React进阶

寒假训练项目-Week4

清华大学自动化系学生科协

温昊哲

2023.2.8

目录

- Props和State
- 函数组件
- Hook
- Debug
- 使用习惯
- 外部组件库和React示例
- 动手尝试

昨日回顾

后端：Django .sqlite3 models.py-数据表 数据库迁移
tools.py urls.py admin.py
get post 字典结构
python manage.py runserver

React基础： yarn install/start index.js App.js <Router>
return单一标签 配套.css import xxx from xxx/xxx
export default xxx; export {xxx , xxx};

Props属性

- **向子组件传递数据/组件**
  - props用于向子组件传递必要的数据
  - props出现在组件的函数参数中，可以换名

  - 不能修改props中的数据，即子组件不能通过此方法向父组件返回数据
  - 父子组件传递的数据的名称需要统一，否则无法在子组件中捕获
  - 可以使用props传递组件，不仅限于数据

State状态（旧）

- **this.state**

  组件中可变的数据成员必须保存在当前对象的state结构中

  state发生变化时由浏览器自动更新界面需要发生改变的内容，不发生改变的组件不会更新。

  state不是约定的名称，是一个react的关键字，不能改名使用。

初始化(constructor专用)：this.state = {xxx：xxx};

修改数据： this.setState({xxx：xxx});

访问数据： this.state.xxx;

函数组件

- **基于组合，避免继承**

```
function my_fun(props){
    ...
    return(<div>...</div>);
}
const my_fun = (props) =>{
    ...
    return(<div>...</div>);
}
```

Hook（旧）

- **事件监听**

  在组件中使用各类事件监听函数可以实现高效的事件处理，也可以用于在指定的条件下触发自定义事件

- **常用的监听函数（旧）**

  componentDidMount() 组件已挂载

  componentWillMount() 组件即将挂载

  componentWillUnmount() 组件即将卸载

  ....

Hook(新)

- **旧版监听/注入的局限**
  - this只能在类中使用，因此无法使用this.state等
  - 函数不能嵌套函数，因此无法使用component...系列的函数监听器
  - 异步问题

- **use...系列**
  - 所有的React预设的监听/注入器均以use...开头
  - 使用之前需要 import {use...} from “react”;

Hook

- **useState() 声明数据成员**

  - 使用函数组件时应使用useState绑定数据成员和修改该数据的函数，同时为其初始化

  - 使用useState定义的数据成员不会因为组件刷新或函数重新调用而销毁或再次初始化

  - useState()返回一个二元组，前者为数据名，后者为更新该数据的函数（？）名。只接受一个参数，即初始化的值。

  - 一次只能声明一个数据成员，但可以多次调用以声明多个数据。

Hook

- **useState()使用示例**
  - 声明数据： const [age, setAge] = useState(20);
    const [fruit, setFruit] = useState(‘pear’);
  - 访问数据：age （取代this.state.age）
  - 修改数据：setAge(15)

  - setAge不需要定义，也没有返回值。调用setAge会将其收到的参数写入到age中。

Hook

- **useEffect() 在render之后的操作**

  - useEffect()可用于自定义在**每次**组件刷新之后的操作，效果等同于componentDidMount、componentDidUpdate和componentWillUnmount 三合一。

  - 无需区分组件首次挂载（mount）和组件更新（update）两类相似的事件

  - 可以在useEffect()中定义嵌套的函数

Hook

- **useEffect() 使用**

  - 定义： useEffect(() => {...}); 即传入一个（匿名）函数

  - 由于useEffect()在组件函数之内，故在传入的函数中可以访由useState()所定义的任何数据成员

  - 嵌套函数定义：useEffect(function my_fun(...){...} ...);

Hook

- **useParams() 从url中提取信息**
  - 可以在任意级别的组件中获得当前url中的参数信息
  - 将url中的父级可变参数打包，返回的结构类似字典，但只需要使用键即可访问值，即该字典没有字典名
  - 可变参数需要在<Route path>标签中有所匹配才能识别
  - 可以同时接收多个参数，如 const {id, name, age} = useParams();
    访问结果时只需使用id, name, age即可

