

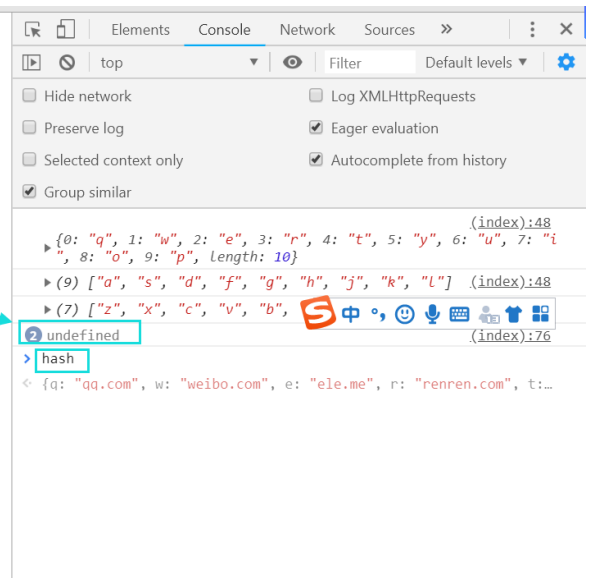
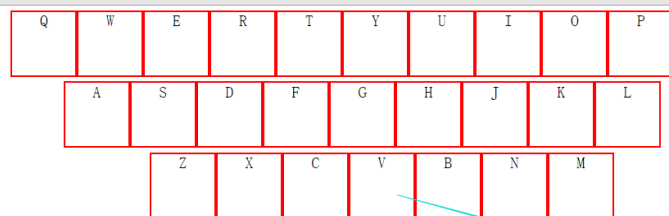
覆盖原来的hash

总结

一个bug的解决

完整代码:

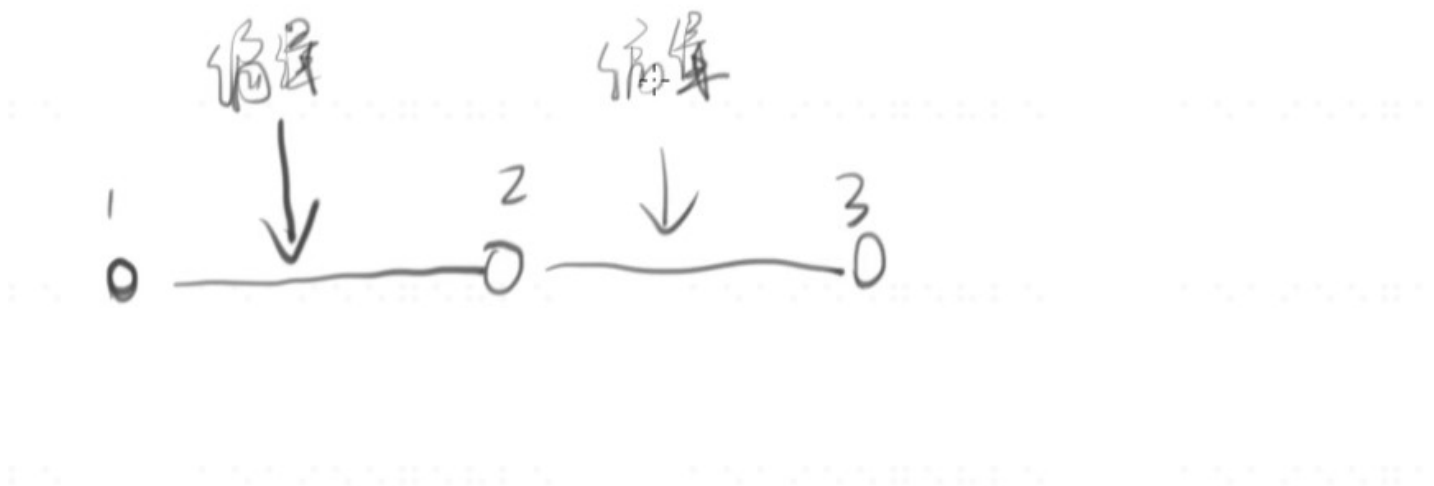
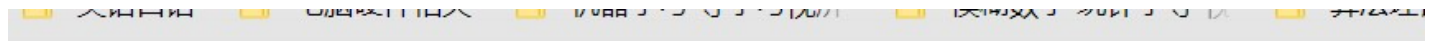
只要刷新一下页面hash就没有了;



有什么办法可以留住这些hash呢;

没有办法让js的变量在下个页面继续保留;

