



# 進捗状況

React Hook

B4 小林紹子

September 26, 2025

# 目次

## 1 React Hook

# useCallback

- useCallback は関数をメモ化するためのフック.
- useCallback の第一引数は関数で, 第二引数は依存配列.
- 関数の再描画が行われる際に, useCallback() は依存配列の中の値を比較.
- 依存配列の値がそれぞれ同じ場合は, useCallback() はメモ化している関数を返す.
- 異なる場合は, 現在の第一引数の関数をメモに保存する. —→ 新しい関数を返す.

# memo 関数と useCallback の関係

- memo 関数
  - props が同じであれば再描画をスキップする.
- useCallback
  - 関数の参照をメモ化.
  - 依存配列が変わらない限り, 同じ関数オブジェクトを再利用.
  - 再レンダーされても「新しい関数が作られる」ことはない.

————→useCallback と memo 関数を同時に使うことで再描画を防げる.

```
24 //DoubleButtonはメモ化した関数コンポーネントでボタンを表示する
25 const DoubleButton = React.memo((props: ButtonProps) => {
26   const { onClick } = props;
27   console.log('DoubleButtonが再描画されました');
28
29   return <button onClick={onClick}>DoubleButton</button>;
30 });
```

```
42 //useCallbackを使って関数をメモ化する
43 const double = useCallback(() => {
44   setCount((c) => c * 2);
45
46   //第二引数は空行列なのでuseCallbackは常に同じ関数を返す
47   }, []);
48
49 return (
50   <div>
51     <p>Count : {count}</p>
52     <DecrementButton onClick={decrement} />
53     <IncrementButton onClick={Increment} />
54     <DoubleButton onClick={double} />
```