



10-5. 实现函数组件渲染的源码

资源

1. [函数组件](#)

```
function FunctionComponent() {  
  return (  
    <div>  
      <h3>FunctionComponent</h3>  
    </div>  
  );  
}
```

React TSX

render 阶段

beginWork

创建 Fiber

ReactFiber.ts

TypeScript

```
// 根据 TypeAndProps 创建fiber
export function createFiberFromTypeAndProps(
  type: any,
  key: null | string,
  pendingProps: any
) {
  let fiberTag: WorkTag = IndeterminateComponent;

  if (isFn(type)) {
    // 函数组件、类组件
    if (type.prototype.isReactComponent) {
      fiberTag = ClassComponent;
    } else {
      fiberTag = FunctionComponent;
    }
  } else if (isStr(type)) {
    // 原生标签
    fiberTag = HostComponent;
  } else if (type === REACT_FRAGMENT_TYPE) {
    fiberTag = Fragment;
  }

  const fiber = createFiber(fiberTag, pendingProps, key);
  fiber.elementType = type;
  fiber.type = type;
  return fiber;
}
```

ReactFiberBeginWork.ts

TypeScript

```
function updateFunctionComponent(current: Fiber | null, workInProgress:
  const { type, pendingProps } = workInProgress;
  const children = type(pendingProps);
  reconcileChildren(current, workInProgress, children);
```

```

return workInProgress.child;
}

```

completeWork

Changes 4	History	packages/react-reconciler/src/ReactFiberCompleteWork.ts	
<input checked="" type="checkbox"/> 4 changed files <input checked="" type="checkbox"/> examples/src/main.tsx <input checked="" type="checkbox"/> packages/react-reconciler/src/ReactFiber.ts <input checked="" type="checkbox"/> packages/react-reconciler/src/ReactFiberBeginWork.ts <input checked="" type="checkbox"/> packages/react-reconciler/src/ReactFiberCompleteWork.ts		...	@@ -3,6 +3,7 @@ import { Fiber } from "../ReactInternalTypes";
		3 3	import {
		4 4	ClassComponent,
		5 5	Fragment,
		6 6 +	FunctionComponent,
		6 7	HostComponent,
		7 8	HostRoot,
		8 9	HostText,
		...	@@ -16,6 +17,7 @@ export function completeWork(
		16 17	switch (workInProgress.tag) {
		17 18	case Fragment:
		18 19	case ClassComponent:
		20 20 +	case FunctionComponent:
		19 21	case HostRoot: {
		20 22	return null;
		21 23	}
	