React

Coopera extension

김낙현 김혜연

Coopera 타임라인

- 글 보여주기
- 글 쓰기
  - ref, refs
  - props를 통한 메소드 전달
  - InnerHTML 설정
  - ReactDOM에서 HTML DOM, jQuery 객체 변환
- 좋아요
- 댓글

ref

- 자식 컴포넌트의 이름을 지정하기 위한 어트리뷰트

```
var Post = React.createClass({  //게시글 쓰기

    render : function(){

        return (
            <div id='post'>
                <input type='text' ref='pck' id='postPck' placeholder='패키지명' />
                <textarea type='text' ref='info' id='postInfo' placeholder='내용 입력' />
                <button type='submit' name='submit' id='postSubmit' onClick={this.postCard}>게시</button>
            </div>
        );
    }
```

this.refs

```
var Post = React.createClass({  //게시글 쓰기

    postCard : function(){
        var self = this,
            pck = ReactDOM.findDOMNode(this.refs.pck),
            info = ReactDOM.findDOMNode(this.refs.info);

        if(pck == '' || info == '') alert("입력값이 없습니다.");
        else {
            //패키지명과 내용을 올림
            //data : 새로 올라간 글 {no, code, info}
            $.ajax({
                type:'POST',
                url: './coopera.php',
                data:{'cmd' : 'up_pck',
                    'pck' : pck.value,
                    'info' : info.value}
            }).done(function(data){
                pck.value='';
                info.value='';
                self.props.onPost(data);
            });
        }
    },
```

이름 붙인

자식 컴포넌트를

참조하기 위한 속성

props로 메소드 전달

```
var Timeline = React.createClass({

    onPost: function(newPck){
        ...
    },

    render: function(){
        ...
        return(
            <div>
                <Post ref='post' onPost={this.onPost} />
                <div ref='cards' className='cards'>{result}</div>
            </div>

        );
    }
```

```
var Post = React.createClass({  //게시글 쓰기

    postCard : function(){
        ...
            $.ajax({
                ...
            }).done(function(data){
                ...
                self.props.onPost(data);
            });
        }
    },
```

InnerHTML

```
var CardInfo = React.createClass({  //패키지 우클릭시 나오는 기본정보
    render : function(){
        var info = this.props.info;

        return (
                <div className='cardInfo' dangerouslySetInnerHTML={{__html: info.info}}>
                </div>
        );
    }
});
```

- React는 자체적으로 XSS 방지
- dangerouslySetInnerHTML 어트리뷰트 사용
- 이름부터 위험

React DOM 변환

```
var Timeline = React.createClass({

    onPost: function(newPck){
        var pcks=this.state.packages,
            newpcks=[newPck],
            post = ReactDOM.findDOMNode(this.refs.post),
            cards = ReactDOM.findDOMNode(this.refs.cards);

        for(var i=0;i<pcks.length;i++) newpcks.push(pcks[i]);

        $(post).css("display","none");
        this.setState({packages: newpcks});

        $(cards).css("margin-top","50px");
        $(cards).animate({"margin-top":"320px"},1000,function(){
            $(post).css({"display":"block", "opacity":"0"}).animate({"opacity":"1"},1000);
            $(cards).css("margin-top","20px");
        });

    },
```

- ReactDOM.findDOMNode(ReactComponent)
- 리액트 컴포턴트를 HTML DOM으로 변환
- $(변환한 HTML DOM)을 통해 jQuery도 이용가능
- 컴포넌트 추상화를 파괴하여 권장되는 사용방법이 아님

시연