

## 组合 & 继承

---

使用一个特殊的 children prop 来将他们的子组件传递到渲染结果中

```
function FancyBorder(props) {
  return (
    <div className={'FancyBorder FancyBorder-' + props.color}>
      {props.children}
    </div>
  );
}
```

```
function WelcomeDialog() {
  return (
    <FancyBorder color="blue">
      <h1 className="Dialog-title">
        Welcome
      </h1>
      <p className="Dialog-message">
        Thank you for visiting our spacecraft!
      </p>
    </FancyBorder>
  );
}
```

组件的内容将作为 children 传递给 props。

也可以不使用 children，而是自定义的属性名：

```
function SplitPane(props) {
  return (
    <div className="SplitPane">
      <div className="SplitPane-left">
        {props.left}
      </div>
      <div className="SplitPane-right">
        {props.right}
      </div>
    </div>
  );
}

function App() {
  return (
    <SplitPane
      left={
```

```
        <Contacts />
      }
      right={
        <Chat />
      } />
    );
  }
```

上面代码将组件作为属性传递。

`<Contacts />` 和 `<Chat />` 之类的 React 元素本质就是对象（object），所以你可以把它们当作 props，像其他数据一样传递。