

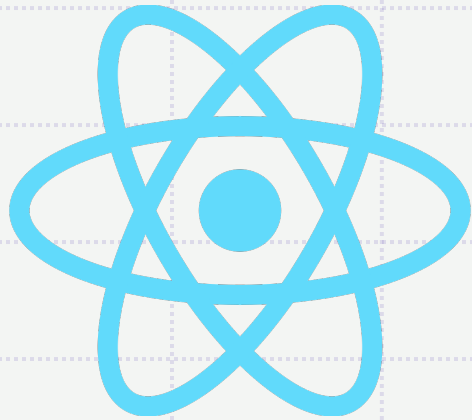
React Hooks

- _ useState
- _ useEffect
- _ useMemo

9/16 Frontend Session 01

박고은 장혜원

React Component



React



component

리액트로 만들어진 앱을 이루는 최소한의 단위

데이터(props)를 입력받아 **state 상태**에 따라
DOM Node를 출력하는 함수

```
function component1(매개변수 - props로서 데이터 받는 곳){  
  state 상태 초기화 및 정의 가능한 함수 설정  
  return(  
    DOM NODE 반환 (jsx 문법 등을 통해 DOM NODE 작성)  
  );  
};
```

```
function component1() {  
  const [loggedIn, setLoggedIn] = useState(false);  
  return (  
    <div>{loggedIn ? '로그인' : '로그아웃'}</div>  
  );  
};
```

React State

component



state

리액트로 만들어진 앱을 이루는 최소한의 단위

데이터(props)를 입력받아 **state 상태**에 따라
DOM Node를 출력하는 함수

React Component에서 **동적**인 값

Component의 렌더링 결과물에
영향을 주는 데이터를 갖고 있는 객체

함수 내에 선언된 변수처럼
컴포넌트 안에서 관리


```
function component1() {  
  const [loggedIn, setLoggedIn] = useState(false);  
  return (  
    <div>{loggedIn ? '로그인' : '로그아웃'}</div>  
  );  
};
```

React Hooks(1)

_useState

React Hooks(1)_useState란?

State



useState

함수 내에 선언된 변수처럼
컴포넌트 안에서 관리되는 동적인 데이터 값인
state의 경우,

동적으로 바뀌는 만큼
변경될 시 상태를 관리하는 도구가 필요하다

함수형 컴포넌트에서
상태를 관리해주는
React Hook

React Hooks(1)_useState 사용법

React Package에서 useState 훅을 불러온다.

```
import React, {useState} from 'react';
```

아래와 같은 형태로 useState 훅을 호출한다.

```
const [상태 값 저장 변수, 상태 값을 갱신해주는 Setter함수]  
= useState(상태의 초기값);
```

-> 위와 같이 useState 함수를 호출해야 하는 이유

: useState 함수를 호출하면 배열을 반환하는데,

반환하는 배열의 첫번째 원소는 현재 상태 값 변수이고,

두번째 원소는 상태 값을 갱신해주는 Setter 함수이다.

: useState 괄호 안의 값은 상태의 초기 값으로,

첫번째 원소인 상태 값 변수의 초기값이라고 이해하면 된다.

Setter 함수명 = set + 상태 값 변수명
(상태 값 변수명의 첫번째 문자는 대문자로 작성해야 한다.)

React Hooks(1)

_useState 실습

React Hooks(2)

_useEffect

React Hooks(2)_useEffect란?

useEffect

React Component가 **rendering** 될 때마다 특정 작업을 실행할 수 있도록 하는 Hook

렌더링(Rendering)이란?

: HTML, CSS, JavaScript 등 개발자가 작성한 문서가 브라우저에서 **출력**되는 과정

렌더링(Rendering)은 언제 일어날까?

: Component가 **생성(mount)**됐을 때, **업데이트(update)**됐을 때, **제거(unmount)**됐을 때 등등

React Hooks(2)_useEffect 사용법

React Package에서 useEffect 훅을 불러온다.

