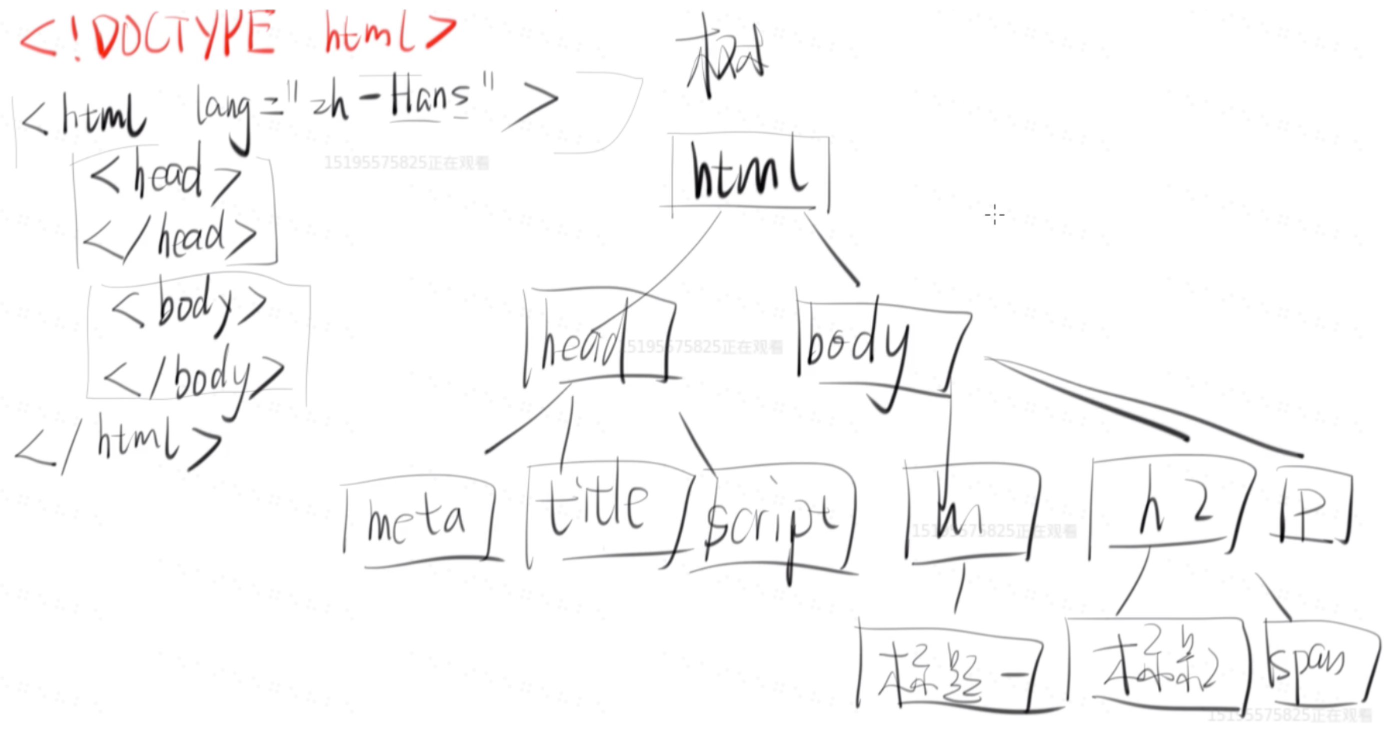
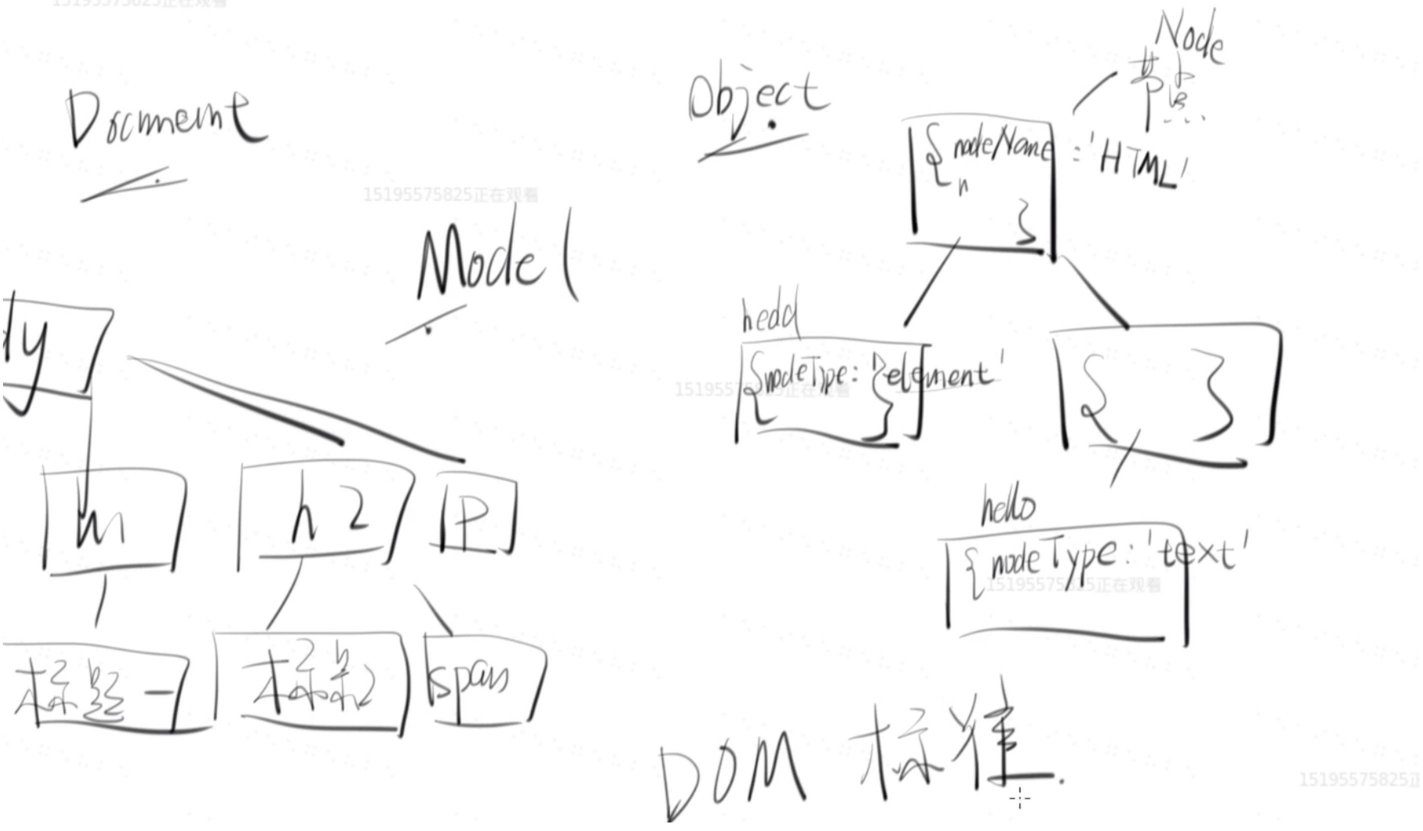
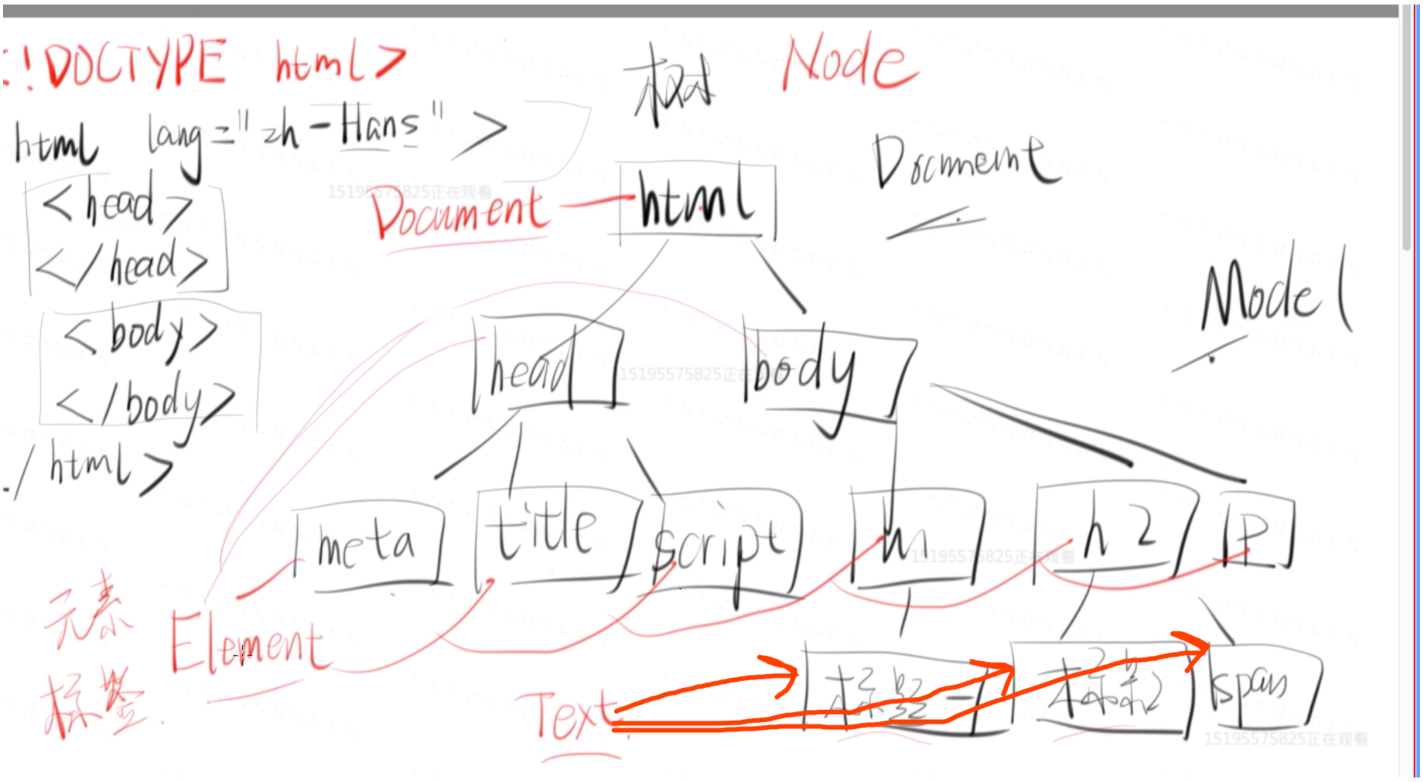
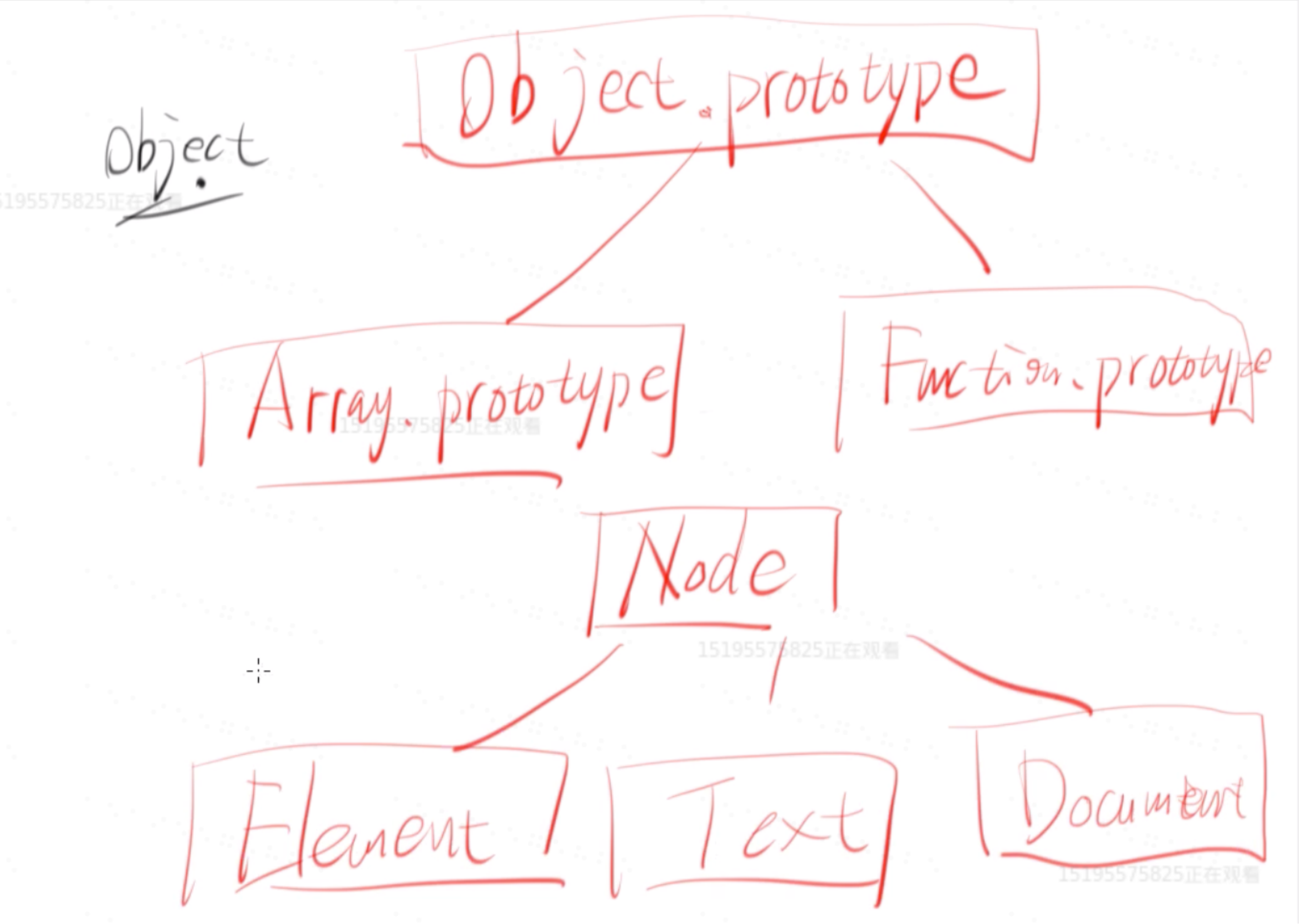
