

04 JSX 和组件

JSX 语法

- JSX 是 JS 语法的扩展，允许在 JS 代码里写组件的 UI 结构，与 HTML 相似。
- JSX 已经成为 ES 语法标准，也可用于其他框架。

标签

1. **首字母大小写含义不同**：首字母小写 - HTML tag；首字母大写 - 自定义组件。
2. JSX 里的标签必须是**闭合**的。
3. 每一段 JSX 只能有一个**根节点**，可以使用 `<></>` 作为根节点，其是 `<Fragment></Fragment>` 的简写。

属性

和 HTML 属性基本一样，但有些和 JS 关键字冲突了

1. `class` 要改为 `className`。
2. `style` 要写成 JS 对象（不能是 string），key 采用**驼峰写法**。
3. `for` 要改为 `htmlFor`。

事件

通过 `onXxx` 的形式给 JSX 标签绑定事件响应函数，如 `onClick`

[关于 TS 的基础知识-1](#)

插值

`{xxx}` 用于插入 **JS 变量、函数或表达式**，也可以用于注释

判断

JS 一般使用 `if...else` 做判断，但不能用于 JSX 的 `{xxx}` 中。所以，可以选择其他方式做判断，

1. 运算符 `&&`
2. 三元表达式 `a ? b : c`
3. 函数封装

```

1  const flag = true;
2  return (
3    <div>
4      {flag && <p>hello</p>}
5      {flag ? <p>你好</p> : <p>再见</p>}
6      <Hello></Hello>
7    </div>
8  )
9  function Hello() {
10    if (flag) return <p>你好</p>; else return <p>再见</p>
11  }

```

循环

使用 `map` 进行循环

JSX 循环必须有 `key` - 帮助 React 识别哪些元素改变了，比如被添加或删除。

- 同级别 `key` 必须唯一
- `key` 是不可改变的 —— 尽量不用 `index`，要用业务 ID（也不要使用随机数）
- `key` 用于优化 VDOM diff 算法

```

1  const list = [
2    { username: 'zhangsan', name: '张三' },
3    { username: 'lisi', name: '李四' },
4    { username: 'shuangyue', name: '双越' },
5  ]
6
7  const ul = <ul>
8    {list.map(user => {
9      return <li key={user.username}>{user.name}</li>
10    })}
11  </ul>

```

防注入攻击

1. JSX 自动处理了很多安全问题，特别是防止跨站脚本攻击（XSS）。React 的 JSX 渲染引擎会自动转义所有插入的内容，以防止恶意脚本被执行。如，

```

1  const userInput = "<script>alert('XSS Attack!');</script>";
2  const element = <div>{userInput}</div>;

```

会被转义为

```
1 <div>&lt;script&gt;alert('XSS Attack!');&lt;/script&gt;</div>
```

2. `dangerouslySetInnerHTML` 是 React 提供的一个特殊属性，用于将 HTML 字符串直接插入到 DOM 中。如，

```
1 const htmlString = '<strong>Dangerous HTML</strong>';
2 const element = <div dangerouslySetInnerHTML={{ __html: htmlString }} />;
```

`dangerouslySetInnerHTML`：这个属性用于直接插入 HTML 内容，但它需要一个对象作为值，对象必须包含 `__html` 属性，包含你要插入的 HTML 字符串。

组件和 props

React apps are made out of components. A component is a piece of the UI (user interface) that has its own logic and appearance. A component can be as small as a button, or as large as an entire page.

1. 组件就是一个**函数**，输入 props，返回一段 JSX。
2. 组件就是一个 **UI 片段**，拥有独立的逻辑和显示。
3. 组件拆分也**有利于代码组织和维护**，尤其对于大型软件，提高开发效率。
4. 组件通过**嵌套**来组织 UI 结构。
5. 组件的类型为 **FC**，有两种引入方式。

```
1 import { FC } from "react"
2 import type { FC } from "react"
3
4 const ComponentName: FC = () => {} // 定义一个组件
```

关于 TS 的基础知识-2

开发者工具：React Developer Tools