

取颜色: (键盘颜色)

怎么做渐变色

设置字体颜色

设置键盘边框

键盘边框

view-source:<https://richard1230.github.io/nav-demo/>

```
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](https://richard1230.github.io/nav-demo/sty
7  </head> |
8  <body> |
9  <header></header> |
10 <main > |
11 <div class="wrapper" id="mainxxx"> |
12 </div> |
13 </main> |
14 <footer></footer> |
15 <script> |
16 keys = { |
17 0: {0: 'q', 1: 'w', 2: 'e', 3: 'r', 4: 't', 5: 'y', 6: 'u', 7: 'i', 8: 'o', 9: 'p', length: 10}, |
18 1: ['a', 's', 'd', 'f', 'g', 'h', 'j', 'k', 'l'], |
19 2: ['z', 'x', 'c', 'v', 'b', 'n', 'm'], |
20 length: 3 |
21 } |
22 hash = { |
23 q: 'qq.com', |
24 w: 'weibo.com', |
25 e: 'ele.me', |
26 r: 'renren.com', |
27 t: 'tianya.com', |
28 y: 'youtube.com', |
29 u: 'uc.com', |
30 i: 'iqiyi.com', |
31 o: 'opera.com', |
32 p: undefined, |
33 a: 'acfun.tv', |
34 s: 'sohu.com', |
35 z: 'zhihu.com', |
```

```
36 m: 'www.mcdonalds.com.cn' |
37 } |
38 //取出localStorage中的zzz对应的hash |
39 hashInLocalStorage = JSON.parse(localStorage.getItem('zzz') || 'null') |
40 if (hashInLocalStorage){ |
41 hash = hashInLocalStorage |
42 } |
43 //遍历keys,生成kbd标签 |
44 index = 0 |
45 while(index < keys['length']){ //0 1 2, 这是第一层循环,生成三个div |
46 divxxx = document.createElement('div') |
47 mainxxx.appendChild(divxxx) |
48 row = keys[index] //第一个数组 第二个数组 第三个数组 |
49 console.log(row) //这是打印,类似于print |
50 index2 = 0 |
51 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
52 kbdxxx = document.createElement('kbd')// |
53 kbdxxx.textContent = row[index2] //kbdxxx里面的文本内容是 |
54 buttonX = document.createElement('button') |
55 buttonX.textContent = '编辑' |
56 buttonX.id = row[index2] |
57 buttonX.onclick = function(xzkjcnx1kcjlk){ |
58 key = xzkjcnx1kcjlk['target']['id'] //q w e r t |
59 x = prompt('给我一个网址') |
60 hash[key] = x//hash变更 |
61 localStorage.setItem('zzz',JSON.stringify(hash))//只要Hash变了,就把hash存到zzz里面; |
62 console.log(hash) |
63 } |
64 kbdxxx.appendChild(buttonX) |
65 divxxx.appendChild(kbdxxx) |
66 index2 = index2 +1 |
67 } |
68 index = index + 1 |
69 } |
70 document.onkeypress = function(xzkjcnx1kcjlk){ |
71 // console.log('我发现你输入了一个键') |
72 // console.log('你按键的时候的所有信息是') |
73 // console.log(xzkjcnx1kcjlk['key']) |
74 key = (xzkjcnx1kcjlk['key'])//拿到用户按的键 |
75 website = hash[key] //得到键所对应的网站 |
76 console.log(website) // |
77 // 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, |
78 window.open('http://'+website,'_blank')//模拟用户新开页面,在那开呢,在blank |
79 } |
80 </script> |
81 </body> |
```

有个head, 有个main, main里面有个 **wrapper**, 里面id是乱叫的, 这里为 **mainxxx**, 还有个footer;  
 目前header和footer是空的, 只用到了main;  
 将keys放到二维数组里面了, 然后把网站放到hash里面了;

```
// 取出 localStorage 中的 zzz 对应的 hash
hashInLocalStorage = JSON.parse(localStorage.getItem('zzz') || 'null')
if(hashInLocalStorage){
  hash = hashInLocalStorage
}
```

每次都会去本地localStorage把hash取出来覆盖我们的hash, 如果localStorage里面的hash是存在的, 那么就把当前的hash赋值为localStorage里面的hash;

02:56

接着就是两层循环:

会生成3行键盘;

```
// 遍历 keys, 生成 kbd 标签
index = 0
while(index < keys['length']){ // 0 1 2
  divXXXXX = document.createElement('div')
  mainXXXXX.appendChild(divXXXXX)
  row = keys[index] // 第一个数组 第二个数组 第三个数组
  index2 = 0
  while(index2 < row['length']){ // 0~9 0~8 0~6
    kbdXXXXX = document.createElement('kbd')
    kbdXXXXX.textContent = row[index2]
    buttonX = document.createElement('button')
    buttonX.textContent = '编辑'
    buttonX.id = row[index2]
    buttonX.onclick = function(xzkjcnxlcjlk){
      // xzkjcnxlcjlk['target'] 就是用户点击的元素
      key = xzkjcnxlcjlk['target']['id'] // q w e r t
      x = prompt('给我一个网址') // qq.com
      hash[key] = x // hash 变更
      localStorage.setItem('zzz', JSON.stringify(hash))
      console.log(hash)
    }
    kbdXXXXX.appendChild(buttonX)
    divXXXXX.appendChild(kbdXXXXX)
    index2 = index2 + 1
  }
}
```

每行键盘我会放一个kbd标签和buttonx标签

```

0 // 遍历 keys, 生成 kbd 标签
1 index = 0
2 while(index < keys['length']){ // 0 1 2
3     divXXXXX = document.createElement('div')
4     mainXXXXX.appendChild(divXXXXX)
5     row = keys[index] // 第一个数组 第二个数组 第三个数组
6     index2 = 0
7     while(index2 < row['length']){ // 0~9 0~8 0~6
8         kbdXXXXX = document.createElement('kbd')
9         kbdXXXXX.textContent = row[index2]
10        buttonX = document.createElement('button')
11        buttonX.textContent = '编辑'
12        buttonX.id = row[index2]
13        buttonX.onclick = function(xzkjcnxlcjlk){
14            // xzkjcnxlcjlk['target'] 就是用户点击的元素
15            key = xzkjcnxlcjlk['target']['id'] // q w e r t
16            x = prompt('给我一个网址') // qq.com
17            hash[key] = x // hash 变更
18            localStorage.setItem('zzz', JSON.stringify(hash))
19            console.log(hash)
20        }
21        kbdXXXXX.appendChild(buttonX)
22        divXXXXX.appendChild(kbdXXXXX)
23        index2 = index2 + 1
24    }
25    index = index+1
26 }