Hook

- **useParams() 使用**

 - Route定义:

```
<Routes>
    <Route path="users">
      <Route path=":userId" element={<ProfilePage />} />
      <Route path="me" element={...} />
    </Route>
 </Routes>
```

Hook

- **useParams() 使用**
  - 参数提取：

```
function ProfilePage() {
        let { userId } = useParams();
        //或
        //const params = useParams();
        //使用params.userId访问数据
  // ...
}
```

Hook

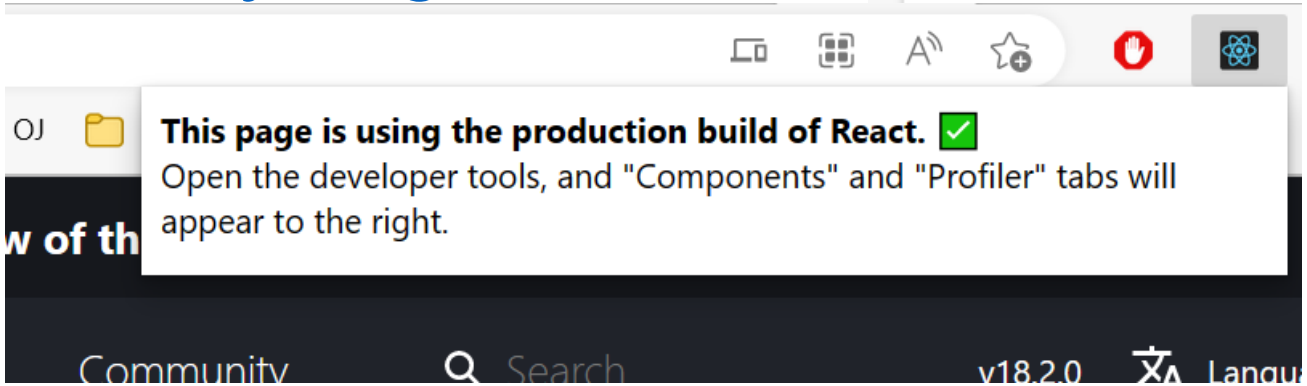
- **自定义Hook**
  - 自定义的Hook结构上形同一般的函数
  - 无法设定触发条件，只能用于包裹已经使用基础Hook写好的模块
  - 调用规则及效果与模块内的代码完全相同
  - 可以设定传入参数、返回值等一般函数所具有的属性
  - 使用时需要遵守Hook的使用规则，即不应在判断、循环及嵌套函数内使用
  - 宜以use...开头，从而易于与一般的函数区分。

