학습결과서

이름 : 이승현

날짜 : 2020.07.04(토)

**오늘 학습 한 내용**

JavaScript for Web Browser (생활코딩)

**- jQuery Object – map()**

map 메소드는 제공된 배열 데이터를 변환시켜 다시 새로운 배열 객체를 반환(returns a new array)한다.

jQuery.map(array, callback(elementOfArray, indexInArray) )

- array : 변환할 배열  
- callback(elementOfArray, indexInArray) 모든 요소에 대해 실행될 함수.  
 첫번째 인자는 배열 요소(item), 두번째 인자는 배열 인덱스

    var li = $('li');

    li.map(function(*index*,*elem*){

        $(*elem*).css('color','red';)

    })

.map(callback(index,elem)) : 콜백함수의 this는 현재 DOM를 참조합니다. 함수는 각각의 데이터 값을 가지고 오며, 데이터 값은 배열에 저장됩니다. 만약 null이나 undefined 값이 나온다면 각각의 데이터 값을 제대로 가지고 오지 않은 것입니다.

리액트를 다루는 기술

**- props 검증 : propTypes**

컴포넌트의 필수 props를 지정하거나 props 타입(type)를 지정할 때는 propTypes를 사용합니다.

import PropTypes from 'prop-types';

코드 위쪽에 propTypes를 불러옵니다.

MyComponent.propTypes={

    name : PropTypes.string,

    age : PropTypes.number.isRequired

}

isRequired를 이용하여 필수 propTypes를 지정합니다. 해당 부분의 prop값이 입력되지 않으면 오류를 출력합니다.

**propTypes 종류**

array / bool / func / number / object / string

symbol – ES6 문법의 심벌 개체

node - 랜더링 할 수 있는 모든 것 (숫자, 문자여, element 또는 이들로 구성된 배열)

element – 리액트요소

instanceOf(Myclass) – 특정 클래스의 인스턴스

oneOf([‘Male’, ‘Female’]) – 주어진 배열 요소 중 하나

oneOfType([React.PropTypes.string, React.PropTypes.number]) : 주어진 배열 안의 종류 중 하나

arrayOf(React.PropTypes.number) : 주어진 종류로 구성된 배열

objectOf(React.PropTypes.number) : 주어진 종류의 값을 가진 객체

shape({name : React.PropTypes.string, age : React.PropTypes.number}) : 주어진 스키마를 가진 객체

any : 아무종류

**-리액트의 이벤트 시스템**

이벤트 사용시 주의사항

1. 이벤트 이름은 camelCase로
2. 이벤트에 실행할 자바스크립트 코드를 전달하는 것이 아니라, 함수형태의 값을 전달
3. DOM 요소에만 이벤트를 설정할 수 있음.

                <input

*type*="text"

*name*="message"

*placeholder*="아무거나 입력해보세요"

*value*={this.state.message}

*onChange*={this.handleChange}

*onKeyPress*={this.handleKeyPress}

                />

이러한 input태그가 있을때, enter를 눌러서 값이 전달되게끔 구현하려고 하면, onKeyPress를 아래와 같이 작성하는데,

    handleKeyPress = (*e*) => {

        console.dir(*e*.keyCode)

        if(*e*.keyCode === 13 ){

            this.handleClick();

        }

    }

일반 자바스크립트에서는 keyCode값으로 Enter를 구현하는 것이 되었었는데, Node환경에서는 e.keyCode 값은 0만 출력됩니다. 이때문에 e.keyCode값으로 하지않고, e.key값으로 구현해서 해결

    handleKeyPress = (*e*) => {

        if(*e*.key === 'Enter' ){

            this.handleClick();

        }

    }

**- ref**

ref는 DOM을 꼭 직접적으로 건드려야 할 때 사용되는데, 이때 id를 사용하지 않는 이유는 같은 컴포넌트를 여러번 사용할 경우, HTML에서 유일해야할 id가 중복적으로 생성될 수 있기 때문입니다. ref는 전역적으로 작동하지 않고 컴포넌트 내부에서만 작동하기 때문에 이런 문제가 발생하지 않습니다.

ref사용법 ref={(ref) => {this.myComponent=ref}}

**오늘 알게된 사실**

- jQuery Object – map() 이 두가지 방법으로 동작한다는 것을 알게 되었습니다.  
특히, 배열을 만드는 map이 아닌DOM의 데이터값을 가져오는건 처음 보았습니다.

- React에서 propTypes과 이벤트, ref에 대해 공부하였습니다.

**내가 부족하다고 느낀 부분**

- 알고리즘 푸는데 시간이 오래걸렸습니다.

**수업에 바라는 점**

-