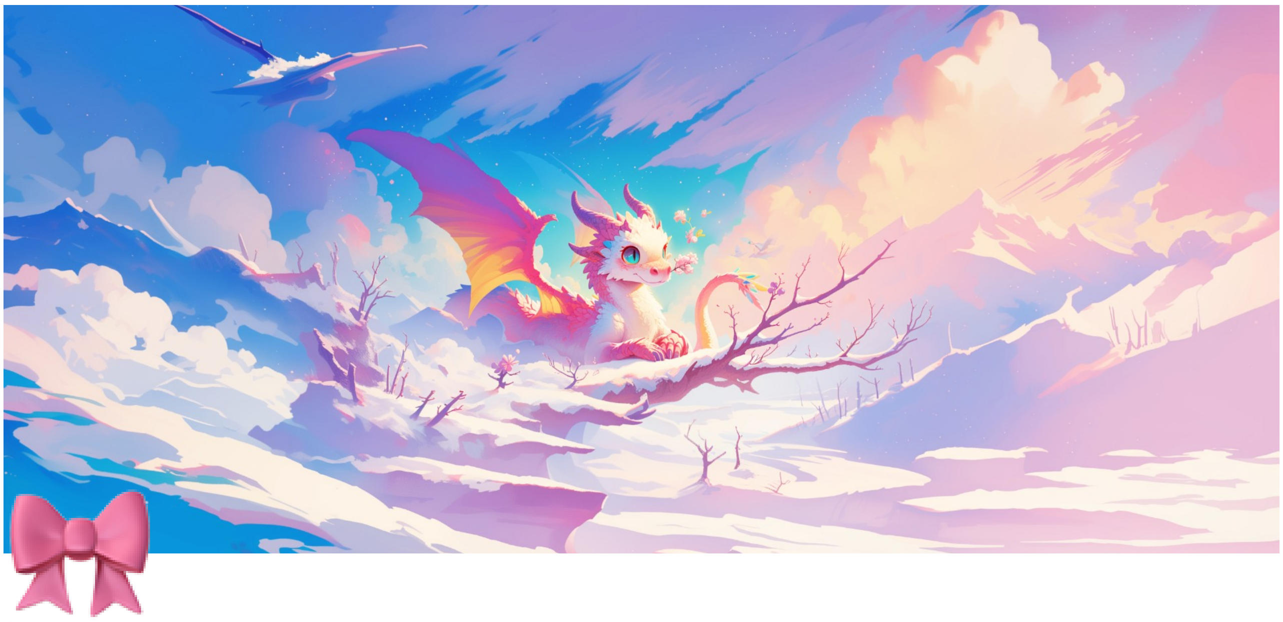

12-6 如何移动 DOM 节点

TypeScript

```
function FunctionComponent() {
  const [count1, setCount1] = useReducer((x) => x + 1, 1);
  // const arr = count1 % 2 === 0 ? [0, 1, 2, 3, 4] : [0, 1, 2, 3];
  const arr = count1 % 2 === 0 ? [0, 1, 2, 3, 4] : [0, 1, 2, 4];

  // const arr = count1 % 2 === 0 ? [0, 1, 2, 3, 4] : [3, 2, 0, 4, 1];

  // old 0, 1, 2, 4
  // new 0, 1, 2, 3, 4
  // 1个before 4

  // old 3, 2, 0, 4, 1
  // new 0, 1, 2, 3, 4
  // 3个before null

  // 0 删除
  return (
    <div className="border">
      <h3>函数组件</h3>
      <button
        onClick={() => {
          setCount1();
        }}
      >
        {count1}
      </button>
      <ul>
        {arr.map((item) => (
          <li key={"li" + item}>{item}</li>
        ))}
      </ul>

      {/* {count1 % 2 === 0 ? (
        <button
          onClick={() => {
            setCount1();
          }}
        >
          {count1}
        </button>
      ) : (
        <span
          onClick={() => {
            setCount1();
          }}
        >
          react
        </span>
      )} */}
    </div>
  );
}
```

commit 阶段

packages/react-reconciler/src/ReactFiberCommitWork.ts

TypeScript

```
function commitPlacement(finishedWork: Fiber) {
  if (finishedWork.stateNode && isHost(finishedWork)) {
    // finishedWork是有dom节点
    // 找domNode的父DOM节点对应的fiber
    const parentFiber = getHostParentFiber(finishedWork);

    let parentDom = parentFiber.stateNode;

    if (parentDom.containerInfo) {
      // HostRoot
      parentDom = parentDom.containerInfo;
    }

    // 1. 遍历fiber结构，检查在老的DOM页面上，finishedWork是否有下一个兄弟DOM
    const before = getHostSibling(finishedWork);
    // 2. 如果有，那么finishedWork对应的的DOM要通过insertBefore放到兄弟DOM的
    // 否则的话，依然通过appendChild存放到parentDom的子节点最后
    insertOrAppendPlacementNode(finishedWork, before, parentDom);
  } else {
    // Fragment
    let kid = finishedWork.child;
    while (kid !== null) {
      commitPlacement(kid);
      kid = kid.sibling;
    }
  }
}
```

返回 fiber 的下一个兄弟 dom 节点

HostComponent 或 HostText

packages/react-reconciler/src/ReactFiberCommitWork.ts

TypeScript
```typescript
function getHostSibling(fiber: Fiber) {
  let node = fiber;

  sibling: while (1) {
    while (node.sibling === null) {
      // 如果检查
      if (node.return === null || isHostParent(node.return)) {
        return null;
      }
      node = node.return;
    }

    node = node.sibling;

    while (!isHost(node)) {
      if (node.flags & Placement) {
        // Placement表示节点是新增插入或者移动位置。而要寻找的是老的稳定位置的DO
        continue sibling;
      }

      if (node.child === null) {
        // 找到最后了。继续下一轮查询
        continue sibling;
      } else {
        // 否则的话，继续查找子节点
        node = node.child;
      }
    }
    if (!(node.flags & Placement)) {
      return node.stateNode;
    }
  }
}
```

新增插入 | 位置移动

packages/react-reconciler/src/ReactFiberCommitWork.ts

TypeScript

```
// 新增插入 | 位置移动
// insertBefore | appendChild
function insertOrAppendPlacementNode(
  node: Fiber,
  before: Element,
  parent: Element
) {
  if (before) {
    parent.insertBefore(getStateNode(node), before);
  } else {
    parent.appendChild(getStateNode(node));
  }
}
```