


找弟弟的几个核心代码

解决看不到作品的bug

这里为视频如何解决问题里面的！！

如何获取获取的节点是文本还是节点！



mdn node type

All

Shopping

Videos

News

Images

More

Settings

Tools

About 36,000 results (0.48 seconds)

[Node.nodeType - Web APIs | MDN](#)

<https://developer.mozilla.org> › Web technology for developers › Web APIs › Node ▼

18 Aug 2015 - The read-only `Node.nodeType` property that represents the type of the node.

[Node.nodeName - Web APIs | MDN](#)

<https://developer.mozilla.org> › Web technology for developers › Web APIs › Node ▼

22 Feb 2015 - `nodeName` read-only property returns the name of the current `node` as a string. ... The returned values for different `types` of `nodes` are: ...

[Node - Web APIs | MDN - Mozilla](#)

需要记忆:

- localName
- namespaceURI
- nextSibling
- nodeName
- nodePrincipal
- nodeType
- nodeValue
- outerText
- ownerDocument
- parentElement
- parentNode
- prefix
- previousSibling
- rootNode [我来译!]
- textContent
- 方法
- appendChild
- cloneNode
- compareDocumentPosition
- contains
- getRootNode() [我来译!]
- getUserData()
- hasChildNodes
- insertBefore()

type

var type = node.nodeType;

返回一个整数，其代表的是节点类型。其所有可能的值请参考 节点类型常量。

常量

节点类型常量

常量	值	描述
Node.ELEMENT_NODE	1	一个 元素 节点，例如 <code><p></code> 和 <code><div></code> 。
Node.TEXT_NODE	3	Element 或者 Attr 中实际的 文字
Node.PROCESSING_INSTRUCTION_NODE	7	一个用于XML文档的 <code>ProcessingInstruction</code> ，例如 <code><?xml-stylesheet ... ?></code> 声明。
Node.COMMENT_NODE	8	一个 <code>Comment</code> 节点。
Node.DOCUMENT_NODE	9	一个 <code>Document</code> 节点。
Node.DOCUMENT_TYPE_NODE	10	描述文档类型的 <code>DocumentType</code> 节点。例如 <code><!DOCTYPE html></code> 就是用于 HTML5 的。
Node.DOCUMENT_FRAGMENT_NODE	11	一个 <code>DocumentFragment</code> 节点

