|  |  |  |
| --- | --- | --- |
| **reactJS** | | |
| Hello World | var helleWorld = **React**.**createClass**({  render:function () {  return **React**.createElement('h1',null,'Hello world')  } }) **ReactDOM**.render(**React**.createElement(helleWorld,null),document.getElementById('hello')) |  |
| JSX语法 | var HelleWorlds = **React**.**createClass**({  render:function () {  return <h1>Hello World</h1>;  } }); **ReactDOM**.render(  <HelleWorlds/>,  document.getElementById('hello') ) | HelleWorlds 首字母大写 |
| JSX循环 | var arr = ['lion','CC','CMQ']; var Jsx = **React**.**createClass**({  render:function () {  return <div>  {  arr.map(function (name) {  return <div key={name}>Hello，{name}</div>  })  }  </div>  } }); **ReactDOM**.render(<Jsx name="前面的"/>,document.getElementById('hello'));  var arr2 = [  <h1 key="1">lion</h1>,  <h1 key="2">CC</h1>,  <h1 key="3">CMQ</h1> ]; **ReactDOM**.render(<div>{arr2}</div>,document.getElementById('hello2')) | return <div>  Return 后不允许换行 |
| state | var TextBoxComponents = **React**.**createClass**({  getInitialState: function () {  return {  **enabledStatus**:false  }  },  handleClick:function (event) {  this.setState({  **enabledStatus**:!this.state.**enabledStatus** })  },  render:function () {  return (  <div>  <input type="text" disabled={this.state.**enabledStatus**} />  <button onClick={this.handleClick}>改变状态</button>  </div>  )  } }) **ReactDOM**.render(<TextBoxComponents/>,document.getElementById('hello')) | onClick大写；  属性赋值不加双引号； |
| props | var PropsComponents = **React**.**createClass**({  render:function () {  return (  <div>myAttr属性值是：{this.**props**.myAttr}</div>  )  } }) **ReactDOM**.render(<PropsComponents myAttr="aaa" />,document.getElementById('hello')) |  |
| hook | componentWillMount()  componentDidMount()  componentWillUpdate(object nextProps, object nextState)  componentDidUpdate(object prevProps, object prevState)  componentWillUnmount() |  |
| 子组件 | var ChildrenComponents = **React**.**createClass**({  render:function () {  return(  <ol>  {  **React**.**Children**.**map**(this.**props**.children,function (child) {  return(  <li>{child}</li>  )  })  }  </ol>  )  } }); **ReactDOM**.render(  <ChildrenComponents>  <span>lion</span>  <span>CMQ</span>  </ChildrenComponents>,document.getElementById('hello')) |  |
| propType | var PropsType = **React**.**createClass**({  **propTypes**:{  **title**:**React**.**PropTypes**.**number**.**isRequired** },  getDefaultProps:function () {  return{  **tip**:'lion'  }  },  render:function () {  return(  <div>  <h1>{this.**props**.**title**}</h1>  <h1>{this.**props**.**tip**}</h1>  </div>  )  } }) var flag = 111 **ReactDOM**.render(<PropsType title={flag} />,document.getElementById('hello')) | propTypes设置；  getDefaultProps初始化props |
| 获取真实DOM | var Refdom = **React**.**createClass**({  render:function () {  return(  <div>  <input type="text" ref="textInput"/>  <button onClick={this.clickhandle}>点击</button>  </div>  )  },  clickhandle:function () {  this.**refs**.textInput.**value** = 111  } }) **ReactDOM**.render(<Refdom/>,document.getElementById('hello')) |  |
| **脚手架搭建** |  |  |
| creat-react-app | npm install -g create-react-app  create-react-app my-app |  |
| generator-react-webpack | npm install -g yo  npm install -g generator-react-webpack  yo react-webpack |  |
| webpack构建 | npm init  npm install --save-dev webpack  配置webpack.config.js  cnpm install --save-dev webpack-dev-server  cnpm install --save-dev babel-core babel-loader babel-preset-es2015 babel-preset-react  cnpm install --save react react-dom  const path = require('path'); **module**.exports = {  **entry**:'./app/index.js',  **output**:{  **filename**:'index.js',  **path**:path.resolve(\_\_dirname,'dist'),  **publicPath**:'temp/'  },  **devServer**:{  **contentBase**:'./',  **host**:'localhost',  **port**:8000,  **compress**:true  },  **module**:{  **loaders**:[{  **test**:/\.js$/,  **exclude**:/node\_modules/,  **loaders**:"babel-loader",  **query**:{  **presets**:['es2015','react']  }  }]  } } |  |
| Router |  |  |
|  | import React from 'react'; import ReactDOM from 'react-dom'; import {BrowserRouter as Router , Route , Switch , Redirect} from 'react-router-dom'; import Liona from './child/liona.js'; import Lionb from './child/lionb.js'; import Lionc from './child/lionc.js'; import Error from './child/error.js'; import Nav from './child/nav.js'; import './index.css'; //import App from './App';   import registerServiceWorker from './registerServiceWorker';  ReactDOM.render(  <Router basename="api" forceRefresh={true}>  <div>  <Nav/>  <Switch>  <Route path="/" exact component={Liona}/>  <Route path="/lionb" component={Lionb}/>  <Route path="/lionc/:react" exact component={Lionc}/>  <Redirect from="/redirct" to="/lionb" />  <Route component={Error}/>  </Switch>  </div>  </Router>  , document.getElementById('root')); registerServiceWorker();  import React from 'React'; import {NavLink as NavLinks} from 'react-router-dom'; import './nav.css'  const navbar = () => (  <div>  <div>  <ul>  <li><NavLinks className="nav" to="/">liona</NavLinks></li>  <li><NavLinks className="nav" to="/lionb">lionb</NavLinks></li>  <li><NavLinks className="nav" to="/lionc/hahah">lionc</NavLinks></li>  <li><NavLinks className="nav" to="/redirct">redirct</NavLinks></li>  <li><NavLinks to='/react'>404</NavLinks></li>  </ul>  </div>  </div> )  export default navbar |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |