React 컴포넌트 생명주기(Component Lifecycle)

React 컴포넌트는 **생성(Mounting)**, **업데이트(Updating)**, **제거(Unmounting)**의 세 가지 주요 단계로 구성된 생명주기를 가지고있다.

생성(Mounting)

컴포넌트가 **처음 DOM에 렌더링될 때 실행되는 과정**.

- 관련 메서드:
  - `constructor` (클래스형 컴포넌트)
  - `render`
  - `componentDidMount` (클래스형 컴포넌트)
  - `useEffect` (함수형 컴포넌트)

작업:

- 상태 초기화, 초기 렌더링, API 호출 등 초기 작업을 수행.

업데이트(Updating)

컴포넌트가 **상태(state)나 속성(props) 변경으로 다시 렌더링되는 과정**.

- 관련 메서드:
  - `shouldComponentUpdate` (클래스형 컴포넌트)
  - `render`
  - `componentDidUpdate` (클래스형 컴포넌트)
  - `useEffect` (함수형 컴포넌트)

작업:

- 상태나 속성 변경에 따라 DOM을 다시 렌더링하거나 로직 실행.

제거(Unmounting)

컴포넌트가 **DOM에서 제거될 때 실행되는 과정**.

- 관련 메서드:
  - `componentWillUnmount` (클래스형 컴포넌트)
  - `useEffect`**의 cleanup 함수** (함수형 컴포넌트)

작업:

- 타이머 정리, 이벤트 리스너 제거, 구독 취소 등 정리 작업 수행.

```
useEffect(() => {
  // Mount 시 실행 (초기 작업)

  return () => {
    // Unmount 시 실행 (정리 작업)
```

```
  };
}, [의존성 배열]); // 의존성 배열로 Update 조건 제어
```

예제코드

클래스형 컴포넌트

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

class MyComponent extends Component {
  constructor(props) {
    super(props);
    this.state = { count: 0 };
  }

  componentDidMount() {
    console.log('컴포넌트 생성');
    // 초기 데이터 로드
  }

  componentDidUpdate(prevProps, prevState) {
    console.log('컴포넌트 업데이트');
    // 상태 변경 후 작업
  }

  componentWillUnmount() {
    console.log('컴포넌트 제거');
    // 정리 작업
  }

  render() {
    console.log('렌더링됨');
    return (
      <div>
        <p>Count: {this.state.count}</p>
        <button onClick={() => this.setState({ count: this.state.count + 1 })}>
          Increment
        </button>
      </div>
    );
  }
}

export default MyComponent;
```

함수형 컴포넌트

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

const MyComponent = () => {
  const [count, setCount] = useState(0);

  useEffect(() => {
    console.log('컴포넌트 생성과 컴포넌트 업데이트');
    // Mount와 상태 변경 시 실행
```

```
    return () => {
      console.log('컴포넌트 제거');
      // 정리 작업
    };
  }, [count]); // count가 변경될 때 실행 (update와 직결)

  return (
    <div>
      <p>Count: {count}</p>
      <button onClick={() => setCount(count + 1)}>Increment</button>
    </div>
  );
};

export default MyComponent;
```