

Day 5-4. Component Event_1

(Feb 27, 21)

이제, 목록(HTML, CSS, JavaScript)을 클릭했을 때, App 컴포넌트의 state의 mode를 'read'로 바꾸고, 클릭한 목록의 content가 본문에 나오도록 하는 이벤트를 만들어 보자.

App.js

```
<TOC
  onChangePage={function(){
    this.setState({mode='read'});

  }.bind(this)}
  data={this.state.contents}

></TOC>
```

TOC 태그에 onChangePage함수 세트 설치 & bind

TOC.js

```
class TOC extends Component{
  render(){
    console.log('TOC 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}
            onClick={function(){
              e.preventDefault();
              this.props.onChangePage();
            }.bind(this)}
          >{data[i].title}</a>
        </li>);
      i = i + 1;
    }
  }
}
```

```
return (  
  <nav>  
    <ul>  
      {lists}  
    </ul>  
  </nav>  
)  
}
```

TOC 컴포넌트에 onClick 함수 설치 & bind

여기까지가, 목록(HTML, CSS, JavaScript)을 클릭했을 때, App 컴포넌트의 state의 mode를 'read'로 바꾼 과정.