

Day 8-3. Create_5 #어려움 (Mar 3rd, 21)

이전에 한 것)

onSubmit 이벤트가 발생했을 때, CreateContent 컴포넌트에 onSubmit props를 실행하도록 했다. 그럼 App 컴포넌트의 onSubmit 함수가 실행된다.

앞으로 할 것)

App컴포넌트의 contents의 끝에 data를 추가한다.

```
App.js

class App extends Component {
  constructor(props){
    super(props);
    this.max_content_id = 3; # max_content_id값 초기화
    this.state = {
      mode:'create',
      subject:{title:'WEB', sub:'World Wid 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, _article, _desc = null;
    if(this.state.mode === 'welcome'){
      _title = this.state.welcome.title;
      _desc = this.state.welcome.desc;
      _article = <ReadContent title = {_title} desc = {_desc}></ReadContent>
    } else if(this.state.mode === 'read'){
      var i = 0;
      while(i < this.state.contents.length){
        var data = this.state.contents[i];
        if(data.id === this.state.selected_content_id){
          _title = data.title;
          _desc = data.desc;
          break;
        }
        i = i + 1;
      }
      _article=<ReadContent title = {_title} desc={_desc}></ReadContent>
    } else if (this.state.mode === 'create'){
      _article = <CreateContent onSubmit={function(_title, _desc){
        // add content to this.state.contents
        this.max_content_id = this.max_content_id +1; # onSubmit이 실행될 때
                                                    # max_content_id 값 증가

        this.state.contents.push(
          {id:this.max_content_id, title:_title, desc:_desc} # 입력받은 data contents에 추가
        )
      }}>
    }
  }
}
```

```

    });
    this.setState({
      contents: this.state.contents          # setState로 바뀐 this.state.contents 업데이트
    });
  }.bind(this)}></CreateContent>
}
return (
  <div className="App">
    <Subject
      title={this.state.subject.title}
      sub={this.state.subject.sub}
      onChangePage={function(){
        this.setState({mode: 'welcome'});
      }.bind(this)}
    >
    </Subject>
    <TOC
      onChangePage={function(id){
        this.setState({
          mode: 'read', selected_content_id: Number(id)});
      }.bind(this)}
      data={this.state.contents}
    ></TOC>
    <Control onChangeMode={function(_mode){
      this.setState({
        mode: _mode
      });
    }.bind(this)}></Control>
    {_article}
  </div>
);
}
}
export default App;

```

콘텐츠를 추가할 때, 원래의 data를 변경하는 push 함수대신, 원래의 data는 보존하고 새로운 리스트를 만들어서 추가하는 concatenation을 사용하는 것을 추천한다.

```

App.js

}
_article=<ReadContent title = {_title} desc={_desc}></ReadContent>
} else if (this.state.mode === 'create'){
  _article = <CreateContent onSubmit={function(_title, _desc){
    // add content to this.state.contents
    this.max_content_id = this.max_content_id +1; # onSubmit이 실행될 때
                                                    # max_content_id 값 증가
    //      this.state.contents.push(
    //        {id:this.max_content_id, title:_title, desc:_desc} # 입력받은 data contents에 추가
    //      );
    this.setState({
      contents: this.state.contents          # setState로 바뀐 this.state.contents 업데이트
    });
  }.bind(this)}></CreateContent>
}
}

```

그래서, this.state.contents.push 대신에,

```
App.js

    }
    _article=<ReadContent title = {_title} desc={_desc}></ReadContent>
  } else if (this.state.mode === 'create'){
    _article = <CreateContent onSubmit={function(_title, _desc){
      // add content to this.state.contents
      this.max_content_id = this.max_content_id +1; # onSubmit이 실행될 때
                                                    # max_content_id 값 증가
//      this.state.contents.push(
//        {id:this.max_content_id, title:_title, desc:_desc} # 입력받은 data contents에 추가
//      );

      var _contents = this.state.contents.concat(
        {id:this.max_content_id, title:_title, desc:_desc}
      )
      this.setState({
        //contents:this.state.contents          # setState로 바뀐 this.state.contents 업데이트
        contents:_contents
      });
    }.bind(this)}></CreateContent>
  }
}
```

concat을 사용해주고, 새로운 data 리스트를 setState 해준다.