Vue事件处理

知识点

- Vue事件处理流程
- 相关api
- 彩蛋

测试代码

examples\test\events\index.html

```
<!DOCTYPE html>
<html>

<head>
    <title>Vue事件处理</title>
    <script src="../../../dist/vue.js"></script>
</head>

<body>
    <div id="demo">
        <h1>事件处理机制</h1>
        <!--普通事件-->
        <p @click="onClick">this is p</p>
        <!--自定义事件-->
        <comp @myclick="onMyClick"></comp>
    </div>
    <script>
        // 声明自定义组件
        Vue.component('comp', {
            template: `<div @click="$emit('myclick')">this is comp</div>`,
        })
        // 创建实例
        const app = new Vue({
            el: '#demo',
            methods: {
                onClick() {
                    console.log('普通事件');
                },
                onMyClick() {
                    console.log('自定义事件');
```

```
                }
            },
        });
    </script>
</body>
</html>
```

事件处理整体流程

- 编译阶段：处理为data中的on

```
ƒ anonymous(
) {
with(this){return _c('div',{attrs:{"id":"demo"}},[
    _c('h1',[_v("事件处理机制")]),_v(" "),
    _c('p',{on:{"click":onClick}},[_v("this is p")]),_v(" "),
    _c('comp',{on:{"myclick":onMyClick}})],1)}
}
```

- 初始化阶段：
  - 原生事件监听 platforms/web/runtime/modules/events.js

    整体流程：patch() => createElm() => invokeCreateHooks(vnode, insertedVnodeQueue) =>updateDOMListeners()
  - 自定义事件监听 initEvents core/instance/events.js

    整体流程：patch() => createElm() => createComponent() => hooks.init() => createComponentInstanceForVnode() => _init() => initEvents(vm) => ...

调试验证

相关API

- $on() 监听自定义事件
- $emit() 派发自定义事件
- $once() 仅监听一次
- $off() 取消事件监听

彩蛋

在Vue当中，hooks可以作为一种event，在Vue的源码当中，称之为hookEvent。

```
<Table @hook:updated="handleTableUpdated"></Table>
```

场景：有一个来自第三方的复杂表格组件，表格进行数据更新的时候渲染时间需要1s，由于渲染时间较长，为了更好的用户体验，我希望在表格进行更新时显示一个loading动画。修改源码这个方案可行，但是不优雅。

callHook src\core\instance\lifecycle.js

调用生命周期钩子时若发现标记则额外派发hook事件

$on src\core\instance\events.js

监听时若发现是hook事件则做一个标记