



9-2 如何实现页面渲染的入口函数 createRoot

createRoot API

`createRoot` 允许在浏览器的 DOM 节点中创建根节点以显示 React 组件。

```
const root = createRoot(document.getElementById("root"));
root.render(jsx);
```

JavaScript

实现 createRoot

```
import { Container, FiberRoot } from "react-reconciler/src/ReactIntern
import { createFiberRoot } from "react-reconciler/src/ReactFiberRoot";
import { updateContainer } from "react-reconciler/src/ReactFiberReconc
import { ReactNodeList } from "shared/ReactTypes";

type RootType = {
  render: (children: ReactNodeList) => void;
  _internalRoot: FiberRoot;
```

TypeScript

```

};

function ReactDOMRoot(this: RootType, internalRoot: FiberRoot) {
  this._internalRoot = internalRoot;
}

ReactDOMRoot.prototype.render = function (children: ReactNodeList) {
  updateContainer(children, this._internalRoot);
};

export function createRoot(container: Container): RootType {
  const root: FiberRoot = createFiberRoot(container);

  return new (ReactDOMRoot as any)(root);
}

export default {
  createRoot,
};

```

实现 updateContainer

TypeScript

```

import { ReactNodeList } from "shared/ReactTypes";
import { FiberRoot } from "../ReactInternalTypes";

export function updateContainer(element: ReactNodeList, container: Fib
  // ! 1. 获取current
  const current = container.current;
  current.memoizedState = { element };

  const root = container;
  // ! 2. 调度更新
  // scheduleUpdateOnFiber(root, current);
}

```