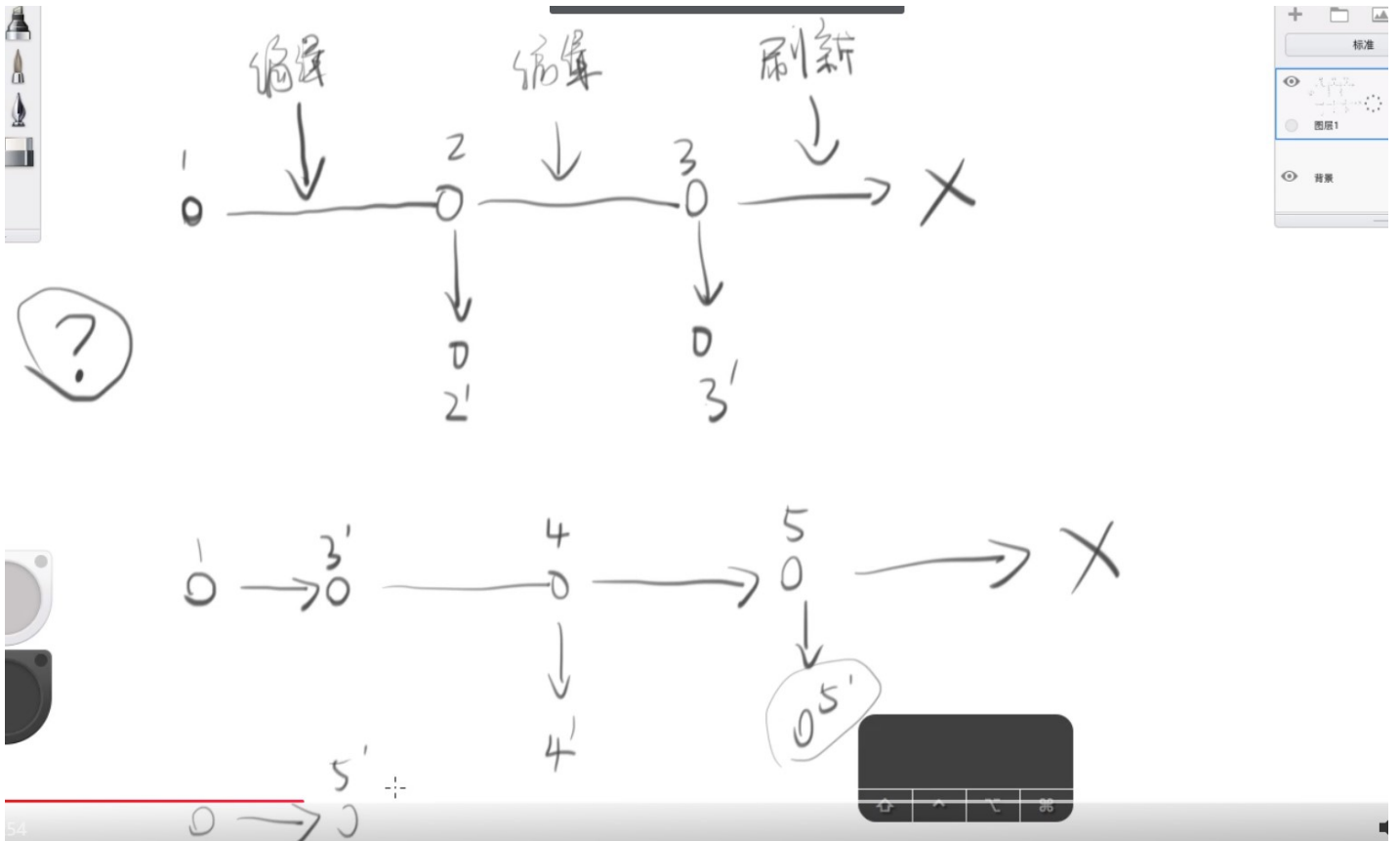
那怎么办呢,让这个变量逃离js即可;



hash的变化永远是因为用户的编辑产生的;刷新就没了;



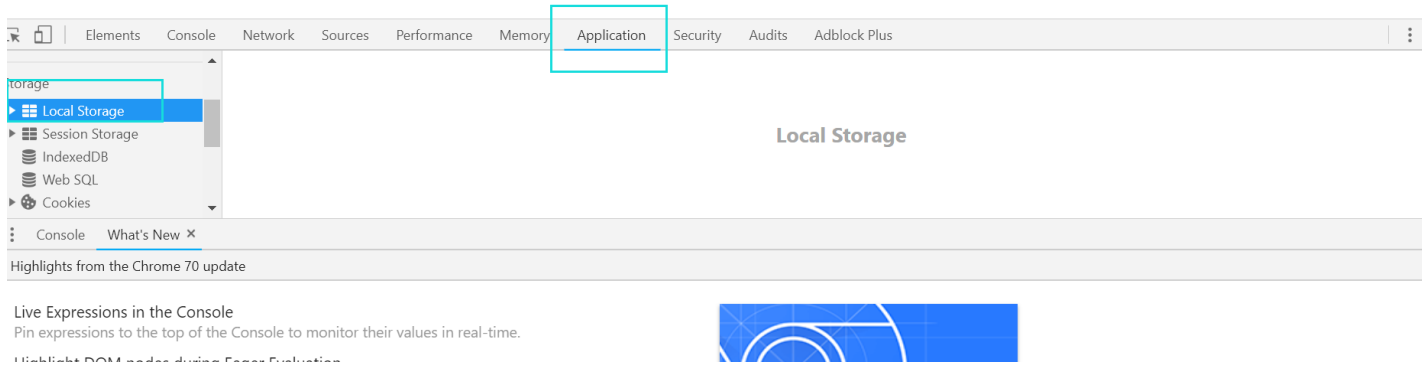
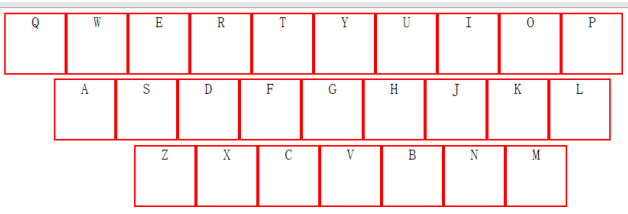
思路:



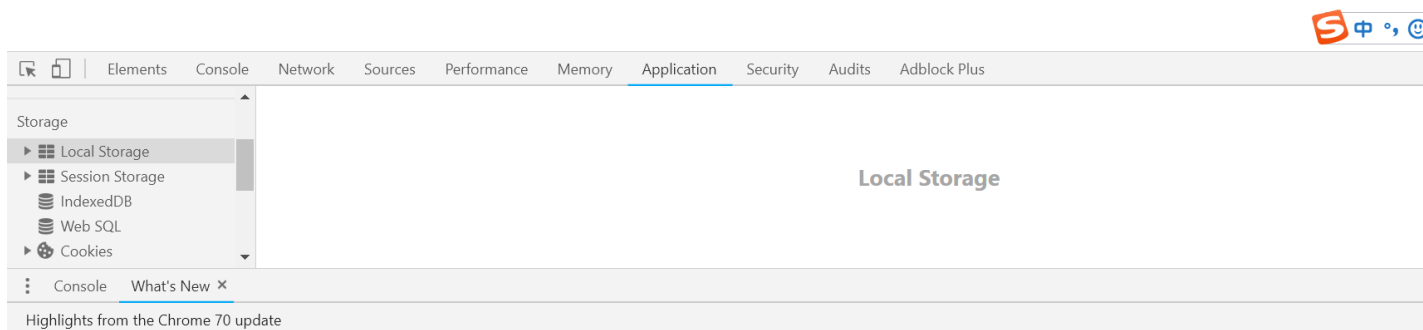
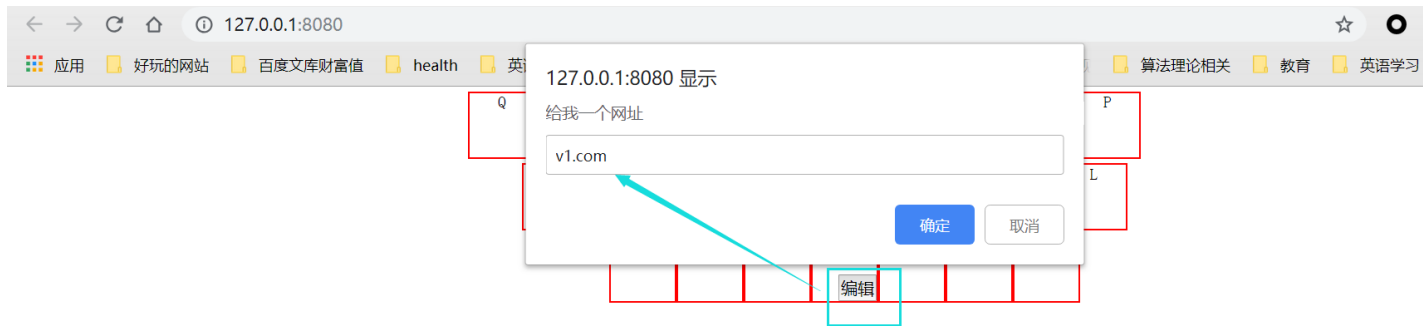
localstage里面可以存很多东西,每个东西可以分配一个桶里,zzz就是某一个桶的名字;

```
JS main.js # style.css <> index.html x
48 console.log(row) //这是打印,类似于print
49 index2 = 0
50 while(index2 < row['length']){//第一行10个 第二行9个 第三行7个;第二层循环,每次循环的时候创建一个kbd,
51 kbdxxx = document.createElement('kbd')//
52 kbdxxx.textContent = row[index2] //kbdxxx里面的文本内容是
53 buttonX = document.createElement('button')
54 buttonX.textContent = '编辑'
55 buttonX.id = row[index2]
56 buttonX.onclick = function(xzkjcnx1kcjlk){
57     key = xzkjcnx1kcjlk['target']['id'] //q w e r t
58     x = prompt('给我一个网址')
59     hash[key] = x//hash变更
60     localStorage.setItem('zzz',JSON.stringify(hash))//只要Hash变了,就把hash存到zzz里面;
61     console.log(hash)
62 }
63 kbdxxx.appendChild(buttonX)
64 divxxx.appendChild(kbdxxx)
65 index2 = index2 +1
66 }
67
68
69 index = index + 1
```

可以看看一个local Storage:

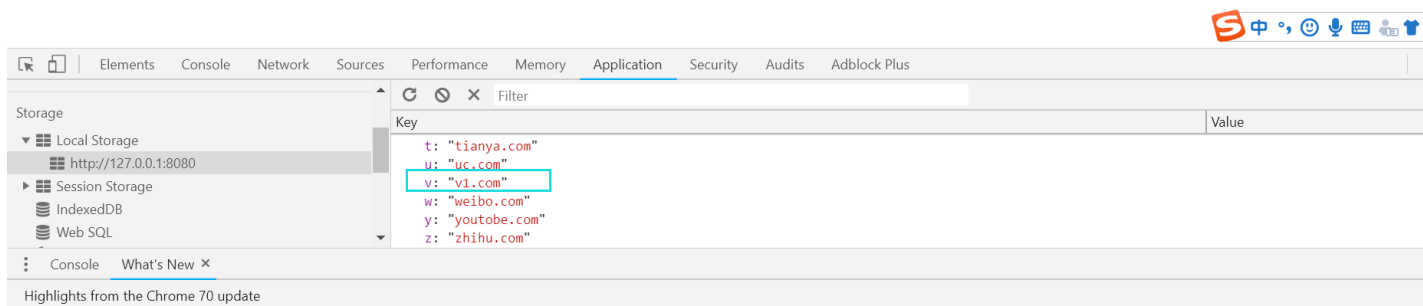
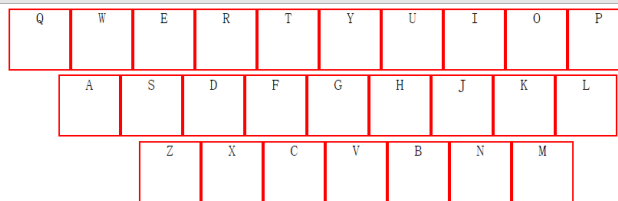


编辑一下:

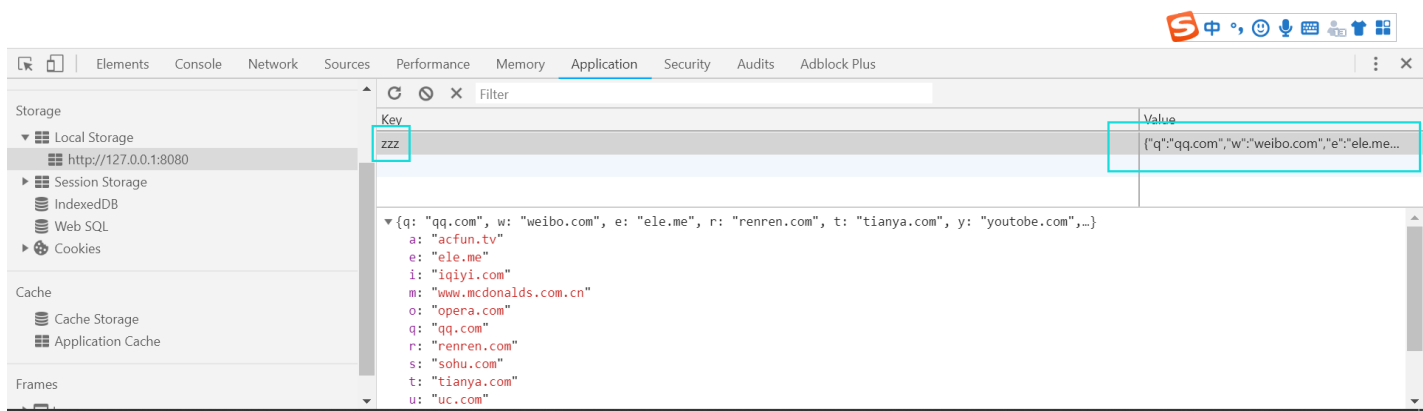
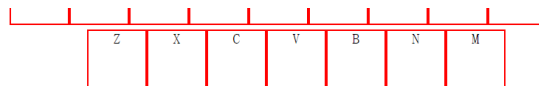


Live Expressions in the Console
Pin expressions to the top of the Console to monitor their values in real-time.

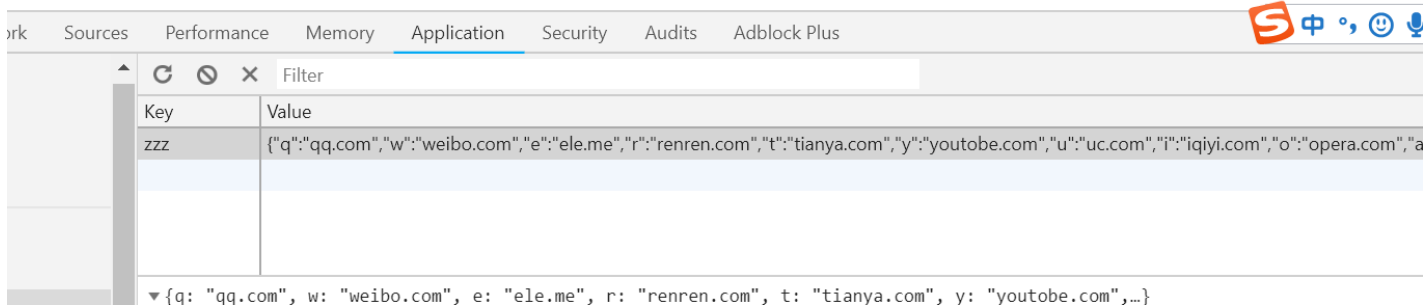
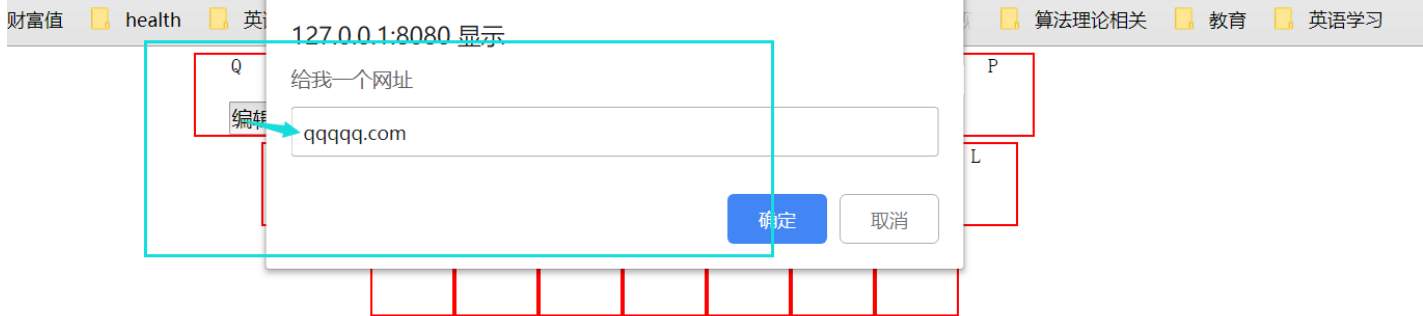
可以发现下面多了一个v1.com