```

上面是生成键盘的代码;

```

document.onkeypress = function(xzkjcnxlcjlk){
    key = xzkjcnxlcjlk['key'] // q w e
    website = hash[key]
    //location.href = 'http://' + website
    window.open('http://' + website, '_blank')
}

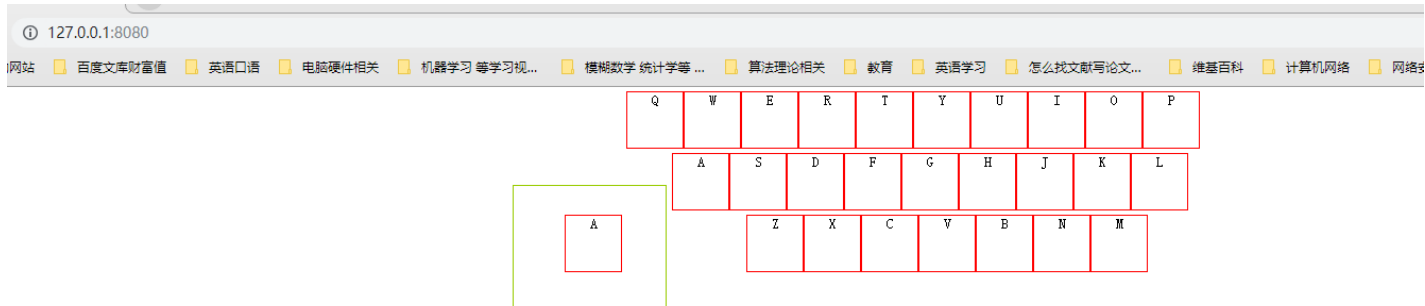
```

监听onkeypress事件;如果发现一个key有一个网址的话,就会在新页面打开这个网址;  
(就是你点击q, 会打开一个网址,如果你有页面定义的话,如果没有定义会出现问题)

7:30

```
index.html x
3 <head>
4   <meta charset="UTF-8">
5
6   <title>我的导航</title>
7   <link rel="stylesheet" href="./style.css">
8 </head>
9 <body>
10   <header></header>
11   <main >
12     <kbd>A</kbd>
13     <div class="wrapper" id="mainxxx">
14
15     </div>
16   </main>
17 <footer></footer>
18 <script>
19   keys ={
20     0:{0:'q',1:'w',2:'e',3:'r',4:'t',5:'y',6:'u',7:'i',8:'o',9:'p',length:10},
21     1:['a','s','d','f','g','h','j','k','l'],
22     2:['z','x','c','v','b','n','m'],
23     length:3
24   }
25 }
```

添加了个 `<kbd>A</kbd>`

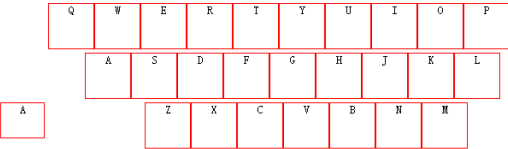


继续修改尺寸:

File Edit Selection View Go Debug Terminal Helpindex.html - Visual Studio Code

index.html x

```
3 <head>
4   <meta charset="UTF-8">
5
6   <title>我的导航</title>
7   <link rel="stylesheet" href="./style.css">
8 </head>
9 <body>
10  <header></header>
11  <main >
12    <style>
13      .key{
14        width: 50px;
15        height: 40px;
16      }
17    </style>
18
19    <kbd class='key'>A</kbd>
20    <div class="wrapper" id="mainxxx">
21
22    </div>
23  </main>
24  <footer></footer>
25  <script>
26    keys ={
27      0:{0:'q',1:'w',2:'e',3:'r',4:'t',5:'y',6:'u',7:'i',8:'o',9:'p',length:10}
```



Elements Console Network Sources

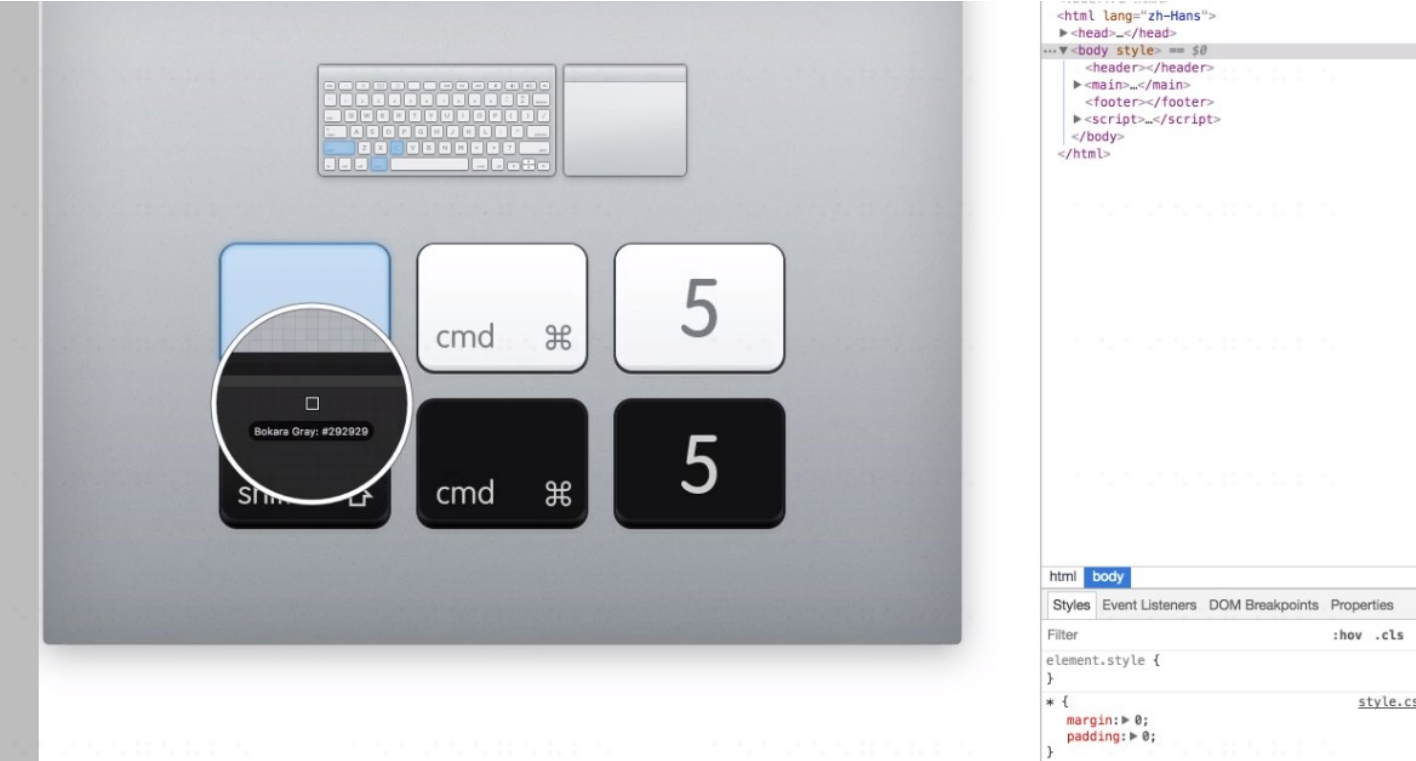
```
<!doctype html>
<html lang="zh-Hans">
  <head>...</head>
  <body>
    <header></header>
    <main>
      <style></style>
      <div class="wrapper" id="mainxxx">...</div>
    </main>
    <footer></footer>
  </body>
</html>
```