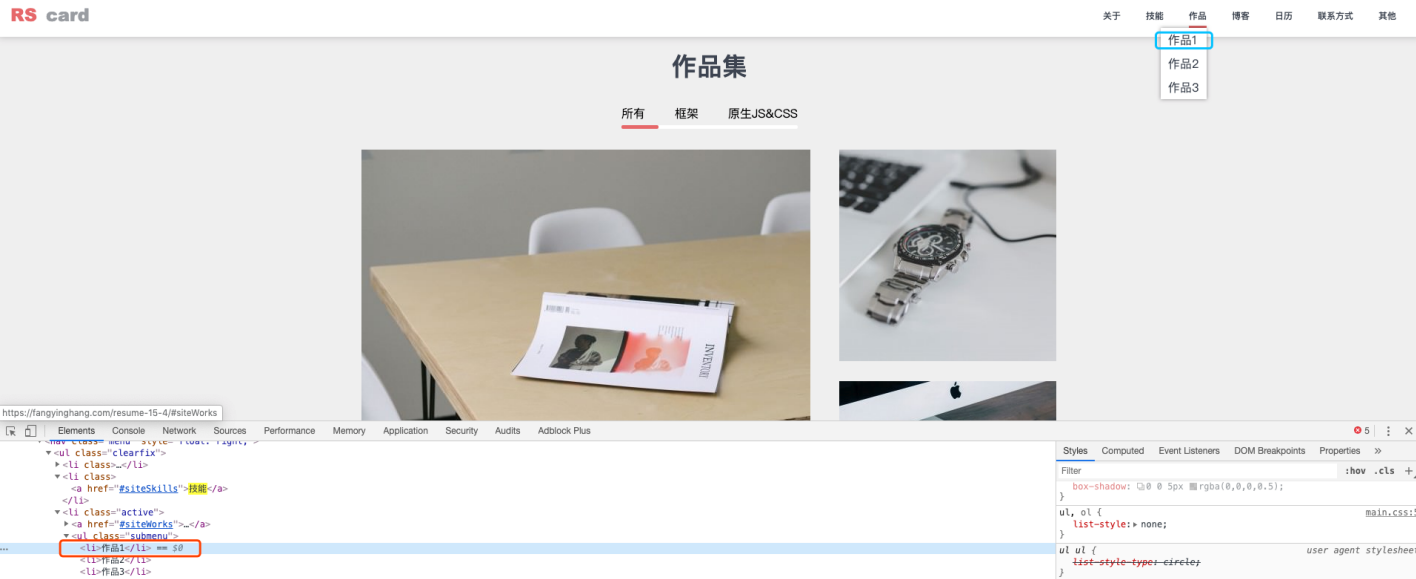
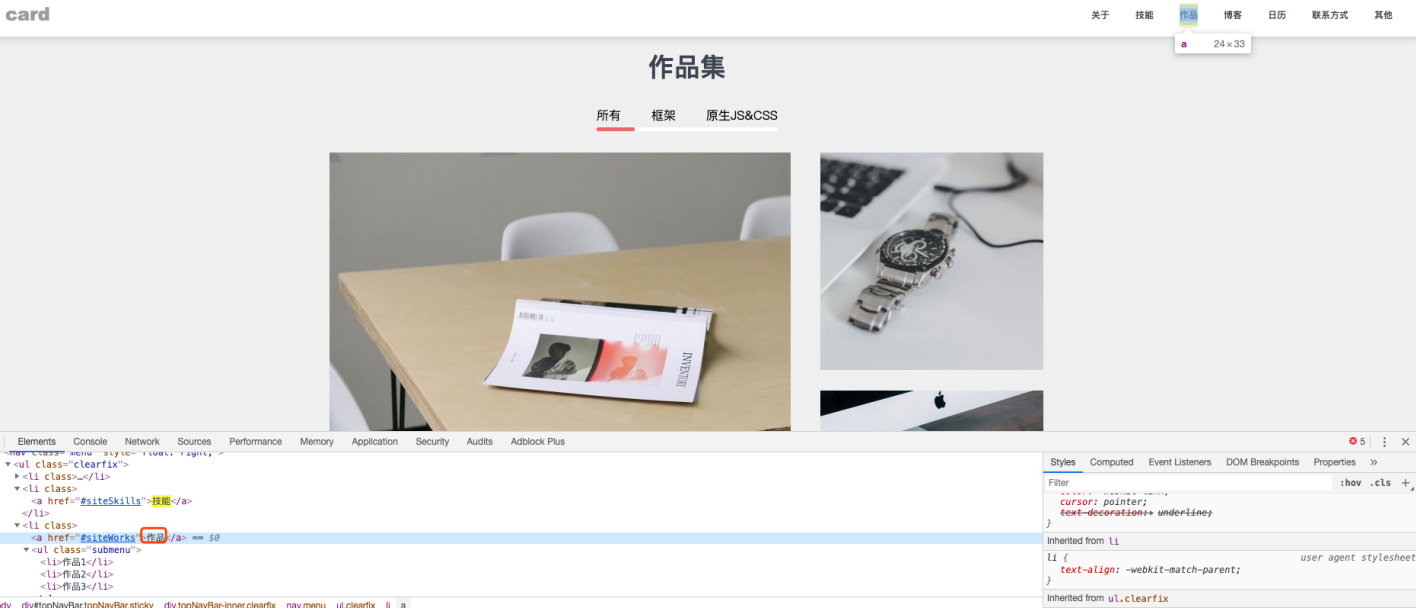
找弟弟的几个核心代码