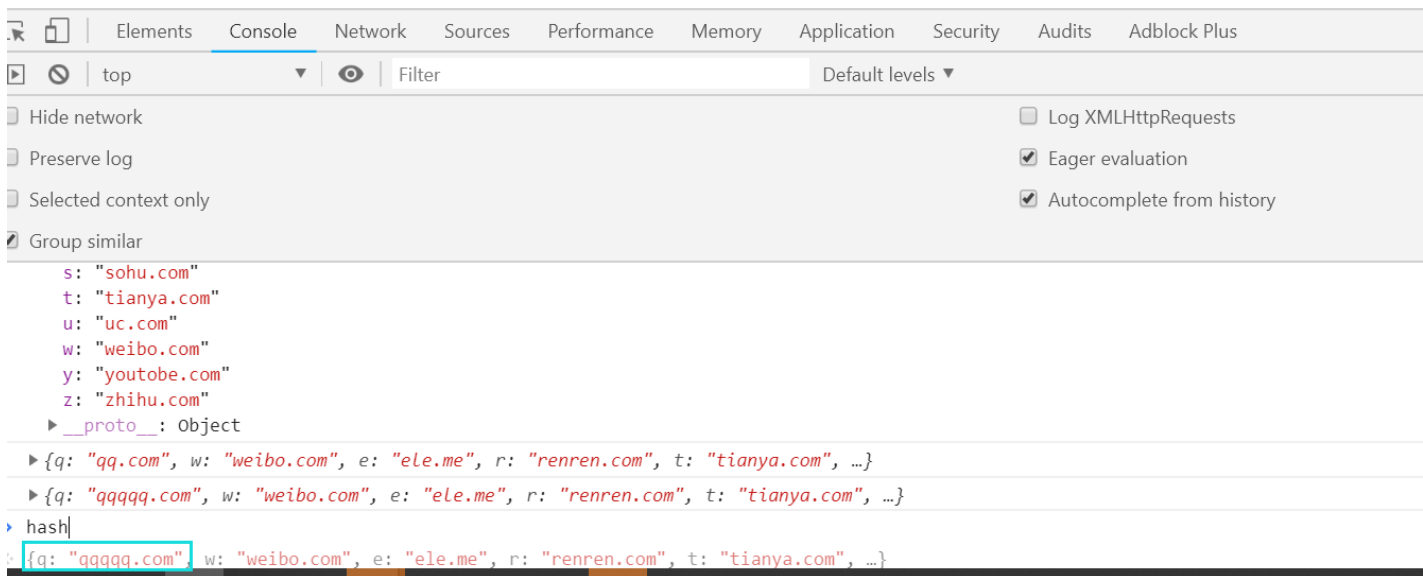
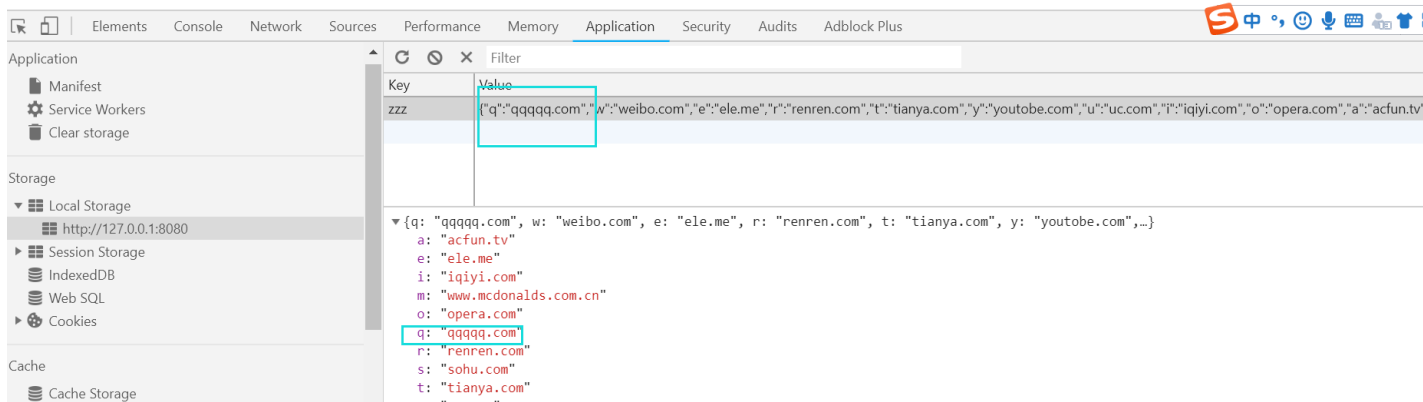
同时出现 zzz 了；



