React will re-render when its completely different object.

**State**: only this component can be modify.

**Event Handler:** Every time, this event has been triggered, there is functionality need to happened.

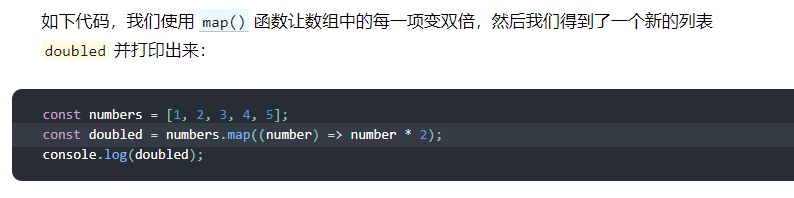
**callBack Function:** pass handler, it want to call some process. (This button want to be called whenever it gotta be changed).

React Render when **state has COMPLETE different object in memory**, then it will re-render.

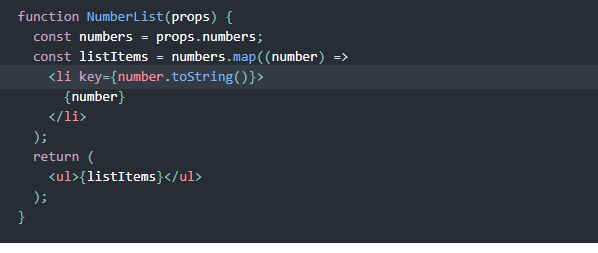
**setState->** doing the shallowmerge when pass a object(find the keys in the state, compare and update the value when they have same keys.)

* **setState(state,props)** -> **state** means the **current state**(update the state based on previous state)

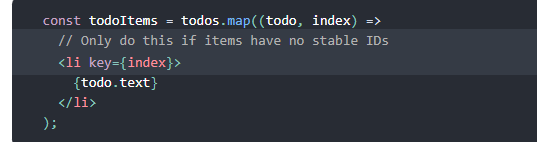
**map**() function:



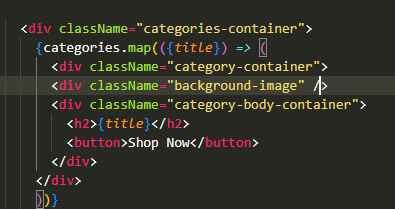
在React中， 使用map function则必须使用 key value(用于确定哪些值被改变



如果没有id，则可以用value的index作为key。



**Map也可以传输component：**

****

**根据特定object，遍历整个array输出该component**

**Single Page Application(SPA)**:

* Send Request to Server
* Server send back HTML,CSS,JS(contain React Code)

**SASS**;

* SCSS 为sass文件
* SASS -> 允许直接使用子类如果该子类在parent class
* backgroundImage = `url(${}`)

**比如：在该子类里面可以直接操控子类下的tag element 渲染**

****

**React-Router:**

* **npm install react-router-dom**
* Import {BrowserRouter} from’react-router-dom’ in index.js
* <BrowserRouter> <App> </BrowserRouter>
* **In App component:** 
  + **Import Router, route**
  + **<routes>**

**<Route path=’/extendName’ element={componentName}>**

**</routes>**

* <outlet /> -> This allows nested UI to show up when **child routes** are rendered.
* **显示两个Component当这两个component使用同一个path (Outlet)**
* <*Route* path='/' element={<*Navigation* />} >
* <*Route* index element={<*Home* />} />
* </*Route*>
* Fragment-> return nothing, do not need rendering all other things. Using Fragment instead of using div
* <Link className to=’/pathName’></Link> (仅限内部链接)
* **Import ReactComponent as logoName from ‘logoPath’;**
  + **<logoName />**