

Day 8-2. Create_4 (Mar 3rd, 21)

앞으로 할 것)

submit 버튼을 사용자가 클릭했을 때, onSubmit 이벤트가 실행될 것인데, onSubmit 이벤트가 실행되었을 때,

CreateContent 컴포넌트를 가져다 쓰고있는 App 컴포넌트의 'contents' data에 입력한 data를 추가하는 작업을 할 것이다.

이를 위해서 필요한 것은 submit 버튼을 클릭했을 때, CreateContent에 이벤트로 설치된 함수를 실행시켜야 한다.

```
CreateContent.js

import React, { Component } from 'react';

class CreateContent extends Component{
  render(){
    console.log('Content render');
    return (
      <article>
        <h2>create</h2>
        <form action="/create_process" method="post"
          onSubmit={function(){
            e.preventDefault();
          }.bind(this)}}
        >
          <p><input type="text" name="title" placeholder="title"></input></p>
          <p>
            <textarea name="desc" placeholder="description"></textarea>
          </p>
          <p>
            <input type="submit"></input>
          </p>
        </form>
      </article>
    );
  }
}

export default CreateContent;
```

>>CreateContent에 이벤트로 설치된 함수

App.js에 onSubmit 이벤트를 만든다. 이벤트가 발생할 때 _title과 _desc를 입력받는 함수를 실행하여 setState를 통해서 App의 contents에 data를 추가하도록 한다.

App.js

```
} else if (this.state.mode === 'create'){
  _article = <CreateContent onSubmit={function(_title, _desc){
    // add content to this.state.contents
  }.bind(this)}></CreateContent>
}
```

CreateContent의 onSubmit props를 호출하고,

App.js

```
<form action="/create_process" method="post"
  onSubmit={function(){
    e.preventDefault();
    this.props.onSubmit();
  }.bind(this)}
>
```

_title과 _desc를 가져오도록 한다.

App.js

```
<form action="/create_process" method="post"
  onSubmit={function(){
    e.preventDefault();
    this.props.onSubmit(
      e.target.title.value,
      e.target.desc.value
    );
  }.bind(this)}
>
```