`localStorage.setItem('zzz', JSON.stringify(hash))` 的意思是可以让localStorage里面出现一个名叫 zzz 的桶,桶里面的内容叫hash; 把js的一个变量存到浏览器里面了;

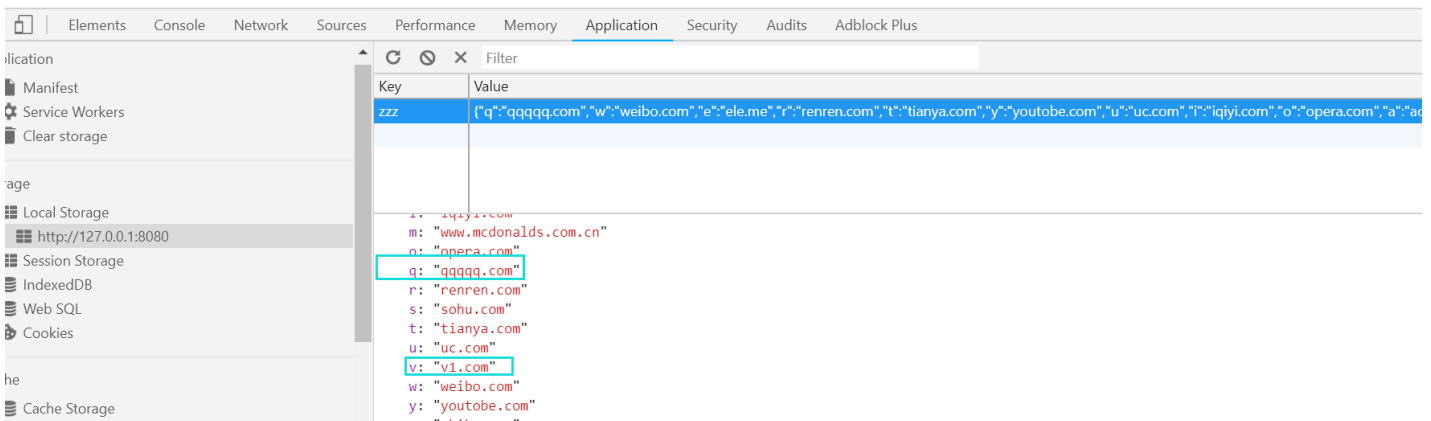


可以看到变化了:



覆盖原来的hash

```
File Edit Selection View Go Debug Terminal Help index.html - Visual Studio Code
JS main.js # style.css index.html x
38 z: 'zhidu.com',
39 m: 'www.mcdonalds.com.cn'
40
41 }
42 //取出localStorage中的zzz对应的hash
43 var hashInLocalStorage = JSON.parse(localStorage.getItem('zzz') || '')
44 if (hashInLocalStorage){
45     hash = hashInLocalStorage
46 }
47 //遍历keys,生成kbd标签
48 index = 0
49 while(index < keys['length']){ //0 1 2, 这是第一层循环,生成三个div
50     divxxx = document.createElement('div')
51     mainxxx.appendChild(divxxx)
52     row = keys[index] //第一个数组 第二个数组 第三个数组
53     console.log(row) //这是打印,类似于print
54     index2 = 0
55     while(index2 < row['length']){//第一行10个 第二行9个 第三行7个;第二层循环,每次循环的时候创建一个
56         kbdxxx = document.createElement('kbd')//
57         kbdxxx.textContent = row[index2] //kbdxxx里面的文本内容是
58         buttonX = document.createElement('button')
59         buttonX.textContent = '编辑'
60         buttonX.id = row[index2]
61         buttonX.onclick = function(xzkjcnx1kcjlk){
62             key = xzkjcnx1kcjlk['target']['id'] //q w e r t y u i o p
```