这里的tagName这个api就是挺sb的，他会返回大写,故这里为大写的UL

```
8 | topNavBar.classList.remove('sticky')
7 | }
6 | }
5 | let aTags = document.getElementsByClassName('menuTigger')
4 | for(let i=0; i<aTags.length; i++){
3 |   aTags[i].mouseenter = function(x){
2 |     let a = x.currentTarget
1 |     let brother = a.nextSibling
256 |     while(brother.tagName !== 'UL'){
1 |       brother = brother.nextSibling
2 |     }
3 |     console.log(brother)
4 |   }
5 |   aTags[i].mouseleave = function(){
6 |     console.log('mouseleave')
7 |   }
8 | }
9 | </script>
10 | </body>
11 |
12 | </html>
```

这几行代码的意思是要a标签找到它的子标签里为ul的标签(a标签和ul标签之间隔着回车(回车算是文本!))的!
上面的menuTigger表示的是作品和博客的css那部分里面的标签所对应的Class的名字!

a标签和ul标签如下:

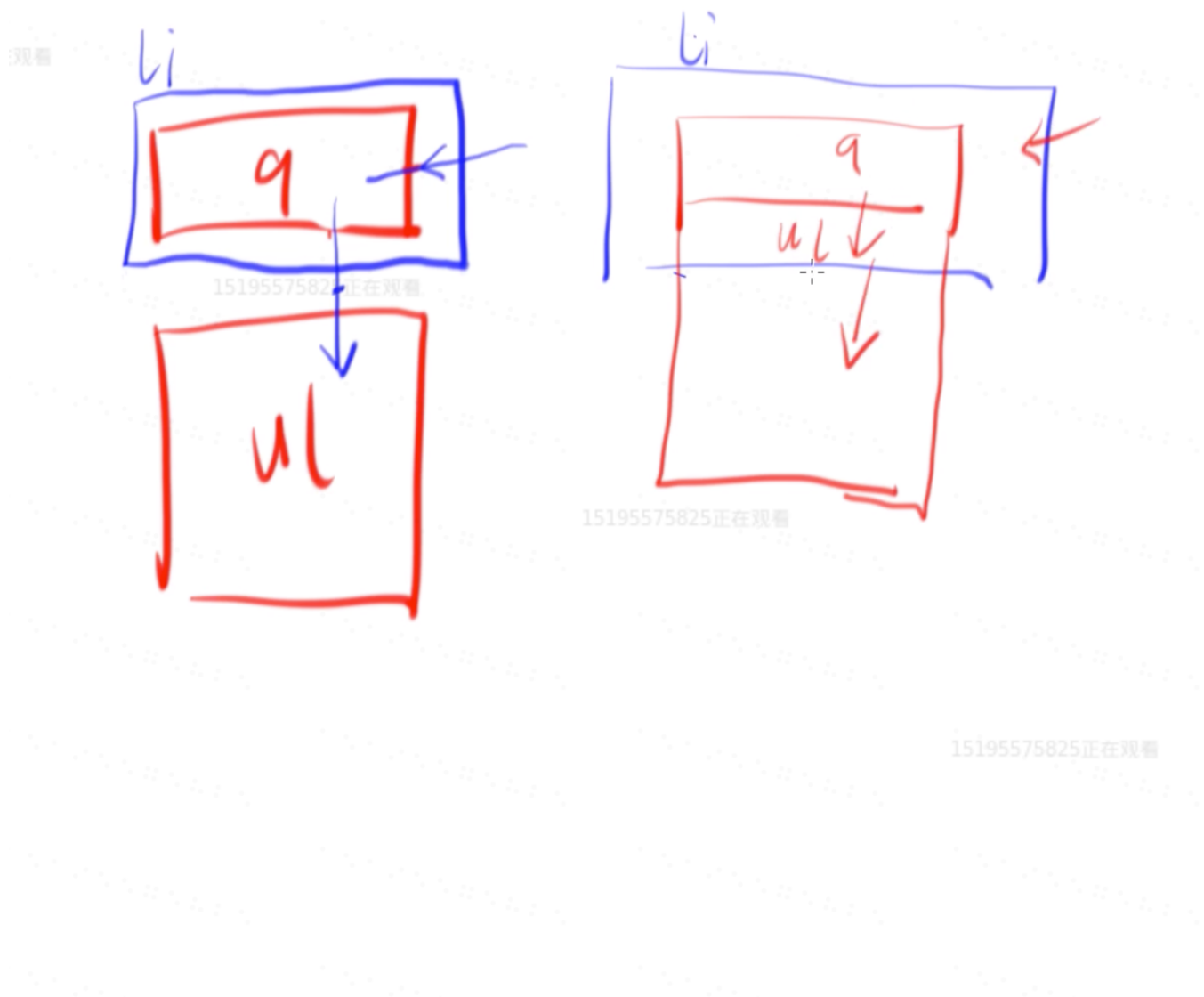


解决看不到作品的bug

这里为视频如何解决问题里面的！！

由于监听对象为a,故鼠标一离开，就看不到ul标签了(看左图)

