类组件

React的JSX创建出来的是虚拟DOM，而不是HTML	
1、定义组件的时候，return 后面只能有一个根标签，不能有多个，但这个标签内部可以有其他多个标签 2、使用组件的时候，首字母必须大写 3、如果最外层实在不想放置一层div根目录，可以使用<></> 空标签代替	
在jsx语法中，需要书写js代码的时候，请先加上{ } 再书写js语法	

state：控制状态
修改字符串，数字，数组
（App1）
	import React, { Component } from 'react'
	
	export default class App6 extends Component {
	state = {
	username: "张三"，
	number：1
	}
	render() {
	return (
	
	 {this.state.username}
	<button onClick={this.handleClick.bind(this)}>按钮</button>
	
	)
	}
	handleClick(){
	this.setState({
	username: "李四"
	})
	
	this.setState({number:this.state.number+1})
	}

	}
Tab栏 (App2)	

函数式组件

父传子
(App4)
	父组件以属性形式将值传递给子组件即可
	
	let msg=”hello”
	const Child=(props) => {
	return {props.msg}
	}
	const Father = () =>{
	return <Child msg={msg}/>
	}

子传父 (App4)	子传父的本质是父组件创建好方法，然后传递到子组件调用 return <Child num={num} setNumm={setNum}/> 父级直接把函数提供给子类，子类能够直接获取函数并修改

Hooks

都写在最顶端（return前面）

useState (App3)	用于修改状态 const [num,setNum] = useState(1) eg：累加 setNum(num+1)
useEffect (App3)	有时我们需要在挂载后请求数据、销毁前清除数据(mount=[]) 或是在更新后获取数据，那么就需要借助useEffect(update=[检测的数据]) useEffect(()=>{ },[])
creatContext (App5)	利用上下文空间context实现跨级/多级组

	件传值 const NumContext=creatContext() provider: 直接共享数字和方法 value={{num,setNum}} consumer: 1) 解构出来使用 ({num,setNum})=>(<></> 2) const {num,SetNum} = useContext(NumContext) <></>
useRef (App6)	不受控组件代表表单元素的值不受state控制，那么获取表单元素的值便只能空过ref获取 受控组件代表表单元素的值受state控制（函数式组件中可以理解为受useState控制），获取其值时直接通过state获取即可
memo (App5)	当父组件更新时，子组件会被迫更新.这就导致性能损耗. 我们需要借助memo这个hook来缓存Sub组件，避免它被强制更新.
useMemo/useCallback (App7)	

状态管理工具React Redux

https://react-redux.js.org/

提供器	顶级组件外
连接器 (App8 and App9) 事件：创建修改字符串和数字的事件	在需要调用数据的组件中使用连接器 connect(state映射,dispatch映射) state： 为了reducer中的state映射成props。让开发者用props.num去调用state中的num dispatch：

	为了reducer中的事件映射成props。

路由

https://reactrouter.com/

Outlet	在App.jsx中，使用Outlet来展示子路由 如此，便可在浏览器地址栏中通过url的切换实现不同路由对应的组件展示
Link (App10)	原生JS中的a标签，在react-router-dom中使用Link标签代替
useLocation	使用useLocation这个路由hook，可以获取当前地址栏路径
useNavigate	使用useNavigate这个路由hook，可以实现路由跳转。若要判断这个路径如果为/，则自动跳转到/home
useParams (pages-List.jsx)	通过在路由中配置子路由形式，可以实现参数传递。如给/list携带参数，以这个形式： App文件：/list/123 路由文件：/list/:id 如果想要获取参数的值，可以使用useParams这个路由hook
useSearchParams (pages-Detail.jsx)	地址栏携带参数还可以通过问号形式 获取该参数的方式，需要使用useSearchParams这个路由hook，并且调用getAll方法才能获取得到
state携带 button按钮 (pages-Home.jsx) (pages-Detail.jsx) 分为两种情况	在使用useNavigate()这个hook实现跳转时，其实也可以携带参数
404匹配 (pages-Error.jsx)	如果碰上未在路由文件中有做配置的路由，便可认定为404页面