html body main kbd.key

Styles	Event Listeners	DOM Breakpoints	Properties	Accessibility
Filter			:hov .cls +	0
element.style {				in -
.key {			(index):13	order 1
width: 50px;				padding-
height: 40px;				50 x 40
kbd {			style.css:3	-
border: 1px solid red;				1
width: 40px;				-
height: 40px;				-
display: inline-block;				align-co...
text-transform: uppercase;				align-it...
position: relative;				align-se...
* {			style.css:1	alignmen...
margin: 0;				all
padding: 0;				animatio...
tt, code, kbd, samp {			user agent stylesheet	animatio...
font-family: monospace;				animatio...
Inherited from main				animatio...
main {			style.css:17	animatio...

取颜色: (键盘颜色)

mac里面用sip, 也可以用qq:



这里是渐变色,取最上面的黑色以及最下面的一个黑色;

### 怎么做渐变色

e

css gradient generator



全部

图片

视频

新闻

图书

更多

设置

工具

找到约 19,100,000 条结果 (用时 0.73 秒)

Ultimate CSS Gradient Generator - ColorZilla.com

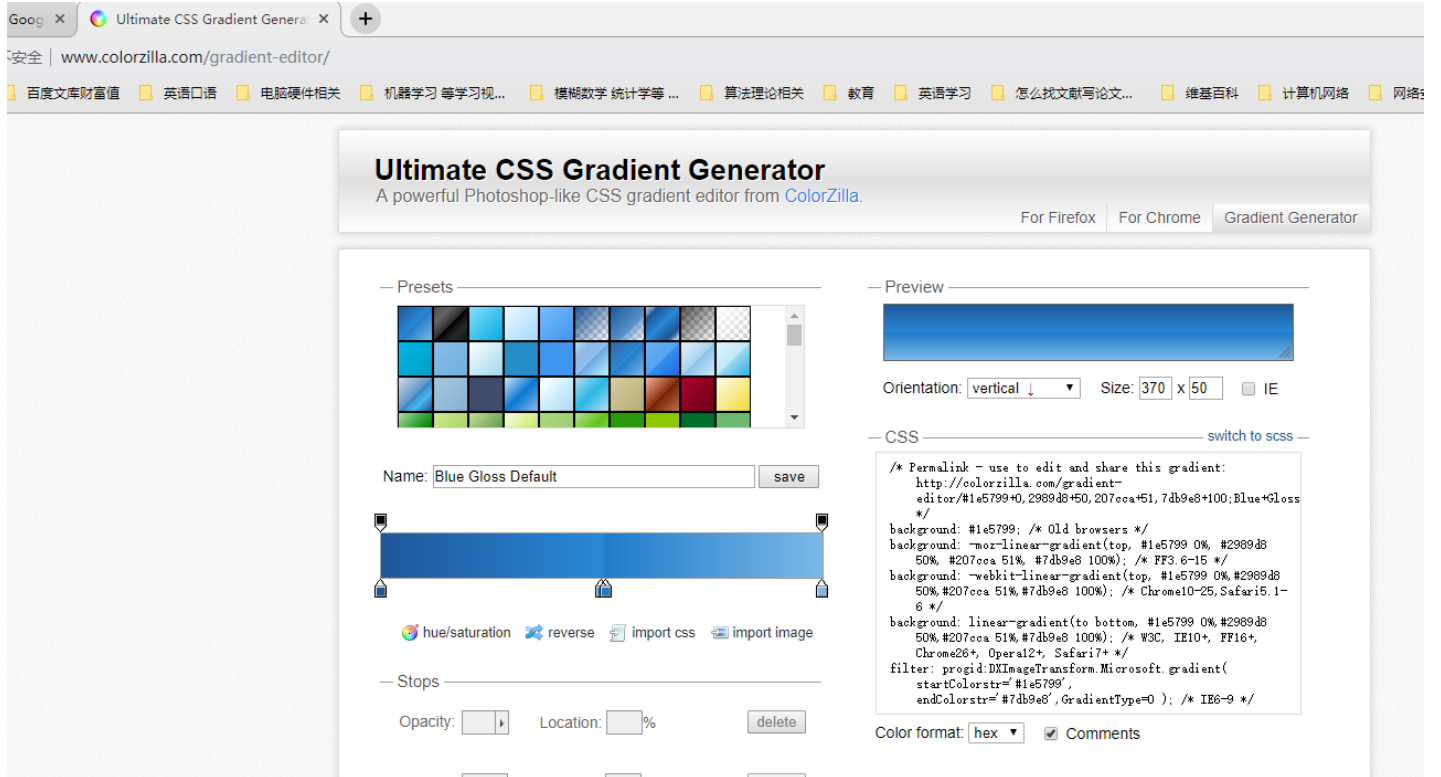
[www.colorzilla.com/gradient-editor/](http://www.colorzilla.com/gradient-editor/) ▼ 翻译此页

What's new. » Don't generate older Opera, IE10 Preview, Webkit syntax by default (turn back on using "Maximize compatibility" mode); » Browser Compatibility ...

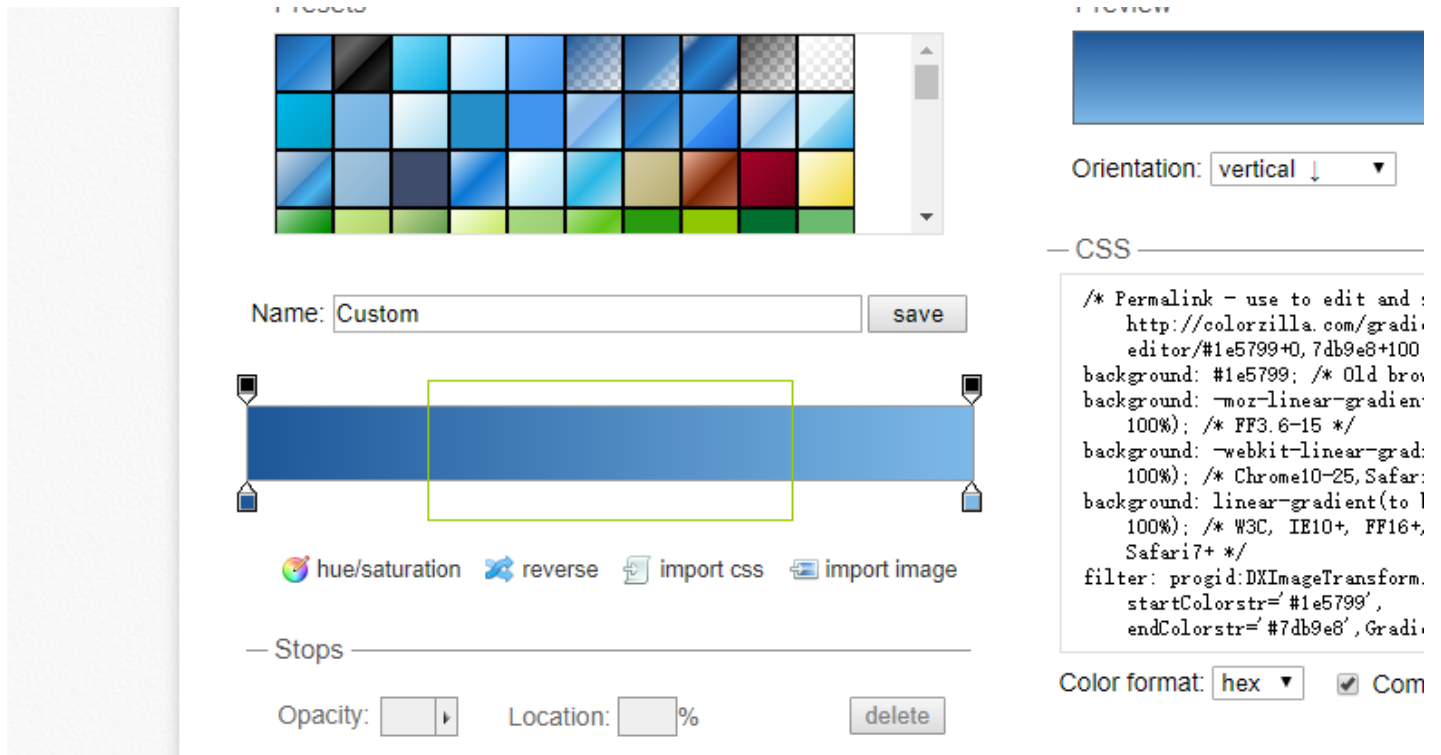
Gradient CSS Generator | CSSmatic

<https://www.cssmatic.com/gradient-generator> ▼ 翻译此页





把中间两个拖掉,现在中间什么都没有了;





## Presets



Name: Custom

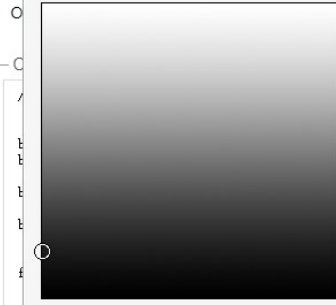
save

[hue/saturation](#) [reverse](#) [import css](#) [import image](#)

## Stops

Opacity:  Location:  % Color:  Location:  0 % 

## Preview



OK

Cancel

H: 0 °

S: 0 %

B: 16 %

R: 41

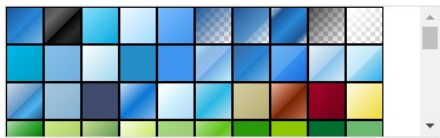
G: 41

B: 41

#: 292929

[Link to, save or share the current gradient using its permalink.](#)[Tweet](#)[Like 13K](#)[Follow @colorzilla](#)

## Presets



Name: Custom

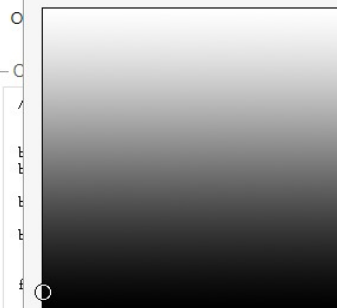
save

[hue/saturation](#) [reverse](#) [import css](#) [import image](#)

## Stops

Opacity:  Location:  % Color:  Location:  100 % 

## Preview



OK

Cancel

H: 0

S: 0 %

B: 6 %

R: 17

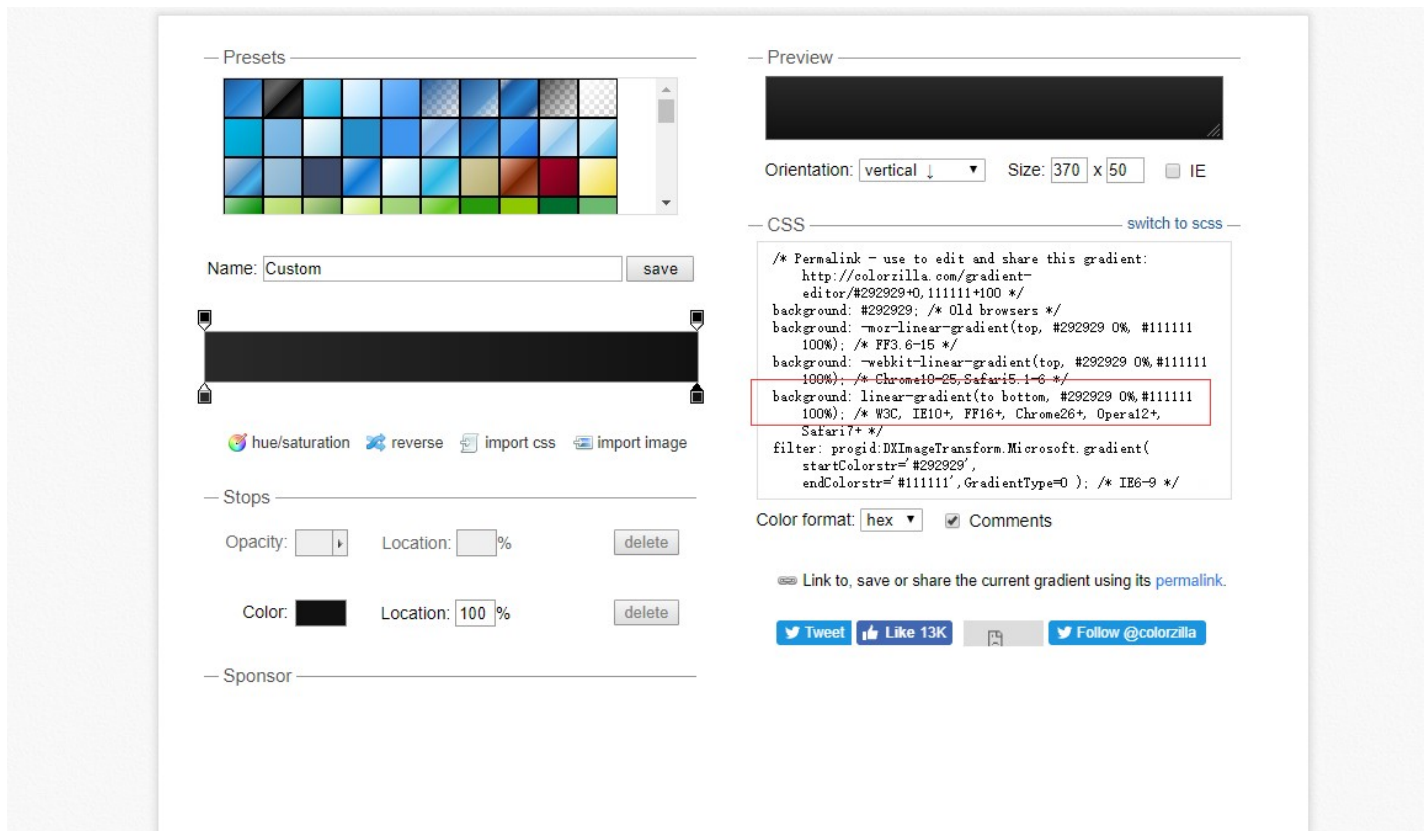
G: 17

B: 17

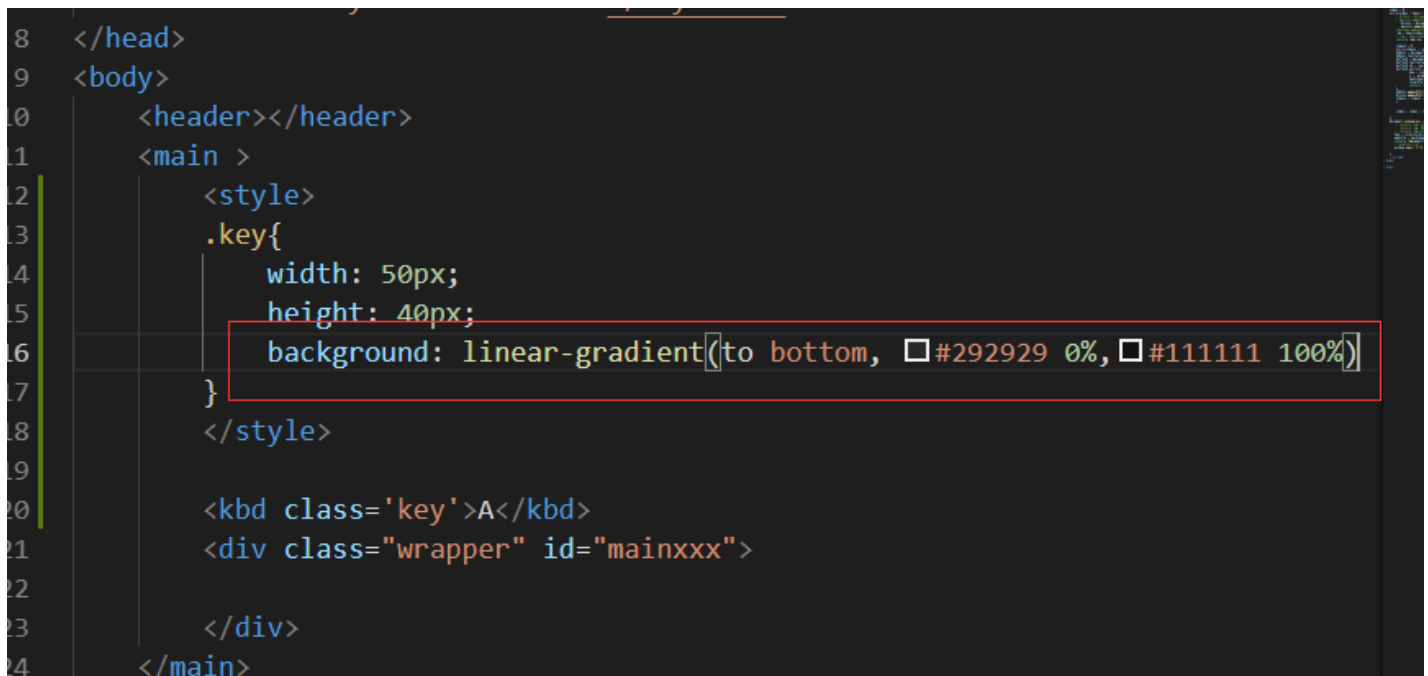
#: 111111

[Link to, save or share the current gradient using its permalink.](#)[Tweet](#)[Like 13K](#)[Follow @colorzilla](#)