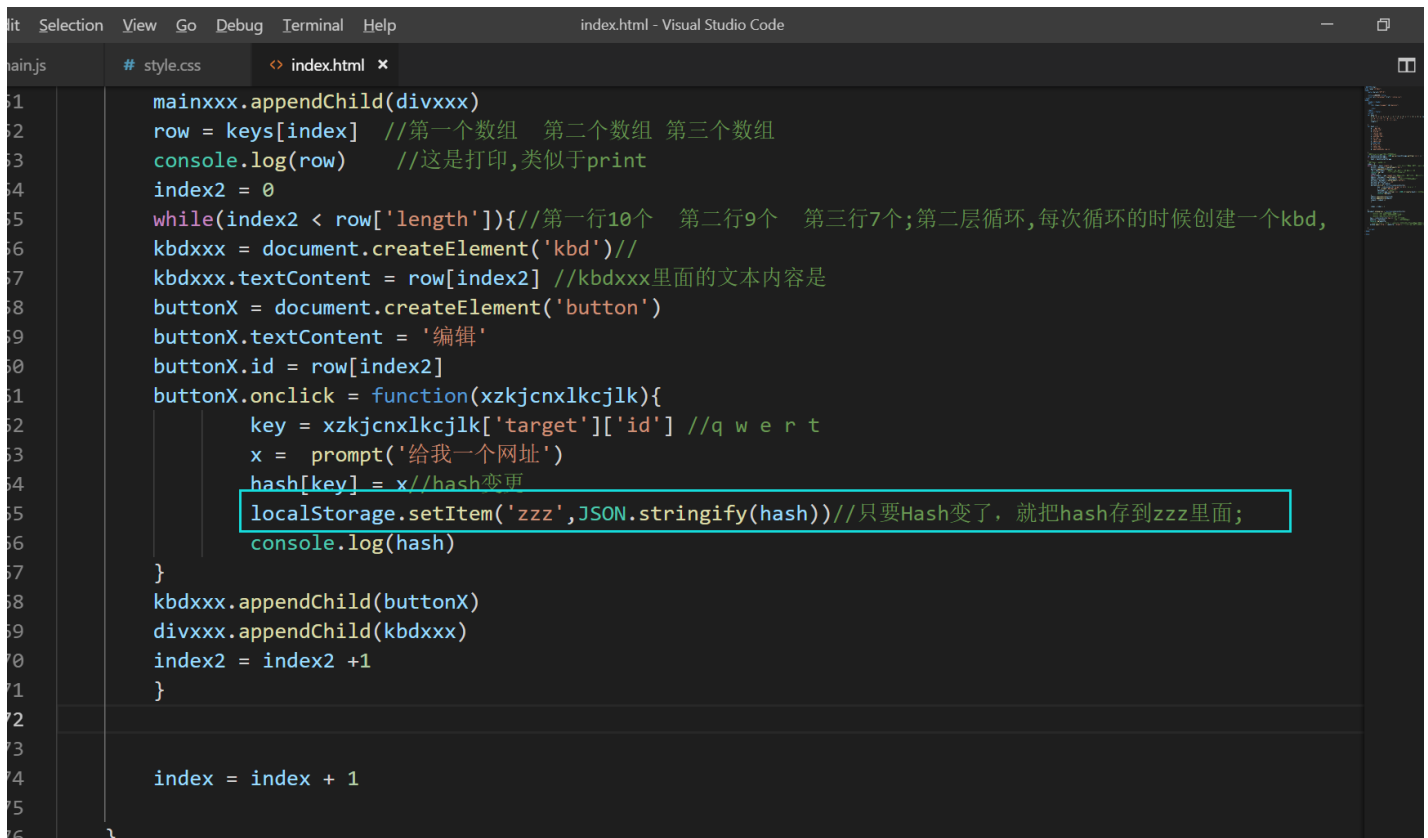


检测有没有之前留着的hash，有的话,就把之前留着的hash从localStorage里面取出来,如果不是空的,就覆盖我的hash；

读取进度就此可以做到；

什么时候可以保存进度呢；

一旦发现你过了一关,你就可以存个档案;(就是之前hash变更那里可以存档)



```
61 mainxxx.appendChild(divxxx)
62 row = keys[index] //第一个数组 第二个数组 第三个数组
63 console.log(row) //这是打印,类似于print
64 index2 = 0
65 while(index2 < row['length']){//第一行10个 第二行9个 第三行7个;第二层循环,每次循环的时候创建一个kbd,
66 kbdxxx = document.createElement('kbd')//
67 kbdxxx.textContent = row[index2] //kbdxxx里面的文本内容是
68 buttonX = document.createElement('button')
69 buttonX.textContent = '编辑'
70 buttonX.id = row[index2]
71 buttonX.onclick = function(xzkjcnx1kcjlk){
72     key = xzkjcnx1kcjlk['target']['id'] //q w e r t
73     x = prompt('给我一个网址')
74     hash[key] = x //hash变更
75     localStorage.setItem('zzz', JSON.stringify(hash))//只要Hash变了,就把hash存到zzz里面;
76     console.log(hash)
77 }
78 kbdxxx.appendChild(buttonX)
79 divxxx.appendChild(kbdxxx)
80 index2 = index2 + 1
81 }
82
83 index = index + 1
```

总结

先声明一个二维数组,而后在声明一个hash,发现如果存过一个hash,就把之前的一个hash覆盖掉;
遍历key,生成kbd便签;每个kbd标签里面添加按钮;同时在按钮上面添加id; 在按钮被点击的时候,去包含所有时间信息的hash里面去取target; 这个target就是用户点击的元素;



```
15 buttonX = document.createElement('button')
16 buttonX.textContent = '编辑'
17 buttonX.id = row[index2]
18 buttonX.onclick = function(xzkjcnx1kcjlk){
19     // xzkjcnx1kcjlk['target'] 就是用户点击的元素
20     key = xzkjcnx1kcjlk['target']['id'] // q w e r t
21     x = prompt('给我一个网址') // qq.com
22     hash[key] = x // hash 变更
23     localStorage.setItem('zzz', JSON.stringify(hash))
24     console.log(hash)
25 }
26 kbdXXXXX.appendChild(buttonX)
```