Debug

- **Component tree**

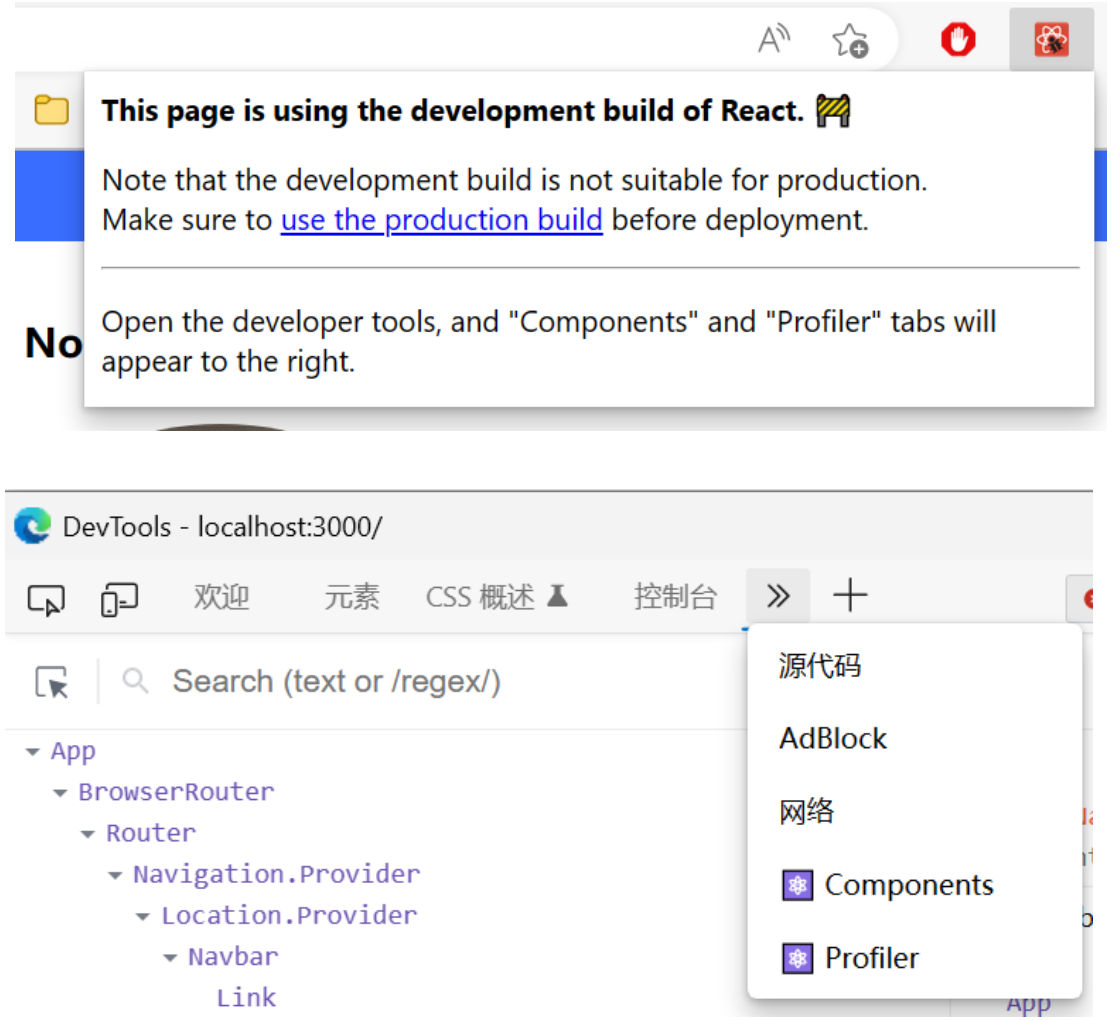
- 安装浏览器插件React Dev Tools
- 启用“允许访问文件url”
- 打开基于React的界面测试插件是否能正常识别其中的React成分

  （例如https://reactjs.org/）



Debug

- **Component tree**
  - 用浏览器打开工程项目，启动React Dev Tools插件
  - 打开控制台，标签页里应多出了components和profiler选项（注意在>>号里）
  - 打开components可以以树形结构查看网页组件的关系



使用习惯

- **应做的**

1、自定义组件均以大写开头，才能被React正确识别

2、为每个组件的.js搭配一个.css，而非将所有css集中到某处

3、使用函数组件而非类组件（同上）

4、组件内部在最高优先级调用Hook

5、Hook只在函数型组件中使用，不用在一般的js函数中

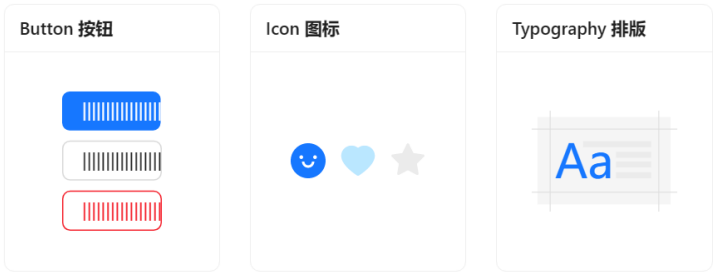
使用习惯

- **应避免的**
  1、使用document.getElementById(xxx)，易发生id冲突
  2、尝试直接为组件数据赋值而不使用this.setState()（过时）
  3、循环逻辑：使用某函数处理某事件，但在该函数中使用了该事件的结果
  4、将props复制到state中（过时）
  5、在字典对象里使用无实际含义的数字型索引
  6、使用.bind()将函数改为公开类型（过时）
  7、在一个.js中塞入多个组件定义
  8、在循环体、条件判断、嵌套函数中使用任意类型的Hook