复制没有前缀的这一行



复制到下面:



干掉边框:

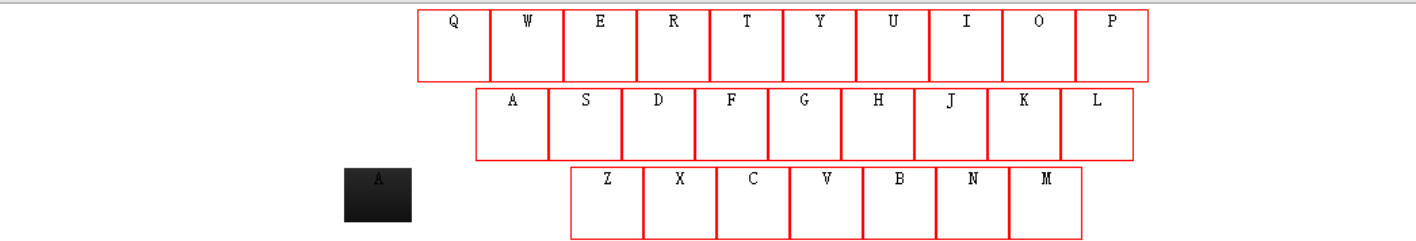
```
</head>
<body>
  <header></header>
  <main >
    <style>
      .key{
        width: 50px;
        height: 40px;
        background: linear-gradient(to bottom, #292929 0%, #111111 100%);
        border:none;
      }
    </style>

    <kbd class='key'>A</kbd>

    <div class="wrapper" id="mainxxx">

      </div>
  </main>
  <footer></footer>
  <script>
    keys ={
      0:{0:'q',1:'w',2:'e',3:'r',4:'t',5:'y',6:'u',7:'i',8:'o',9:'p',length:10},
      1:['a','s','d','f','g','h','j','k','l'],
      2:['z','x','c','v','b','n','m'],
```

看效果:

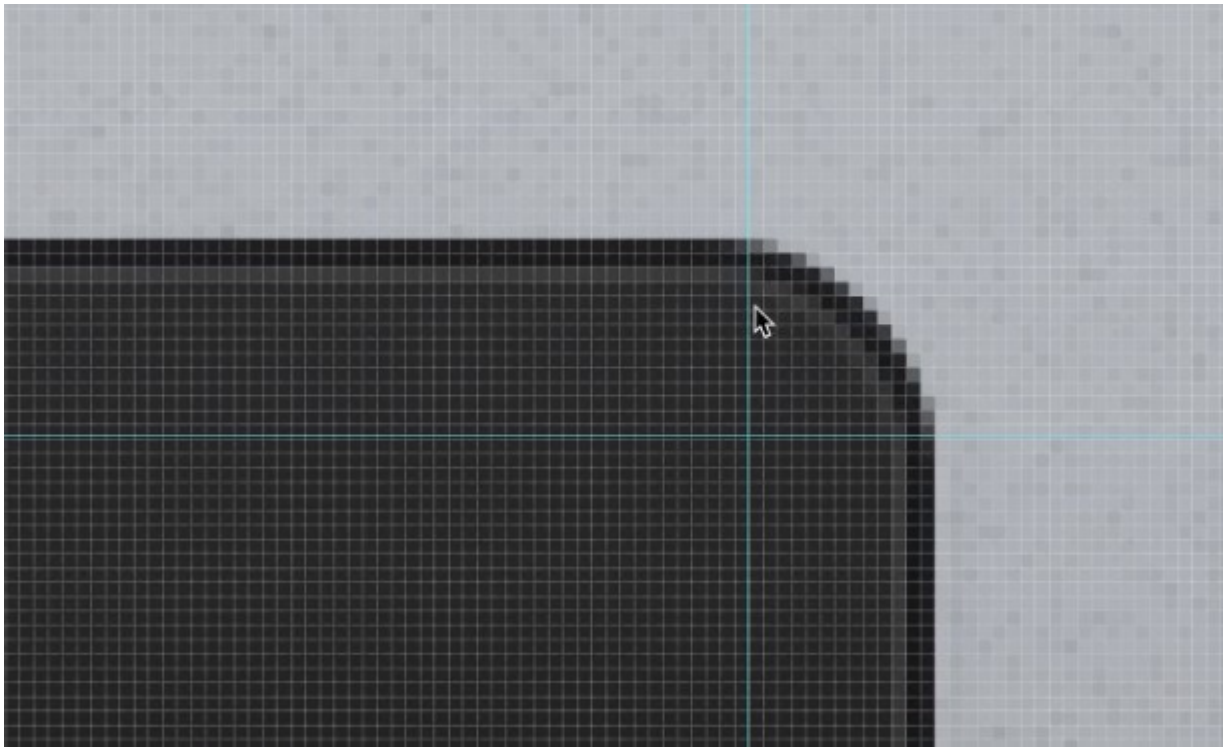


设置字体颜色



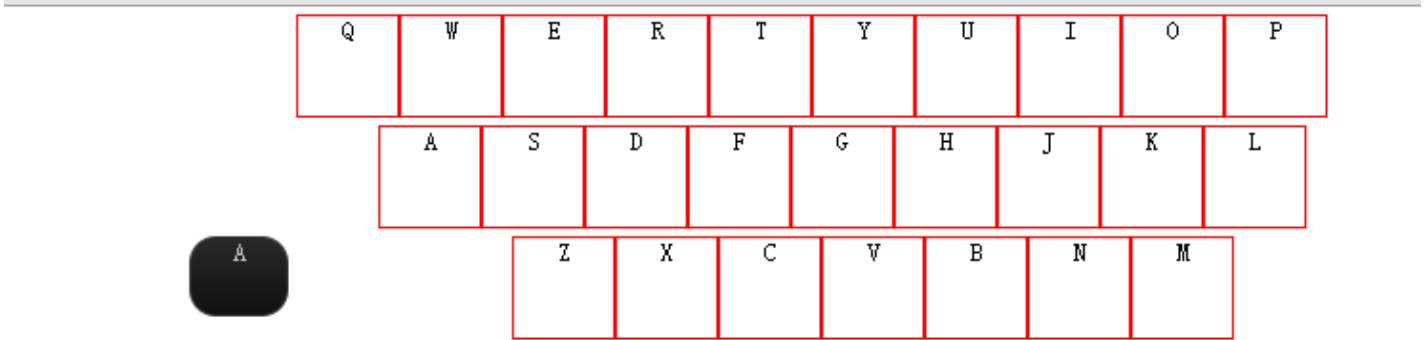
```
index.html x
3 <head>
4   <meta charset="UTF-8">
5
6   <title>我的导航</title>
7   <link rel="stylesheet" href="./style.css">
8 </head>
9 <body>
10   <header></header>
11   <main >
12     <style>
13       .key{
14         width: 50px;
15         height: 40px;
16         background: linear-gradient(to bottom, #292929 0%, #111111 100%);
17         border:none;
18         color: #C5C5C5;
19       }
20     </style>
21
22     <kbd class='key'>A</kbd>
23     <div class="wrapper" id="mainxxx">
24
25     </div>
26   </main>
27   <footer></footer>
28   <script>
```

## 设置键盘边框

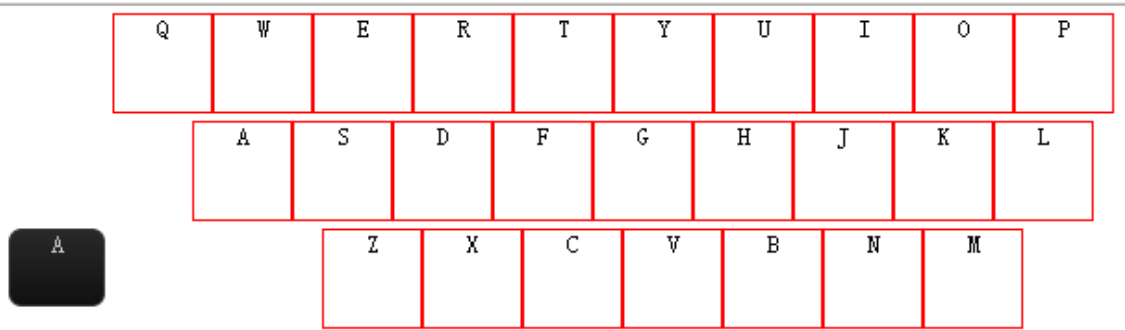


数了一下有14个格子;

```
background: linear-gradient(to bottom, #292929 0,  
border:1px solid #373737;  
color:#C5C5C5;  
border-radius:14px;  
}  
</style>  
  
<kbd class='key'>A</kbd>
```



改为7就正常了:(因为边框宽度不一样的缘故)



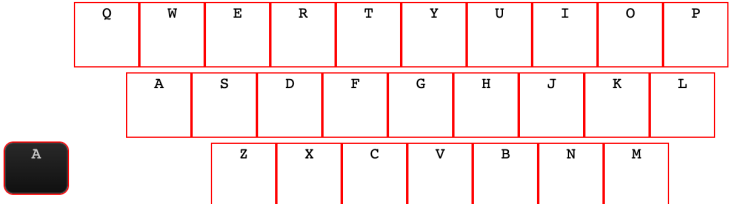
15:30

## 键盘边框

box-shadow:

← → ↻ 🏠 ⓘ 127.0.0.1:8080 ☆ ⚙️ ASP 🔴 📺 📶 🌐 🐼 🟢 📁 | 🤖

📱 应用 📁 好玩的网站 📁 百度文库财富值 📁 英语口语 📁 机器学习 等学习视... 📁 教育 📁 英语学习 📁 计算机网络 📁 计算机必上网站 🌐 QQ恢复官方网站 » 📁 其他书:



🔍 📄 Elements Console Sources Network Performance >> ⋮

```
<!doctype html>
<html lang="zh-Hans">
  <head>...</head>
  <body>
    <header></header>
    <main>
      <style>...</style>
      ...
      <kbd class="key" style="
        box-shadow: 0 0 0px 1px red;
      ">A</kbd> == $0
    <div class="wrapper" id="mainxxx">...</div>
    </main>
    <footer></footer>
    <script>...</script>
  </body>
</html>
```

html body main **kbd.key**

Styles Event Listeners DOM Breakpoints Properties Accessibility

Filter :hov .cls +

element.style {

☒ box-shadow: 0 0 0px 1px red;

;

}

.key {

(index):13

width: 50px;

height: 40px;

background: ▶ linear-gradient(to bottom,

■ #292929 0%, ■ #111111 100%);

border: ▶ 1px solid ■ #373737;

color: ■ #C5C5C5;

border-radius: ▶ 7px;

}

kbd {

style.css:3

border: ▶ 1px solid red;

}

position 0

margin -

border 1

padding 50 x 40

1 -

1 -

1

0

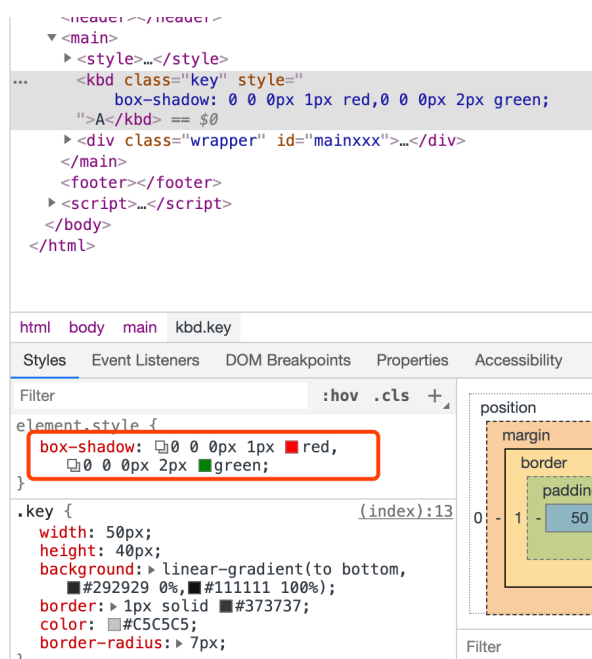
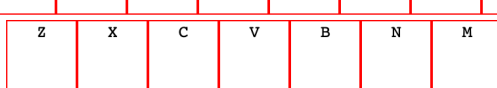
Filter

