Ask Company

리액트와 함께 장고 시작하기 / 리액트

속성값

여러분의 파이썬/장고 페이스메이커가 되겠습니다.

속성값

컴포넌트 생성 시에 넘겨지는 값의 목록

읽기 전용으로 취급하고, 변경하지 않습니다.

자식 컴포넌트 입장에서는 데이터/함수를 전달받는 유일한 통로 (But, Context API로 인해, 새로운 통로가 생겼어요.)

부모 컴포넌트의 데이터/함수를 자식 컴포넌트에게 넘겨주게 됩니다.

컴포넌트는 HOC (High Order Components) 기법을 통해, Redux의 값이나 함수를 넘겨 받기도 합니다.

값 지정 시에 중괄호를 통해 다양한 타입의 값 및 표현식 지정 가능.

중괄호를 빼면, 문자열 타입의 값만 지정 가능

```
<div>
  <Counter color="red" size="10" />
  <Counter color={"green"} size={10} />
  <Counter color="blue" size={"10"} />
</div>
```

속성값으로부터 상탯값 정의하기

```
class CustomComponent extends React.Component {
    state = {
        messageLength: this.props.message.length,
        counter: 0,
    };
```

초기에만 message 속성값을 참조.
message 속성값 변경에는 반영하지 못합니다.

경우에 따라 3가지를 적절히 활용

state는 아니지만, 변경되는 props값에 의존적인 속성을 정의. 변경 불가.

```
get messageLength() {
  return this.props.message.length;
}
```

state는 아니지만, 이를 함수로서 대응. 변경 불가

```
getMessageLength() {
  return this.props.message;
}
```

render 호출 직전에 속성값으로부터 상탯값을 계산하여 반영 상탯값으로 반영이 되기에 필요시에 변경도 가능.

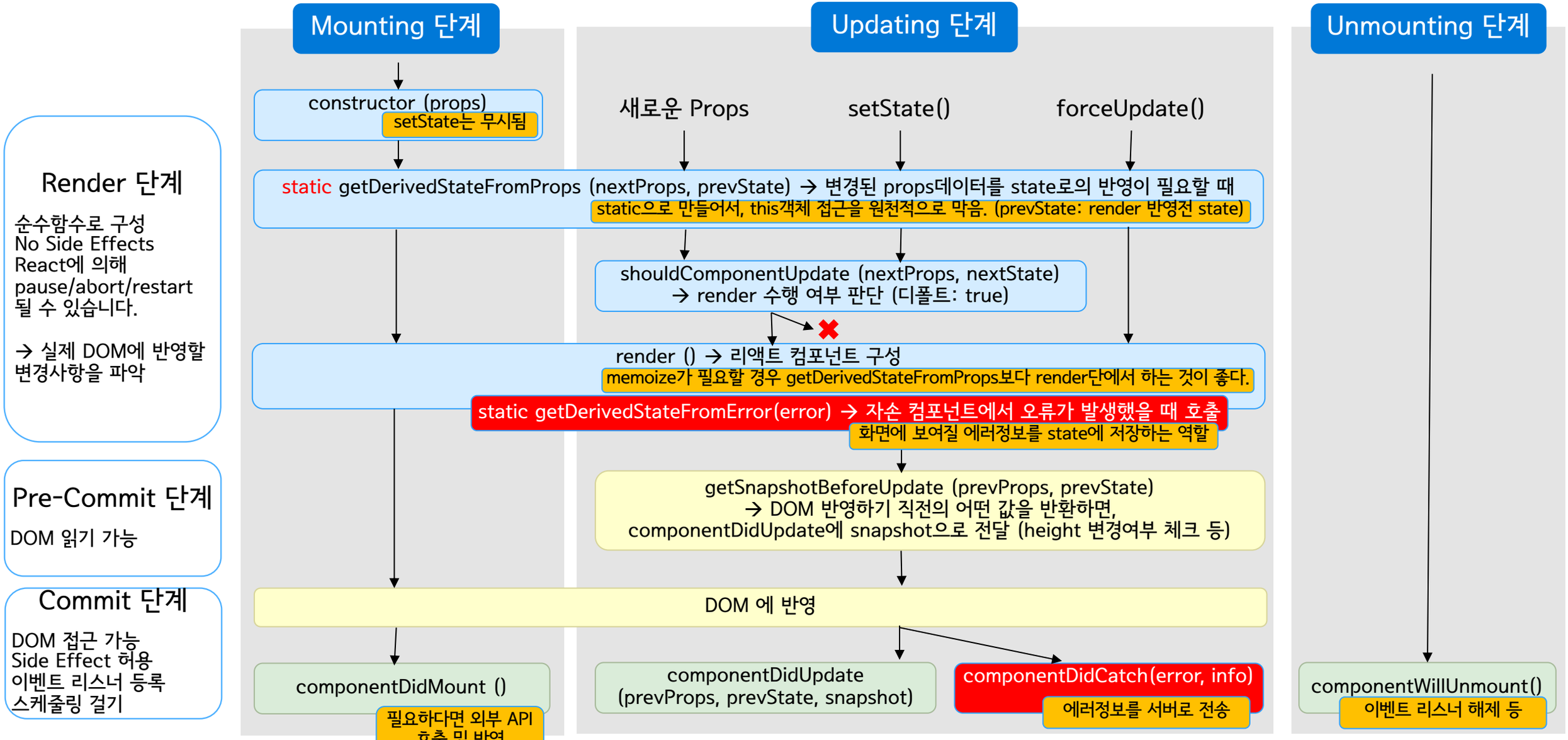
```
state = {
  messageLength: 0
}

// render 메서드 호출 직전에 호출
static getDerivedStateFromProps(props, state) {
  return {
    messageLength: props.message.length
  };
}
```

반영할 상탯값이 없을 경우, null을 반환

참고) 클래스 컴포넌트, 생명 주기

"클래스 컴포넌트, 성명주기" 에피소드에서 자세히 다룹니다.



http://projects.wojtekmaj.pl/react-lifecycle-methods-diagram/

속성값이 변경될 때, API 호출하기

getDerivedStateFromProps는 정적 메서드이기에 this 객체에 접근 불가

- 속성값 → 상탯값 루틴에 집중하도록, 원천적인 봉쇄

그렇기에 componentDidUpdate(props)에서 처리합니다.

```
componentDidUpdate(prevProps) {
  const { postId } = this.props;
  if ( prevProps.postId !== postId ) {
    this.requestPost(postId);
  }
}
```

함수형 컴포넌트에서는 useEffect 훅을 활용하여,
속성값/상탯값이 변경될 때 원하는 함수를 호출할 수 있습니다.

Life is short.
You need Python and Django.

I will be your pacemaker.