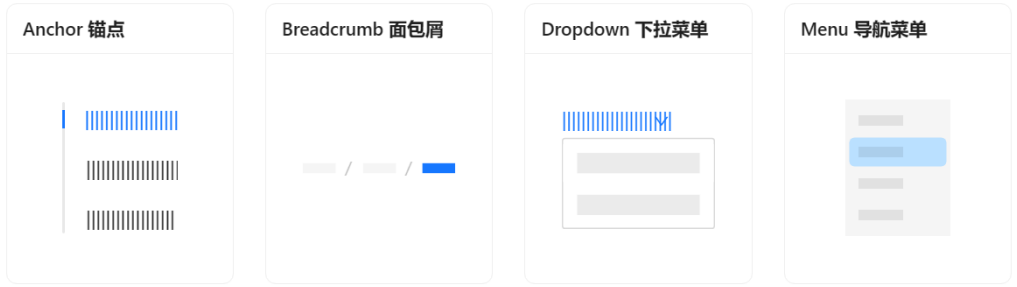
antd组件库

- 丰富

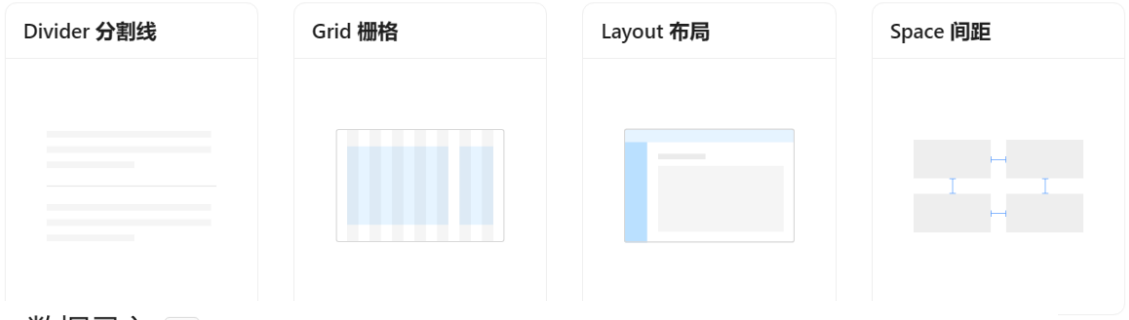
通用 3



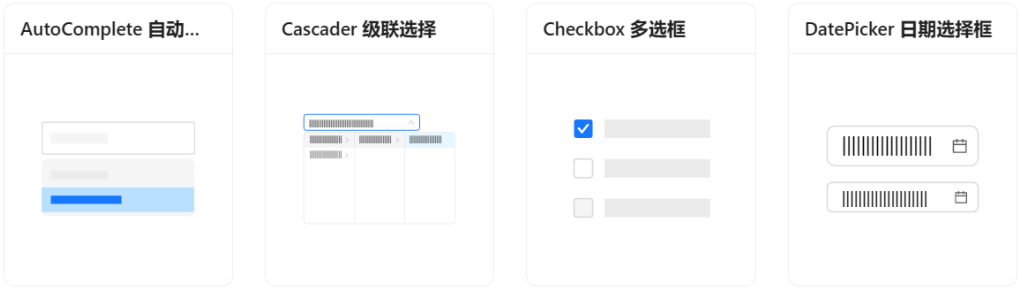
导航 6



布局 4



数据录入 17



antd组件库

- **安装**
  - npm install antd / yarn add antd
- **选择组件**
  - 前往 https://ant.design/components/overview-cn/
  - 挑选所需的组件，记下它的标签名
- **使用组件**
  - 在.js中使用 impot {标签名} from ‘antd’；
  - 可能还需要引入.css文件：import 'antd/dist/antd.css';

React示例

- 科协网站 https://thuasta.cn/#/

资料链接

- **新版React教程（从函数组件开始）**
  - https://www.runoob.com/react/react-props.html
- **Hook**
  - https://reactjs.org/docs/hooks-reference.html
- **React Dev Tools 使用**
  - https://juejin.cn/post/6877546408925200391
- **antd 组件库**
  - https://ant.design/components/icon-cn/
- **科协网站.git （dev分支最新）**
  - https://git.tsinghua.edu.cn/zhengcj20/asta-website

谢谢大家