**ch 03. Componet 생성**

1. React가 없을 때 소스

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <title>Document</title>

</head>

<body>

    <header>

        <h1>WEB</h1>

        World Wide Web!

    </header>

    <nav>

        <ul>

            <li><a href="1.html">HTML</a></li>

            <li><a href="2.html">CSS</a></li>

            <li><a href="3.html">JavaScript</a></li>

        </ul>

    </nav>

    <article>

        <h2>HTML</h2>

        HTML is HyperText Markup Language.

    </article>

</body>

</html>

1. 컴포넌트 생성

import './App.css';

import React, {Component} from 'react'

// function App() {

//   return (

//     <div className="App">

//       <header className="App-header">

//         <img src={logo} className="App-logo" alt="logo" />

//         <p>

//           Edit <code>src/App.js</code> and save to reload.

//         </p>

//         <a

//           className="App-link"

//           href="https://reactjs.org"

//           target="\_blank"

//           rel="noopener noreferrer"

//         >

//           Learn React

//         </a>

//       </header>

//     </div>

//   );

// }

class App extends Component{

  render(){

    return (

      <div className="App">

        Hello, React !!

        <Subject></Subject>

        <TOC></TOC>

        <Content></Content>

      </div>

    );

  }

}

class Subject extends Component{ // 이부분은 jsx임. create-react-app이 jsx를 js로 변경

  render(){

    return(

      <header>

        <h1>WEB</h1>

        world wide web!

    </header>

    );

  }

}

class TOC extends Component{

  render(){

    return (

      <nav>

      <ul>

          <li><a href="1.html">HTML</a></li>

          <li><a href="2.html">CSS</a></li>

          <li><a href="3.html">JavaScript</a></li>

      </ul>

  </nav>

    );

  }

}

class Content extends Component{

  render(){

    return (

      <article>

        <h2>HTML</h2>

        HTML is HyperText Markup Language.

    </article>

    );

  }

}

export default App;

1. Props

import './App.css';

import React, {Component} from 'react'

// <https://reactjs.org/docs/components-and-props.html> 참조

class App extends Component{

  render(){

    return (

      <div className="App">

        <h1>Hello, React!!</h1>

        <hr/>

        <Subject title="1. WEB" sub="world wide web!!"/>

        <Subject title="2. React" sub="For UI"/>

        <TOC/>

        <Content title="HTML" desc="HTML is HyperText Markup Language."></Content>

      </div>

    );

  }

}

class Subject extends Component{ // 이부분은 jsx임. create-react-app이 jsx를 js로 변경

  render(){

    return(

      <header>

        <h1>{this.props.title}</h1>

        {this.props.sub}

        <hr/>

      </header>

    );

  }

}

class TOC extends Component{

  render(){

    return (

      <nav>

        <ul>

            <li><a href="1.html">HTML</a></li>

            <li><a href="2.html">CSS</a></li>

            <li><a href="3.html">JavaScript</a></li>

        </ul>

        <hr/>

      </nav>

    );

  }

}

class Content extends Component{

  render(){

    return (

      <article>

        <h2>{this.props.title}</h2>

        {this.props.desc}

        <hr/>

      </article>

    );

  }

}

export default App;

* 크롬 웹스토어에서 React Developer Tools 확장 프로그램 설치 : react 상의 컴포넌트를 볼 수 있음

1. 컴포넌트를 파일로 분리하기

* App.js

import './App.css';

import React, {Component} from 'react';

import TOC from "./components/TOC";

import Content from "./components/Content";

import Subject from "./components/Subject";

class App extends Component{

  render(){

    return (

      <div className="App">

        <h1>Hello, React!!</h1>

        <hr/>

        <Subject title="1. WEB" sub="world wide web!!"/>

        <Subject title="2. React" sub="For UI"/>

        <TOC/>

        <Content title="HTML" desc="HTML is HyperText Markup Language."></Content>

      </div>

    );

  }

}

export default App;

* Subject.js

import React, {Component} from 'react';

class Subject extends Component{ // 이부분은 jsx임. create-react-app이 jsx를 js로 변경

    render(){

      return(

        <header>

          <h1>{this.props.title}</h1>

          {this.props.sub}

          <hr/>

        </header>

      );

    }

  }

export default Subject;

* TOC.js

import React, {Component} from 'react'

class TOC extends Component{

    render(){

      return (

        <nav>

          <ul>

              <li><a href="1.html">HTML</a></li>

              <li><a href="2.html">CSS</a></li>

              <li><a href="3.html">JavaScript</a></li>

          </ul>

          <hr/>

        </nav>

      );

    }

  }

export default TOC;

* Content.js

import React, {Component} from "react";

class Content extends Component{

    render(){

      return (

        <article>

          <h2>{this.props.title}</h2>

          {this.props.desc}

          <hr/>

        </article>

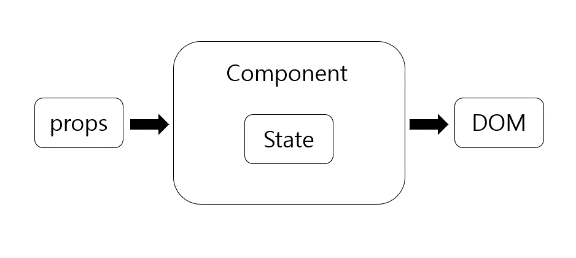
      );

    }

  }

export default Content;

1. State와 key



<compoent props\_name=”props\_value”>

Props : 사용자 입장에서 compoent를 사용할 때 필요한 데이터

State : props값에 따라 내부에서 구현에 필요한 데이터

* App.js

import './App.css';

import React, {Component} from 'react';

import TOC from "./components/TOC";

import Content from "./components/Content";

import Subject from "./components/Subject";

class App extends Component{

  constructor(props){

    super(props);

    this.state = {

      subject : {title : 'WEB', sub:'World Wide Web !!'},

      contents : [

        {id:1, title:'HTML', desc:'HTML is for information'},

        {id:2, title:'CSS', desc:'CSS is for design'},

        {id:3, title:'JavaScript', desc:'JavaScript is for interactive'},

      ],

    }

  }

  render(){

    return (

      <div className="App">

        <h1>Hello, React!!</h1>

        <hr/>

        <Subject

          title={this.state.subject.title}

          sub={this.state.subject.sub}></Subject>

        <Subject title="2. React" sub="For UI"/>

        <TOC data={this.state.contents}/>

        <Content title="HTML" desc="HTML is HyperText Markup Language."></Content>

      </div>

    );

  }

}

export default App;

* TOC.js

import React, {Component} from 'react'

class TOC extends Component{

    render(){

        var lists = [];

        var data = this.props.data;

        for(var i=0 ; i<data.length ; i++){

            lists.push(<li key={data[i].id}><a href={"/content/"+data[i].id+".html"}+>{data[i].title}</a></li>);

        }

        return (

            <nav>

                <ul>

                    {lists}

                </ul>

            </nav>

        );

        // return(

        //     <nav>

        //         <ul>

        //             <li><a href="1.html">HTML</a></li>

        //             <li><a href="2.html">CSS</a></li>

        //             <li><a href="3.html">JavaScript</a></li>

        //         </ul>

        //     <hr/>

        //     </nav>

        // );

    }

}

export default TOC;

1. 이벤트 들어가기 전

import React,{ Component } from 'react';

import './Subject.css';

class Subject extends Component{

  render(){

    console.log('Subject render');

    return (

      <header>

        <h1><a href="/">{this.props.title}</a></h1>

        {this.props.sub}

      </header>

    );

  }

}

export default Subject;

header a{

  text-decoration: none;

}

import './App.css';

import React,{ Component } from 'react';

import Subject from './components/Subject';

import TOC from './components/TOC';

import Content from './components/Content';

class App extends Component{

  constructor(props){ // 컴포넌트 실행전 생성자가 실행되서 초기화 먼저.

    super(props);

    this.state = {

      mode : 'welcome',

      subject : {title:'WEB', sub:'World Wide Web!!'},

      welcome : {title:'WELCOME', desc:'Hello, React!!!'},

      contents : [

        {id:1, title:'HTML', desc:'HTML is for information'},

        {id:2, title:'CSS', desc:'CSS is for design'},

        {id:3, title:'JavaScript', desc:'JavaScript is for interactive'},

      ],

    }

  }

  render(){

    // props나 state가 바뀌면 render가 다시 실행(화면이 다시 그려짐)

    console.log('★App render');

    var \_title, \_desc = null;

    if(this.state.mode === 'welcome'){

      \_title = this.state.welcome.title;

