

添加文本的两种方式

document是html的爸爸

referer

close

查找querySelector和querySelectorAll

element里面的innerText和innerHTML区别

添加文本的两种方式



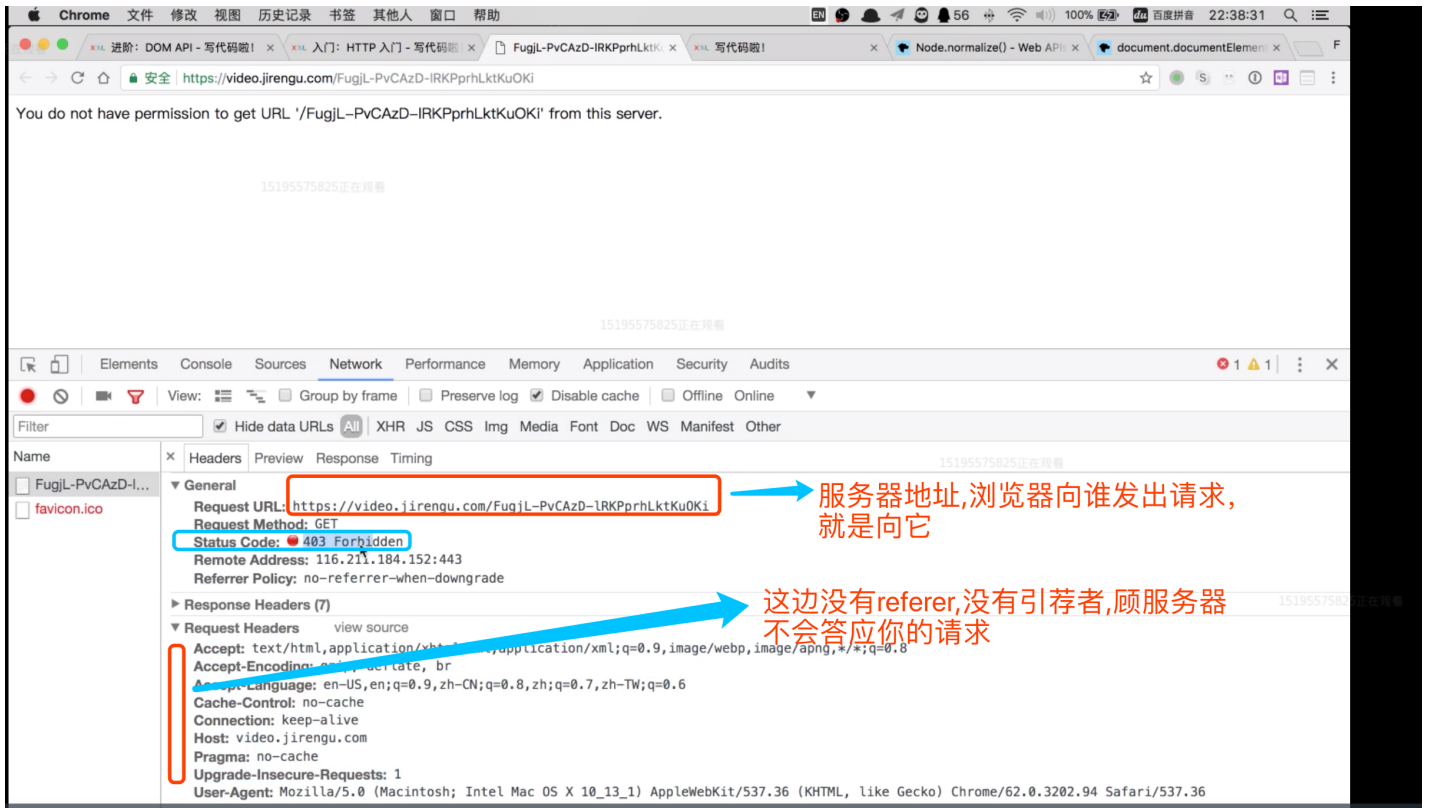
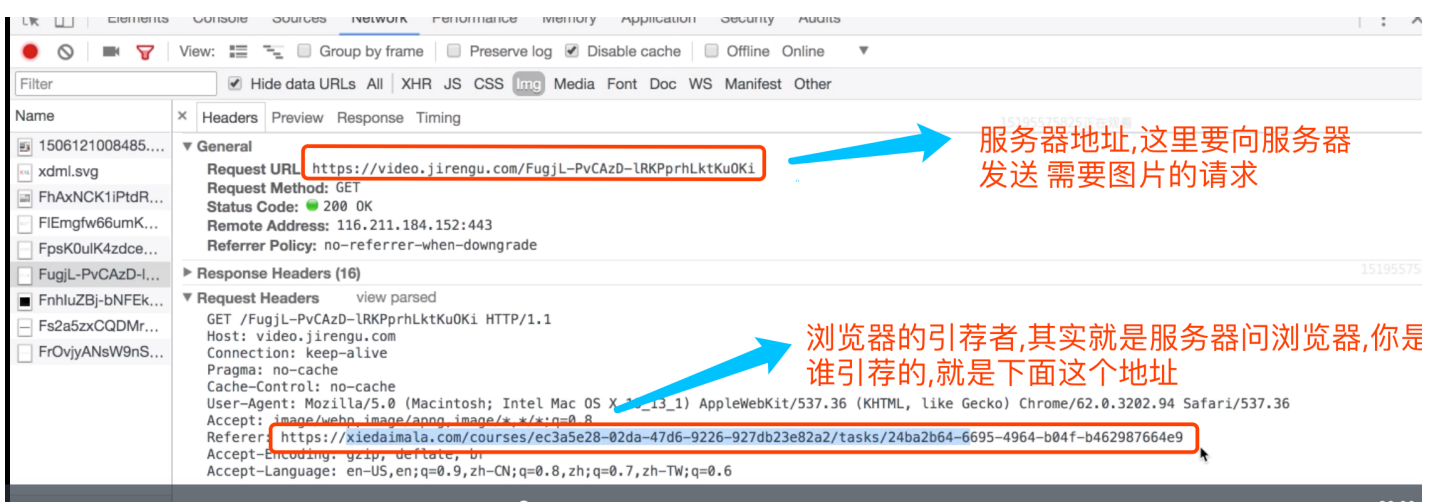
两种都可以用,如果哪天在某种场景下使用其中一种遇到了bug,那么请换成另外一种!

