Component：exported

reusable building blocks: js function that returns jsx

Node js: run time for js, allows js running out of browser

Redux: state management liberay

Cmds:

Create project: Npx create-react-app project-name

Install/uninstall npm: npm install/uninstall

Start : npm start localhost:3000

Fix npm problem: npm audit fix –force

Zipped project: unzip, cd to folder then: npm install, npm start

Shortcut:

Format document: shift alt f

useEffect: responds to an action? useEffect(()=>{

//effect logic

Return()=>{

//clean up logic, this runs everytime before effect logic, except the first time

}

}, [ dependency])

Usually, functions rerun upon state value update, but useEffect rerun only upon dependencies change.

UseReducer: update state based on another state

Const[state, dispatchFn]= useReducer(reducerFn, initialState, initFn)0

reducerFn: (state, action)=>{ returns newState} : can be outside the component function

context: pass data in between components, avoid long prop chain

context: app-wide state{}

Portal: used to place tag to defined tag(docu.getelementbyID, ReactDom.createPortal(<tag>,<des>))

Ref: 一个有current(其中有actual dom node)属性的obj，可用以代替state(but read only)，或avoid updating state every key strok

**Parent component**中

Const amountInputRef=useRef()

SomeHandler=event=>{

Const enteredAmountInput=amountInputRef.current.value ---将ref值赋予变量

}

Return<form>

<input ref={amountInputRef}/>

</form>

**Child component**中

const Input = React.fowardRef((props, ref)=>{

Return <div>

<Input {…props.input}/>

<div>

})