      \_desc  = this.state.welcome.desc;

    }else if(this.state.mode === 'read'){

      \_title = this.state.contents[0].title;

      \_desc = this.state.contents[0].desc;

    }

    console.log('App render');

    return (

      <div className='App'>

        <Subject title={this.state.subject.title}

        sub={this.state.subject.sub}></Subject>

        {/\* 하위컴포넌트(Subject)의 props의 값을 App(상위 컴포넌트)에서의 state로 전달 \*/}

        <TOC data={this.state.contents}></TOC>

        <Content title={\_title} desc={\_desc}></Content>

      </div>

    );

  }

}

export default App;

1. 이벤트 설치하기

import './App.css';

import React, {Component} from 'react';

//import Subject from './components/Subject';

import TOC from './components/TOC';

import Content from './components/Content';

class App extends Component{

  constructor(props){

    super(props)

    this.state = {

      mode : 'read',

      Subject : {title:'WEB', sub:'World Wide Web!'},

      welcome : {title:'Welcome', desc:'Hello, React!!!'},

      contents : [

        {id:1, title:'HTML', desc:'HTML is for information'},

        {id:2, title:'CSS', desc:'CSS is for design'},

        {id:3, title:'JavaScript', desc:'JavaScript is for interactive'},

      ],

    };

  }

  render(){

    console.log('App render');

    var \_title, \_desc = null;

    if(this.state.mode === 'welcome'){

      \_title = this.state.welcome.title;

      \_desc = this.state.welcome.desc;

    }else if(this.state.mode === 'read'){

      \_title = this.state.contents[0].title;

      \_desc = this.state.contents[0].desc;

    }

    return (

      <div className="App">

        <header>

          <h1><a href="/1" onClick={function(e){

            //alert('제목 클릭');

            //console.log(e);

            //debugger;

            e.preventDefault();// 태그의 기본적인 동작방법 막음

//console.log(this);

            // this.state.mode = 'welcome'; // 이러면 안 됨

            this.setState({ // 컴포넌트가 생성된 후 state값을 동적으로 바꿀 때 setState 함수사용

              mode : 'welcome',

            });

          }.bind(this)}>{this.state.Subject.title}</a></h1>

          {this.state.Subject.sub}

          <hr/>

        </header>

        {/\* <Subject title={this.state.Subject.title} sub={this.state.Subject.sub}></Subject> \*/}

        <TOC data={this.state.contents}></TOC>

        <Content title={\_title} desc={\_desc}/>

      </div>

    );

  }

}

export default App;

1. bind 설명

var obj = {name : 'hong'};

function bindTest(){

console.log(this.name);

}

bindTest(); // 결과가 undefind

var bindTest2 = bindTest.bind(obj);

bindTest2(); // 결과가 hong

1. 컴포넌트 이벤트 만들기 Subject

* App.js

import './App.css';

import React, {Component} from 'react';

import Subject from './components/Subject';

import TOC from './components/TOC';

import Content from './components/Content';

class App extends Component{

  constructor(props){

    super(props)

    this.state = {

      mode : 'read',

      subject : {title:'WEB', sub:'World Wide Web!'},

      welcome : {title:'Welcome', desc:'Hello, React!!!'},

      contents : [

        {id:1, title:'HTML', desc:'HTML is for information'},

        {id:2, title:'CSS', desc:'CSS is for design'},

        {id:3, title:'JavaScript', desc:'JavaScript is for interactive'},

      ],

    };

  }

  render(){

    console.log('App render', this);

    var \_title, \_desc = null;

    if(this.state.mode === 'welcome'){

      \_title = this.state.welcome.title;

      \_desc = this.state.welcome.desc;

    }else if(this.state.mode === 'read'){

      \_title = this.state.contents[0].title;

      \_desc = this.state.contents[0].desc;

    }

    return (

      <div className="App">

        {/\* <header>

          <h1><a href="/" onClick={function(e){

            //alert('제목 클릭');

            //console.log(e);

            //debugger;

            e.preventDefault();// 태그의 기본적인 동작방법 막음

            //this.state.mode = 'welcome'; // 이러면 안 됨

            //console.log(this);

            this.setState({

              mode : 'welcome',

            });

