

# Day 3-1. Props & State & Key (Feb 25, 21)

```
App.js

class App extends Component{
  constructor(props){
    super(props);
    this.state={
      subject:{title:'WEB', sub:'World Wide Web'}
    }
  }
  render() {
    return(
      <div className="App">
        <Subject title={this.state.subject.title} sub={this.state.subject.sub}></Subject>npx
        <Subject title="React" sub="For UI"></Subject>
        <TOC></TOC>
        <Content></Content>
      </div>
    );
  }
}
```

Subject Component의 property 하나에 state기능을 적용한 코드

여러 개의 property를 다룰 때는 어떻게 될까?

```
TOC.js

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>
    )
  }
}
```

TOC안에 있는 data를 App의 내부 state를 TOC에 주입하는 방법을 이용해서 자동으로 data가 바뀌게 되도록 해줄 것이다.

(TOC의 목록이 추가될 때 마다 TOC를 열어서 수정하지 않아도 되도록 한다.)

(TOC의 부모가 가지고 있는 state를 이용해서 TOC의 내부 data가 바뀌도록 한다.)

App.js

```
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">
        <Subject title={this.state.subject.title} sub={this.state.subject.sub}></Subject>npx
        <Subject title="React" sub="For UI"></Subject>
        <TOC data={this.state.contents}></TOC>
        <Content></Content>
      </div>
    );
  }
}
```

TOC.js

```
class TOC extends Component{
  render(){
    var lists = [];
    var data = this.props.data
    var i = 0;
    while(i < data.length){
      lists.push(<li key={data[i].id}><a href={"/content/"+data[i].id}>{data[i].title}</a></li>);
      i = i + 1;
    }
    return (
      <nav>
        <ul>
          {lists}
        </ul>
      </nav>
    )
  }
}
```