如果把监听对象改为li，同时将a和ul靠近，就好；注意：鼠标在li里面还是外面不是通过面积来算的,而是通过你的这个元素在不在li里面来算的！



修改:

```
1519 </a>25正在观看
|
| <nav style="float: right;">
|   <ul class="clearfix">
|     <li>
|       <a href="#">关于</a>
|     </li>
|     <li>
|       <a href="#">技能</a>
|     </li>
|     <li>
|       <a class="menuTigger" href="#">作品</a>
|       <ul class="submenu">
|         <li>作品 1</li>
|         <li>作品 2</li>
|         <li>作品 3</li>
|       </ul>
|     </li>
|     <li>
|       <a class="menuTigger" href="#">博客</a>
|       <ul class="submenu">
|         <li>博客 1</li>
|         <li>博客 2</li>
|         <li>博客 3</li>
|       </ul>
|     </li>
|     <li>
|       <a href="#">日历</a>
|     </li>
|     <li>
|       <a href="#">联系方式</a>
|     </li>
|     <li>
```

修改为:

```

</a>
<nav style="float: right;">
  <ul class="clearfix">
    <li>
      <a href="#">关于</a>
    </li>
    <li>
      <a href="#">技能</a>
    </li>
    <li class="menuTigger" >
      <a href="#">作品</a>
      <ul class="submenu">
        <li>作品 1</li>
        <li>作品 2</li>
        <li>作品 3</li>
      </ul>
    </li>
    <li class="menuTigger">
      <a href="#">博客</a>
      <ul class="submenu">
        <li>博客 1</li>
        <li>博客 2</li>
        <li>博客 3</li>
      </ul>
    </li>
    <li>
      <a href="#">日历</a>
    </li>
    <li>
      <a href="#">联系方式</a>
    </li>
    <li>
      <a href="#">其他</a>
    </li>
  </ul>
</nav>

```

同时,js那边也要修改:

原来的:

```

8 window.onscroll = function(xxxxx){
7   if(window.pageY > 0){
6     topNavBar.classList.add('sticky')
5   }else{
4     topNavBar.classList.remove('sticky')
3   }
2 }
1 let aTags = document.getElementsByClassName('menuTigger')
52 for(let i=0; i<aTags.length; i++){
1   aTags[i].onmouseenter = function(x){
2     let a = x.currentTarget
3     let brother = a.nextSibling
4     while(brother.tagName !== 'UL'){
5       brother = brother.nextSibling
6     }
7     brother.classList.add('active')
8   }
9   aTags[i].onmouseleave = function(x){
10    let a = x.currentTarget
11    let brother = a.nextSibling
12    while(brother.tagName !== 'UL'){
13      brother = brother.nextSibling
14    }
15    brother.classList.remove('active')
16  }
17 }
18 </script>
19 </body>
20 </html>

```

修改为:

通过 `li.getElementsByTagName('ul')[0]` 来获取ul标签, 这里数组虽然只有一个元素,但是还是要写[0]

```
3 }
2 }
1 let liTags = document.getElementsByClassName('menuTigger')
52 for(let i=0; i<liTags.length; i++){
1   liTags[i].onmouseenter = function(x){
2     let li = x.currentTarget
3     let brother = li.getElementsByTagName('ul')[0]
4     brother.classList.add('active')
5   }
6   liTags[i].onmouseleave = function(x){
7     let li = x.currentTarget
8     let brother = li.getElementsByTagName('ul')[0]
9     brother.classList.remove('active')
10  }
11 }
12 </script>
13 </body>
14
```

15195575825正在观看

15