          }.bind(this)}>{this.state.Subject.title}</a></h1>

          {this.state.Subject.sub}

          <hr/>

        </header> \*/}

        <Subject title={this.state.subject.title} sub={this.state.subject.sub}

          onChangePage={function(){

            //alert('hi');

            this.setState({

              mode:'welcome'

            });

          }.bind(this)}

        ></Subject>

        <TOC data={this.state.contents} ></TOC>

        <Content title={\_title} desc={\_desc}/>

      </div>

    );

  }

}

export default App;

* Subject.js

import React, {Component} from 'react'

class Subject extends Component{

  render(){

    return (

      <header>

        <h1><a href="/" onClick={function(e){

          //console.log(this);

          e.preventDefault();

          this.props.onChangePage();

        }.bind(this)}>{this.props.title}</a></h1>

        {this.props.sub}

        <hr/>

      </header>

    );

  }

}

export default Subject;

1. 컴포넌트 만들기 TOC

* App.js

import './App.css';

import React, {Component} from 'react';

import Subject from './components/Subject';

import TOC from './components/TOC';

import Content from './components/Content';

class App extends Component{

  constructor(props){

    super(props)

    this.state = {

      mode : 'read',

      selected\_content\_id : 1,

      Subject : {title:'WEB', sub:'World Wide Web!'},

      welcome : {title:'Welcome', desc:'Hello, React!!!'},

      contents : [

        {id:1, title:'HTML', desc:'HTML is for information'},

        {id:2, title:'CSS', desc:'CSS is for design'},

        {id:3, title:'JavaScript', desc:'JavaScript is for interactive'},

      ],

    };

  }

  render(){

    console.log('App render');

    var \_title, \_desc = null;

    if(this.state.mode === 'welcome'){

      \_title = this.state.welcome.title;

      \_desc = this.state.welcome.desc;

    }else if(this.state.mode === 'read'){

      // var content = this.state.contents[this.state.selected\_content\_id-1];

      // \_title = content.title;

      // \_desc = content.desc;

for(var i=0 ; i<this.state.contents.length ; i++){

        var data = this.state.contents[i];

        if(data.id === this.state.selected\_content\_id){

          \_title = data.title;

          \_desc = data.desc;

          break;

        }

      }

    }

    return (

      <div className="App">

        <Subject title={this.state.Subject.title} sub={this.state.Subject.sub}

          onChangePage={function(e){

            //alert('hi');

            this.setState({

              mode:'welcome',

            });

          }.bind(this)}

        ></Subject>

        <TOC

          onChangePage={function(id){

            // alert('hi');

            // console.log(typeof(id));

            this.setState({

              mode:'read',

              selected\_content\_id:Number(id)

            });

          }.bind(this)}

          data={this.state.contents}></TOC>

        <Content title={\_title} desc={\_desc}/>

      </div>

    );

  }

}

export default App;

* TOC.js

import React, {Component} from 'react'

class TOC extends Component{

  render(){

    var lists = [];

    var data = this.props.data;

    for(var i=0 ; i<data.length ; i++){

      lists.push(

        <li key={data[i].id}>

          <a href={"/content/"+data[i].id}

            data-id={data[i].id}

            onClick={function(e){

              e.preventDefault();

              //console.log(e.target.dataset.id);

              this.props.onChangePage(e.target.dataset.id);

            }.bind(this)}>

            {data[i].title}

          </a>

        </li>);

    }

    return (

      <nav>

        <ul>

          {lists}

        </ul>

        <hr/>

      </nav>

    );

  }

}

export default TOC;

* + 위의 소스를 아래와 같이도 가능

class TOC extends Component{

  render(){

    var lists = [];

    var data = this.props.data;

    for(var i=0 ; i<data.length ; i++){

      lists.push(

        <li key={data[i].id}>

          <a href={"/content/"+data[i].id}

            onClick={function(id, e){

              e.preventDefault();

              //console.log(e.target.dataset.id);

              this.props.onChangePage(id);

            }.bind(this, data[i].id)}>

            {data[i].title}

          </a>

        </li>);

    }

    return (

      <nav>

        <ul>

          {lists}

        </ul>

        <hr/>

      </nav>

    );

  }

}

export default TOC;