基本概念

目录

- 组件__
- 元素__
- element
- 参考链接____

组件

UI使用数据结构表示。 UI的变化，就是一种数据结构，变成另一种数据结构。

同样的数据结构，就得到同样的 UI。这就是说， UI是一个纯函数。

```
function NameBox(name) {
  return { fontWeight: 'bold', labelContent: name };
}
```

一个大的 UI组件可以包含小的组件。

```
function FancyUserBox(user) {
  return {
    borderStyle: '1px solid blue',
    childContent: [
      'Name: ',
      NameBox(user.firstName + ' ' + user.lastName)
    ]
  };
}
```

各种小组件可以组合成一个大组件（ composition）。

同一个组件可以有内部状态，即有多种数据结构表示。

元素

在 React中，类就是组件（ component），`element`是描述组件实例的对象。

`element`并不是真的实例。 Element上面不能调用任何方法。它只是实例的一个不可变（ immutable ）描述，带有两个属性： `type`和 `props`。

如果 `type`属性是一个字符串， `element`就代表一个 DOM节点， `type`就是它的标签名， `props`是它的属性。

```
{
  type: 'button',
  props: {
    className: 'button button-blue',
    children: {
      type: 'b',
      props: {
        children: 'OK!'
      }
    }
  }
}
```

上面代码对应下面的 HTML代码结构。

```
<button class='button button-blue'>
  <b>
    OK!
  </b>
</button>
```

`children`属性就是它的子元素。

如果 `type`属性是函数或对应 React组件的类。

```
{
  type: Button,
  props: {
    color: 'blue',
    children: 'OK!'
  }
```

```
}
```

□□□□□□□□□□ `children` □□□

```
const DeleteAccount = () => ({
  type: 'div',
  props: {
    children: [{
      type: 'p',
      props: {
        children: 'Are you sure?'
      }
    }, {
      type: DangerButton,
      props: {
        children: 'Yep'
      }
    }, {
      type: Button,
      props: {
        color: 'blue',
        children: 'Cancel'
      }
   }]
});
```

□□□ JSX□□□□

```
const DeleteAccount = () => (
  <div>
    <p>Are you sure?</p>
    <DangerButton>Yep</DangerButton>
    <Button color='blue'>Cancel</Button>
  </div>
);
```

□□□□□□□□□□□□□□□□□□□□□□

```
const Button = ({ children, color }) => ({
  type: 'button',
```

```
  props: {
    className: 'button button-' + color,
    children: {
      type: 'b',
      props: {
        children: children
      }
    }
  }
});
```

element

`element`是一个描述组件如何显示在屏幕上的对象。`element`可以在`props`里面包含其他`element`。一旦创建，就不再发生变化。

`component`可以是一个带有`render`方法的类，也可以定义为一个函数。它总是接受`props`作为输入，在输出之中返回一个`element`树。

参考链接

- [React - Basic Theoretical Concepts](), by sebmarkbage

版本号 #2
由 壳壳虫 创建于 2017-04-09 10:10:41 UTC
由 壳壳虫 更新于 2017-04-09 10:10:41 UTC