然后这个值只要去取它的id就是用户点击的东西;
而后用户输入一个网址,只要用户已输入这个网址,我就改变这个hash; 只要hash一改变,我就存档一下(方便用户刷新的时候读档);而后用户在点击某个按钮的时候打开hash里面对应的网址;
同时要求是新标签打开(window.open);

一个bug的解决

```

}
//取出localStorage中的zzz对应的hash
hashInLocalStorage = JSON.parse(localStorage.getItem('zzz') || 'null')
if (hashInLocalStorage){
    hash = hashInLocalStorage
}
//遍历keys,生成kbd标签
index = 0
while(index < keys['length']){          //0 1 2, 这是第一层循环,生成三个div
    divxxx = document.createElement('div')

```

完整代码:

```

1 <!DOCTYPE html>
2 <html lang="zh-Hans">
3 <head>
4 <meta charset="UTF-8">
5 <title>我的导航</title>
6 <link rel="stylesheet" href="./style.css">
7 </head>
8 <body>
9 <header></header>
10 <main >
11 <div class="wrapper" id="mainxxx">
12
13 </div>
14 </main>
15 <footer></footer>
16 <script>
17 keys ={
18 0:{0:'q',1:'w',2:'e',3:'r',4:'t',5:'y',6:'u',7:'i',8:'o',9:'p',length:10},
19 1:['a','s','d','f','g','h','j','k','l'],
20 2:['z','x','c','v','b','n','m'],
21 length:3
22
23 }
24 hash = {
25 q : 'qq.com',
26 w: 'weibo.com',
27 e: 'ele.me',
28 r: 'renren.com',
29 t: 'tianya.com',
30 y: 'youtube.com',
31 u: 'uc.com',
32 i: 'iqiyi.com',
33 o: 'opera.com',

```



```
34 p:undefined,
35 a:'acfun.tv',
36 s:'sohu.com',
37 z:'zhihu.com',
38 m:'www.mcdonalds.com.cn'
39
40 }
41 //取出localStorage中的zzz对应的hash
42 hashInLocalStorage = JSON.parse(localStorage.getItem('zzz') || 'null')
43 if (hashInLocalStorage){
44 hash = hashInLocalStorage
45 }
46 //遍历keys,生成kbd标签
47 index = 0
48 while(index < keys['length']){ //0 1 2, 这是第一层循环,生成三个div
49 divxxx = document.createElement('div')
50 mainxxx.appendChild(divxxx)
51 row = keys[index] //第一个数组 第二个数组 第三个数组
52 console.log(row) //这是打印,类似于print
53 index2 = 0
54 while(index2 < row['length']){//第一行10个 第二行9个 第三行7个;第二层循环,每次循环的时候创建一个
55 kbdxxx = document.createElement('kbd')//
56 kbdxxx.textContent = row[index2] //kbdxxx里面的文本内容是
57 buttonX = document.createElement('button')
58 buttonX.textContent = '编辑'
59 buttonX.id = row[index2]
60 buttonX.onclick = function(xzkjcnxlkjlk){
61 key = xzkjcnxlkjlk['target']['id'] //q w e r t
62 x = prompt('给我一个网址')
63 hash[key] = x//hash变更
64 localStorage.setItem('zzz',JSON.stringify(hash))//只要Hash变了,就把hash存到zzz里面;
65 console.log(hash)
66 }
67 kbdxxx.appendChild(buttonX)
68 divxxx.appendChild(kbdxxx)
69 index2 = index2 +1
70 }
71 index = index + 1
72
73 }
74 document.onkeypress = function(xzkjcnxlkjlk){
75 // console.log('我发现你输入了一个键')
76 // console.log('你按键的时候的所有信息是')
77 // console.log(xzkjcnxlkjlk['key'])
78 key = (xzkjcnxlkjlk['key'])//拿到用户按的键
79 website = hash[key] //得到键所对应的网站
```

```

80 console.log(website) //
81 // location.href = 'http://' + website // 把当前地址变成新的网站的地址，模拟用户在地址栏输入，
82 window.open('http://' + website, '_blank') // 模拟用户新开页面，在那开呢，在blank
83
84 }
85 </script>
86 </body>
87
88 </html>

```

style.css

```

1  *{margin: 0;padding: 0;}
2  a{text-decoration: none;}
3  kbd{border: 1px solid red;width: 4em;height: 4em;display: inline-block;
4  text-transform: uppercase;
5  position: relative;
6  }
7  kbd > button {
8  position: absolute;
9  right: 0;
10 bottom:0;
11 display: none;
12 }
13 kbd:hover > button{
14 display: inline-block;
15 }
16
17 main{
18 text-align: center;
19 }
20 #mainxxx{
21 display: inline-block;
22 }
23 #mainxxx > div:nth-child(2){
24 margin-left: 2em;
25 }
26 #mainxxx > div:nth-child(3){
27 margin-left: 4em;
28 }

```