**ToolBar.jsx**

import React from "react";

const styles = {

    wrapper: {

        padding: 16,

        display: "flex",

        flexDirection: "row",

        borderBottom: "1px solid grey",

    },

    greeting: {

        marginRight: 8,

    },

};

function Toolbar(props) {

    const { isLoggedIn, onClickLogin, onClickLogout } = props;

    return (

        <div style={styles.wrapper}>

            {isLoggedIn && <span style={styles.greeting}>환영합니다!</span>}

            {isLoggedIn ? (

                <button onClick={onClickLogout}>로그아웃</button>

            ) : (

                <button onClick={onClickLogin}>로그인</button>

            )}

        </div>

    );

}

export default Toolbar;

import React, { useState } from "react";

import Toolbar from "./Toolbar";

function LandingPage(props) {

    const [isLoggedIn, setIsLoggedIn] = useState(false);

    const onClickLogin = () => {

        setIsLoggedIn(true);

    };

    const onClickLogout = () => {

        setIsLoggedIn(false);

    };

    return (

        <div>

            <Toolbar

                isLoggedIn={isLoggedIn}

                onClickLogin={onClickLogin}

                onClickLogout={onClickLogout}

            />

            <div style={{ padding: 16 }}>소플과 함께하는 리액트 공부!</div>

        </div>

    );

}

export default LandingPage;