☐ Show

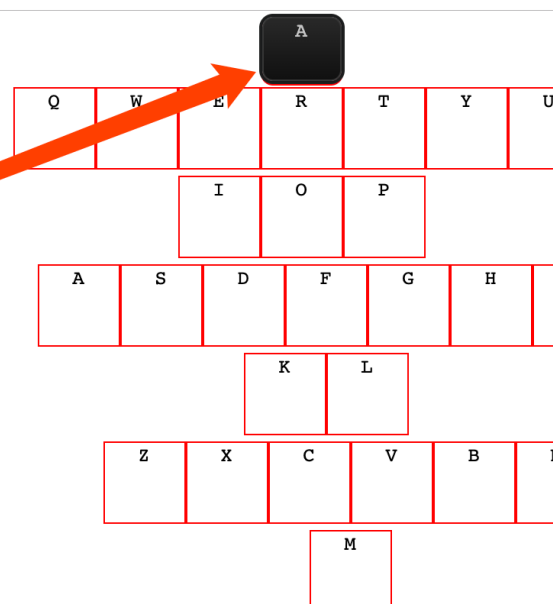
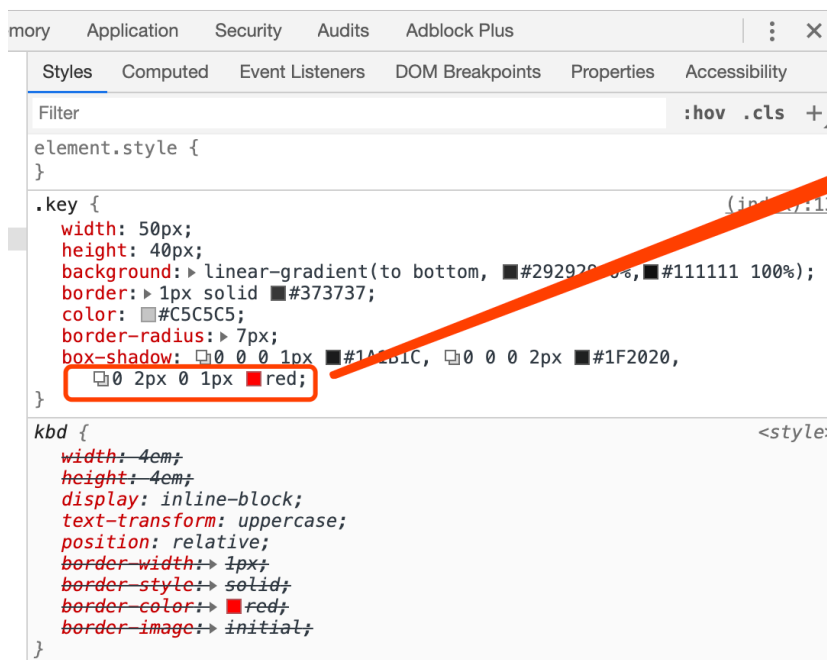
▶ background-attachment

scroll

▶ background-clip



这是键盘上的键上面的靠近外边框的一条阴影线





最后吃改成这个样子(靠近外边框的一条阴影线):

```
<title>我的导航</title>
<link rel="stylesheet" href="./style.css">
</head>
<body>
  <header></header>
  <main >
    <style>
      .key{
        width: 50px;
        height: 40px;
        background: linear-gradient(to bottom, #292929 0%, #111111 100%);
        /* border: none ;          这里是把边框干掉 */
        border: 1px solid #373737;
        color: #C5C5C5 ;          /*键盘上面的字母(比如A)的颜色*/
        border-radius: 7px;
        box-shadow: 0 0 0 1px #1A1B1C, 0 0 0 2px #1F2020, 0 3px 0 2px #080808;
      }
    </style>
    <kbd class="key">A</kbd>
```

```
<main >
  <style>
    .key{
      width: 50px;
      height: 40px;
      background: linear-gradient(to bottom, #292929 0%, #111111 100%);
      /* border: none ;          这里是把边框干掉 */
      border: 1px solid #373737;
      color: #C5C5C5 ;          /*键盘上面的字母(比如A)的颜色*/
      border-radius: 7px;
      box-shadow: 0 0 0 1px #1A1B1C, 0 0 0 2px #1F2020, 0 3px 0 2px #080808;
      display: inline-flex;
      align-items: center;
      justify-content: center;
    }
  </style>
  <kbd class="key">A</kbd>
  <div class="wrapper" id="mainxxx">
```

调试控制台 终端

筛选。例: text, \*\*/\*.ts, !\*\*/node\_m

在工作区检测到问题。

改名称(这里会出现问题):

```

console.log(row) //这是打印,类似于print
index2 = 0
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,
  kbd= document.createElement('kbd')//
  kbd.textContent = row[index2] //kbdxxx里面的文本内容是
  button = document.createElement('button')
  button.textContent = '编辑'
  button.id = row[index2] (parameter) xzkjcnxlcjlk: any
  button.onclick = function(xzkjcnxlcjlk){
    key = xzkjcnxlcjlk['target']['id'] //q w e r t
    x = prompt('给我一个网址')
    hash[key] = x//hash变更
    localStorage.setItem('zzz',JSON.stringify(hash))//只要Hash变了,就把hash存到zzz里面;
    console.log(hash)
  }
  kbd.appendChild(button)
  div.appendChild(kbd)
  index2 = index2 + 1
}

```

```

div = document.createElement('div')
main.appendChild(div)
row = keys[index] //第一个数组 第二个数组 第三个数组
console.log(row) //这是打印,类似于print
index2 = 0
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
  kbd= document.createElement('kbd')//
  kbd.textContent = row[index2] //kbdxxx里面的文本内容是
  kbd.className = 'key'
  button = document.createElement('button')
  button.textContent = '编辑'
  button.id = row[index2]
  button.onclick = function(xzkjcnxlcjlk){
    key = xzkjcnxlcjlk['target']['id'] //q w e r t
    x = prompt('给我一个网址')
    hash[key] = x//hash变更
    localStorage.setItem('zzz',JSON.stringify(hash))//只要Hash变了,就把
    console.log(hash)
  }

```