实现mobx

思路

1.通过autoRun触发get拦截。

2.通过get注册target,key和autoRun的一一对应关系。

3.通过set去更新属性，执行autoRun。

Mobx.js

```
let proxies = new WeakMap();
let observers = new WeakMap();
let currentObserver = null;

const isObserver = (obj)  => {
    return obj === proxies.get(obj);
}

const toObserver = (obj ,_fun) =>{
    return new Proxy(obj,{
        get (target,key,receiver) {
            console.log('获取值');
            // 判断代理是否存在
            const targetProxy = proxies.get(target);
            const observer =  observers.get(target)
            const autoFunSet = observer && observer.get(key);
            if(currentObserver && targetProxy && !autoFunSet){
                observer.set(key, new Set().add(currentObserver));
            }

            if(currentObserver && targetProxy && autoFunSet){
                autoFunSet.add(currentObserver);
            }
            // 看看 这个target -> key - currentFunc是否是一一对应关系，如果不是，则添加，如果是这不作为。
            return Reflect.get(target,key,receiver);
        },
        set (target,key,value,receiver) {
            console.log('设置值');
            // 查看这个target ，key 是否是代理，如果是则找其相关的对应关系，执行对应关系，不是则直接返回值
            const targetProxy = proxies.get(target);
            const observer = observers.get(target);
            const autoFunSet = observer && observer.get(key);

            if(typeof target !== 'object' && !targetProxy){
                return Reflect.set(target,key,value,receiver);
            }

            if(targetProxy && observer && autoFunSet){
                const autos = observer.get(key);
                for ( let _func of [...autos]){
                    _func();
```

```
                }
            }
            return Reflect.set(target,key,value,receiver);
        }
    })
}

const registerObserver = (target,key) => {
    observers.set(target,new Map(key,this.currentObserver));
}

const autoRun = (_fun) => {
    console.log('autoRun')
    currentObserver = _fun;
    if(typeof _fun === 'function'){
        _fun();
    }
    currentObserver = null;
}

const observable = (obj) => {
    const proxy = proxies.get(obj);
    if(proxy){
        return proxy;
    }
    const observer = toObserver(obj);
    proxies.set(obj,observer);
    observers.set(obj,new Map());
    return  observer;
}
```

test.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>Document</title>
    <style>
        div {
            width: 500px;
            height: 500px;
            margin: auto;
        }
    </style>
</head>
<body>
    <p>
        颜色: <input id="input"/>
    </p>
    <p>
```

```
        文案: <input id="text"/>
    </p>
    <ul id="list"></ul>
    <script type="text/javascript" src="./Mobx.js"></script>
    <script>
        const input = document.querySelector('#input');
        const text = document.querySelector('#text');
        const list = document.querySelector('#list');

        const obj = observable({
            text:'hehe',
            color:'red'
        });

        input.oninput = (e) => {
            const value = e.target.value;
            console.log(value)
            obj.color = value;
        }

        text.onblur = (e) => {
            const value = e.target.value;
            console.log(value)
            obj.text = value;
        }

        autoRun(() =>{
            input.style.color = obj.color;
        });

        autoRun(() =>{
            let li = document.createElement('li');
            li.innerText = obj.text;
            list.appendChild(li);
        })
        console.log(obj.a,input);
    </script>
</body>
</html>
```