



16-4 后代组件消费的三种方式

1. useContext

只能用在函数组件或者自定义 Hook 中。

DebugReact > src > react > packages > react-reconciler > src > JS ReactFiberHooks.js >

```
3481  const HooksDispatcherOnMount: Dispatcher = {
3482    readContext,
3483
3484    use,
3485    useCallback: mountCallback,
3486    useContext: readContext,
3487    useEffect: mountEffect,
3488    useImperativeHandle: mountImperativeHandle,
3489    useLayoutEffect: mountLayoutEffect,
3490    useInsertionEffect: mountInsertionEffect,
3491    useMemo: mountMemo,
3492    useReducer: mountReducer,
3493    useRef: mountRef,
3494    useState: mountState,
3495    useDebugValue: mountDebugValue,
3496    useDeferredValue: mountDeferredValue,
3497    useTransition: mountTransition,
3498    useSyncExternalStore: mountSyncExternalStore,
3499    useId: mountId,
3500  };
3501
3502
3503  const HooksDispatcherOnUpdate: Dispatcher = {
3504    readContext,
3505
3506    use,
3507    useCallback: updateCallback,
3508    useContext: readContext,
3509    useEffect: updateEffect,
```

2. contextType

contextType: 只能用在类组件且只能订阅单一的 Context 来源。

```

DebugReact > src > react > packages > react-reconciler > src > JS ReactFiberClassComponent.js > constructClassInstance
592 function constructClassInstance(
593   workInProgress: Fiber,
594   ctor: any,
595   props: any,
596 ): any {
597   let isLegacyContextConsumer = false;
598   let unmaskedContext = emptyContextObject;
599   let context = emptyContextObject;
600   const contextType = ctor.contextType;
601
602   > if (__DEV__) { ...
638   }
639
640   if (typeof contextType === 'object' && contextType !== null) {
641     context = readContext((contextType: any));
642   } else if (!disableLegacyContext) {
643     unmaskedContext = getUnmaskedContext(workInProgress, ctor, true);
644     const contextTypes = ctor.contextTypes;
645     isLegacyContextConsumer =
646       | contextTypes !== null && contextTypes !== undefined;
647     context = isLegacyContextConsumer
648       ? getMaskedContext(workInProgress, unmaskedContext)
649       : emptyContextObject;
650   }
651
652   let instance = new ctor(props, context);

```

3. Consumer

Consumer 组件，无限制。

react/packages/react-reconciler/src/ReactFiberBeginWork.js

JavaScript

```

function updateContextConsumer(
  current: Fiber | null,
  workInProgress: Fiber,
  renderLanes: Lanes,
) {
  let context: ReactContext<any> = workInProgress.type;

  const newProps = workInProgress.pendingProps;
  const render = newProps.children;

  prepareToReadContext(workInProgress, renderLanes);
  const newValue = readContext(context);
  if (enableSchedulingProfiler) {
    markComponentRenderStarted(workInProgress);
  }
  let newChildren = render(newValue);

```

```
reconcileChildren(current, workInProgress, newChildren, renderLanes)
return workInProgress.child;
}
```