document是html的爸爸

document是html的爸爸;但是Html还是根元素!

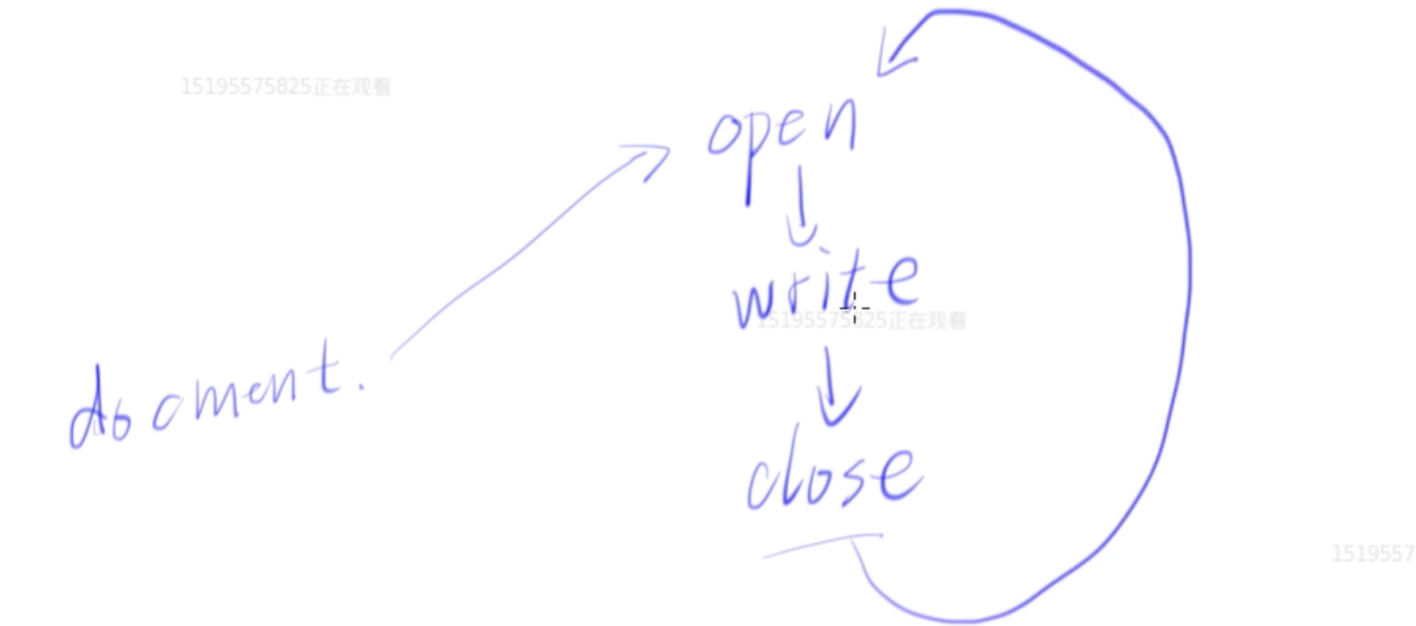


referer



close

页面加载完就会自动close!!! 如果再取write就会把之前的内容覆盖掉!



看下面这个例子:

HTML

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>JS Bin</title>
</head>
<body>
 hello
 <script>
 document.write(1)
 document.write(2)
 setTimeout(()=>{
 document.write(3)
 },1000)
 </script>
</body>
</html>

CSS

Output 480px
hello 12

1秒之后:(就是因为自动close了之后把之前的内容都覆盖了)

HTML

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>JS Bin</title>
</head>
<body>
 hello
 <script>
 document.write(1)
 document.write(2)
 setTimeout(()=>{
 document.write(3)
 },1000)
 </script>
</body>
</html>

CSS

Output 480px
3

查找querySelector和querySelectorAll

querySelector返回一个元素;

querySelectorAll返回一个数组(即使只有一个元素也是以数组的形式出现!),需要注意的是:这里返回的是伪数

组!!!

```
>> var div = document.querySelector('ul li')
← undefined

>> div
← ▶ <li class="btn-color first skin-default selected" data-title="蓝色海洋" data-type="blue"> ⚙️

⚠️ ▶ 鉴于已开启内容拦截功能，而这个请求来自跟踪器，已拦截“<URL>”对 Cookie 或存储空间的访问请求。 2

>> var div = document.querySelectorAll('ul li')
← undefined

>> div
← ▼ NodeList(334)
  ▶ [0...99]
  ▶ [100...199]
  ▶ [200...299]
  ▶ [300...333]
  length: 334
  ▶ <prototype>: NodeListPrototype { item: item(), keys: keys(), values: values(), ... }

>>
```

element里面的innerText和innerHTML区别

The screenshot shows a web development environment with three panels:

- HTML:** Contains the following code:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
</head>
<body>
  <input type="text" id="input1"> <button id=x>写入页面
</button>
  <div id="output"></div>
</body>
</html>
```
- JavaScript:** Contains the following code:

```
x.onclick = function(){
  output.innerText = input1.value
}
```
- Output:** Shows the result of the JavaScript execution: `12345657`. A blue arrow points from the `output.innerText` property in the JavaScript panel to the output result.

使用innerText的时候:input1里面写什么,output里面就输出什么!

而使用innerHTML的时候:

input1里面写什么,output里面不一定就输出什么!(看下面两个例子!)(使用innerHTML是十分危险的!因为如果写script的话可能会十分危险!)

HTML

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>JS Bin</title>
</head>
<body>
 <input type="text" id="input1"> <button id=x>写入页面
</button>
<div id="output"></div>
</body>
</html>

JavaScript

x.onclick = function(){
 output.innerHTML = input1.value
}

Output

<i>1231241</i>
1231241

写入页面

Run w

HTML

<!DOCTYPE html>
<html>
<head>
 <meta charset="utf-8">
 <title>JS Bin</title>
</head>
<body>
 <input type="text" id="input1"> <button id=x>写入页面
</button>
 <div id="output"></div>
</body>
</html>

JavaScript

x.onclick = function(){
 output.innerHTML = input1.value
}

Output

1231241
1231241

写入页面

Run with JS