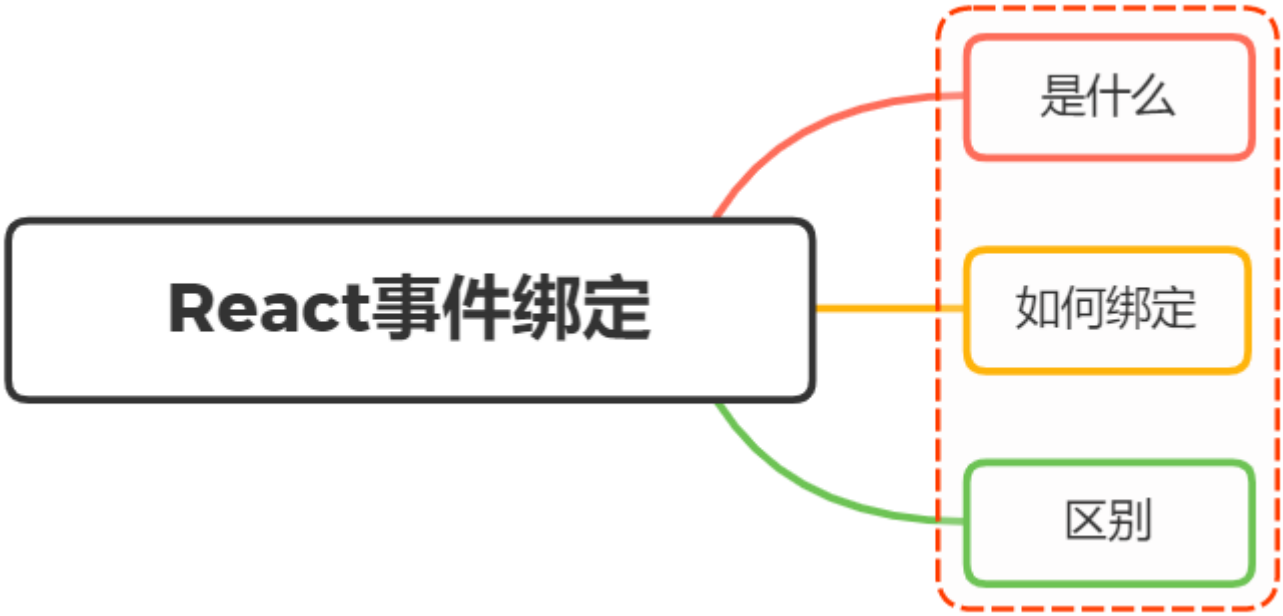
面试官：React构建组件的方式有哪些？区别？



一、是什么

组件就是把图形、非图形的各种逻辑均抽象为一个统一的概念（组件）来实现开发的模式

在React中，一个类、一个函数都可以视为一个组件

在之前文章中，我们了解到组件所存在的优势：

- 降低整个系统的耦合度，在保持接口不变的情况下，我们可以替换不同的组件快速完成需求，例如输入框，可以替换为日历、时间、范围等组件作具体的实现
- 调试方便，由于整个系统是通过组件组合起来的，在出现问题的时候，可以用排除法直接移除组件，或者根据报错的组件快速定位问题，之所以能够快速定位，是因为每个组件之间低耦合，职责单一，所以逻辑会比分析整个系统要简单
- 提高可维护性，由于每个组件的职责单一，并且组件在系统中是被复用的，所以对代码进行优化可获得系统的整体升级

二、如何构建

在React目前来讲，组件的创建主要分成了三种方式：

- 函数式创建
- 通过 React.createClass 方法创建
- 继承 React.Component 创建

函数式创建

在React Hooks出来之前，函数式组件可以视为无状态组件，只负责根据传入的props来展示视图，不涉及对state状态的操作

大多数组件可以写为无状态组件，通过简单组合构建其他组件

在React中，通过函数简单创建组件的示例如下：

```
function HelloComponent(props, /* context */) {
  return <div>Hello {props.name}</div>
}
```

通过 React.createClass 方法创建

React.createClass是react刚开始推荐的创建组件的方式，目前这种创建方式已经不怎么用了

像上述通过函数式创建的组件的方式，最终会通过babel转化成React.createClass这种形式，转化成如下：

```
function HelloComponent(props) /* context */{
  return React.createElement(
    "div",
    null,
    "Hello ",
    props.name
  );
}
```

由于上述的编写方式过于冗杂，目前基本上不使用上

继承 React.Component 创建

同样在react hooks出来之前，有状态的组件只能通过继承React.Component这种形式进行创建

有状态的组件也就是组件内部存在维护的数据，在类创建的方式中通过this.state进行访问

当调用this.setState修改组件的状态时，组价会再次会调用render()方法进行重新渲染

通过继承React.Component创建一个时钟示例如下：

```
class Timer extends React.Component {
  constructor(props) {
    super(props);
    this.state = { seconds: 0 };
  }

  tick() {
    this.setState(state => ({
      seconds: state.seconds + 1
    }));
  }

  componentDidMount() {
    this.interval = setInterval(() => this.tick(), 1000);
  }
```

```
  componentWillUnmount() {
    clearInterval(this.interval);
  }

  render() {
    return (
      <div>
        Seconds: {this.state.seconds}
      </div>
    );
  }
}
```

三、区别

由于`React.createClass` 创建的方式过于冗杂，并不建议使用

而像函数式创建和类组件创建的区别主要在于需要创建的组件是否需要为有状态组件：

- 对于一些无状态的组件创建，建议使用函数式创建的方式
- 由于`react hooks`的出现，函数式组件创建的组件通过使用`hooks`方法也能使之成为有状态组件，再加上目前推崇函数式编程，所以这里建议都使用函数式的方式来创建组件

在考虑组件的选择原则上，能用无状态组件则用无状态组件

参考文献

- https://react.docschina.org/