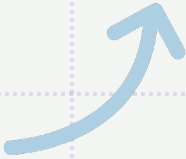
```
import React, {useEffect} from 'react';
```

아래와 같은 형태로 useEffect 훅을 호출한다.

```
useEffect( function, deps)
```

- **function** : 렌더링 시 수행하고자 하는 작업
- **deps** : 배열 형태, 배열 안에는 검사하고자 하는 특정 값 or 빈 배열

deps에 따라
useEffect 함수의 효과가 달라진다.



React Hooks(2)_useEffect 효과

deps에 따른 useEffect 효과 `useEffect(function, deps)`

< deps가 **생략**될 경우>

: Component가 **리렌더링(rendering)**될 때마다 function 실행

< deps가 **빈리스트[]**일 때 >

: Component가 mount 됐을 때 (생성 시), function 실행
즉, 컴포넌트가 **화면에 가장 처음 렌더링 될 때** 한 번만 실행

< deps가 **특정 props나 state**일 때 >

: Component의 특정 props나 state가
생성(mount)되거나 **업데이트(update)**될 때마다 function 실행

```
// deps가 생략될 경우  
useEffect(()=>{  
  // 실행하고자 하는 코드  
});
```

```
// deps가 빈리스트[]일 때  
useEffect(()=>{  
  // 실행하고자 하는 코드  
}, []);
```

```
// deps가 특정 props나 state일 때  
useEffect(()=>{  
  // 실행하고자 하는 코드  
}, [특정 props나 state]);
```


React Hooks(2)_useEffect 효과

Q :

Component가 사라질(unmount될) 때,
function을 실행하고 싶다면 어떻게 해야 할까?

A :

useEffect(function, deps)에서 두 번째 파라미터인 **deps**는 빈 배열([])을 넣고,
첫 번째 파라미터인 function(실행하고자 하는 함수)의 **return값으로**
cleanup 함수(Component가 사라질 때 실행하고자 하는 함수)를 설정해야 한다.

* **cleanup 함수** : return 뒤에 나오는 함수로, useEffect에 대한 뒷정리함수

```
useEffect(()=>{  
  return(  
    //Cleanup 함수 (Component 제거 시, 실행하고픈 함수)  
  )  
}, [])
```

React Hooks(2)

`useEffect` 실습

React Hooks(3)

_useMemo

React Hooks(3)_useMemo란?

useMemo

Memoization을 통해 컴포넌트 최적화를 돕는 Hook

Memoization이란?

: 동일한 값을 리턴하는 함수를 반복적으로 호출해야 할 때 **맨 처음 계산한 값을 저장**해두고, 다음 호출부터는 계산하지 않고 **메모리에서 꺼내서 재사용**하는 기법

즉, 기존에 수행한 연산의 결과값을 어딘가에 저장해두고 동일한 입력이 들어오면 **재활용**해 최적화

React Hooks(3)_useMemo 사용법

React Package에서 useMemo 훅을 불러온다.

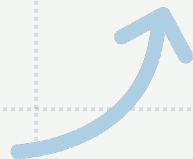
```
import React, {useMemo} from 'react';
```

아래와 같은 형태로 useMemo 훅을 호출한다.

```
useMemo( 콜백함수, 배열)
```

- 콜백함수 : memoization할 값을 계산해서 리턴하는 함수
- 배열: 배열 안의 요소 값이 업데이트 될때만 콜백함수를 호출해서 memoization된 값을 업데이트하고 다시 memoization

다른 함수의 인자로
사용되는 함수



React Hooks(3)_useMemo의 효과

useMemo

UseMemo(함수, 배열)

: 콜백함수를 통해 memoization된 “값”을 재사용해 컴포넌트 최적화

useCallback

UseCallback(함수, 배열)

: 배열의 값이 변할때만 함수를 반환하는 hook

-> 함수 컴포넌트에서 불필요하게 함수 업데이트 하는 것을 방지

: memoization된 “함수”를 반환해 컴포넌트 최적화

React Hooks(3)

`useMemo` 실습