http://luopq.com/2015/11/30/javascript-dom/ Javascript操作DOM常用API总结
https://github.com/wojiaofengzhongzhuifeng/study/blob/master/blog/B8-DOM.md 优秀笔记
https://xiedaimala.com/tasks/60d4d732-19f5-442b-9993-4d53ff94657c 总体知识点概要

概要



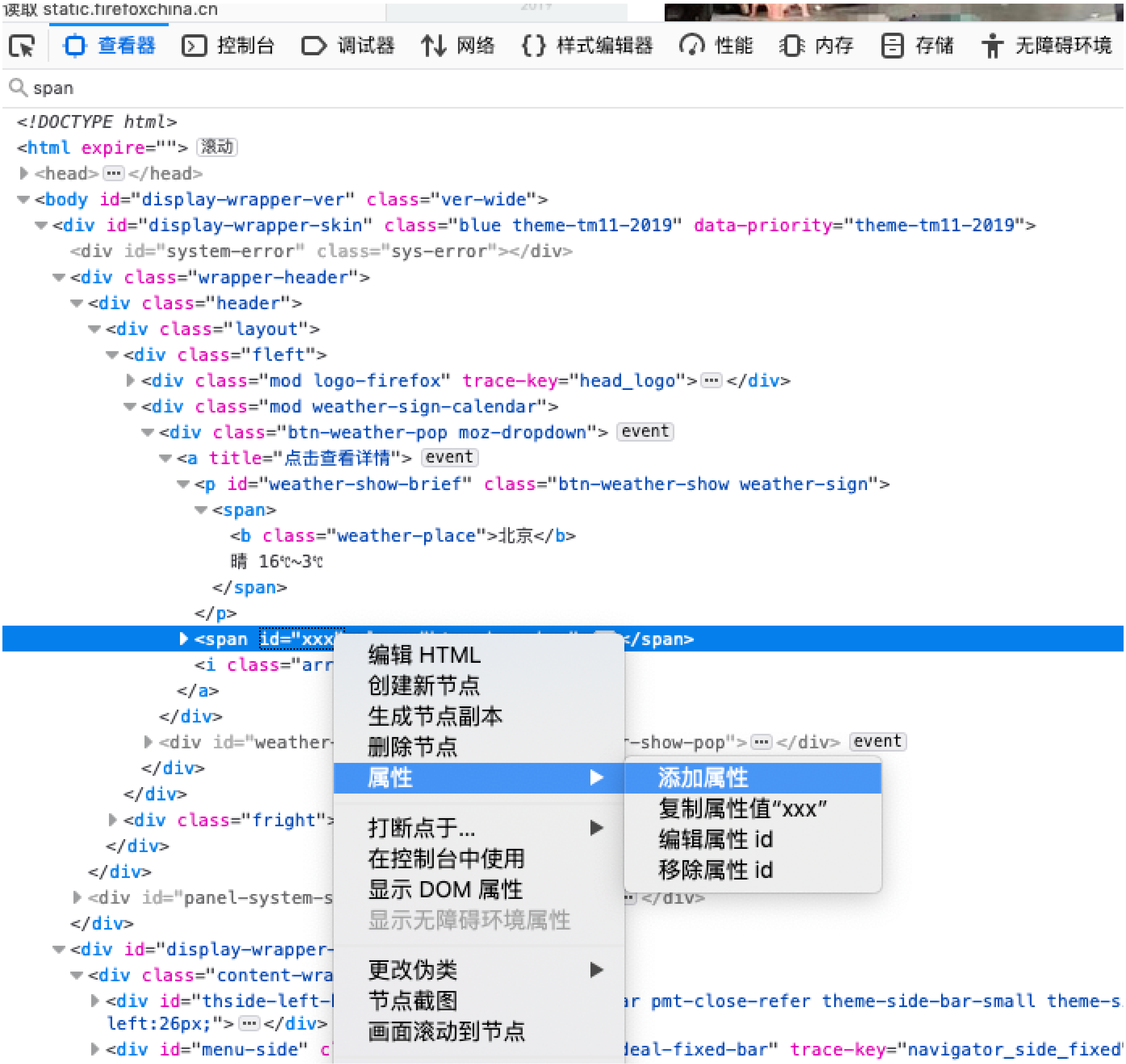

Document
Model
y
h1
h2
p
标题一
标题2
span
Object
Node
节点
{nodeName: 'HTML'
}
head
{nodeType: 'element'
}
{ }
hello
{nodeType: 'text'
DOM 标准.


<!DOCTYPE html>
html lang="zh-Hans" >
<head>
</head>
<body>
</body>
</html>
树
Node
Document
Model
Document
html
head
body
meta
title
script
h1
h2
p
标题一
标题2
span
元素
标签
Element
Text

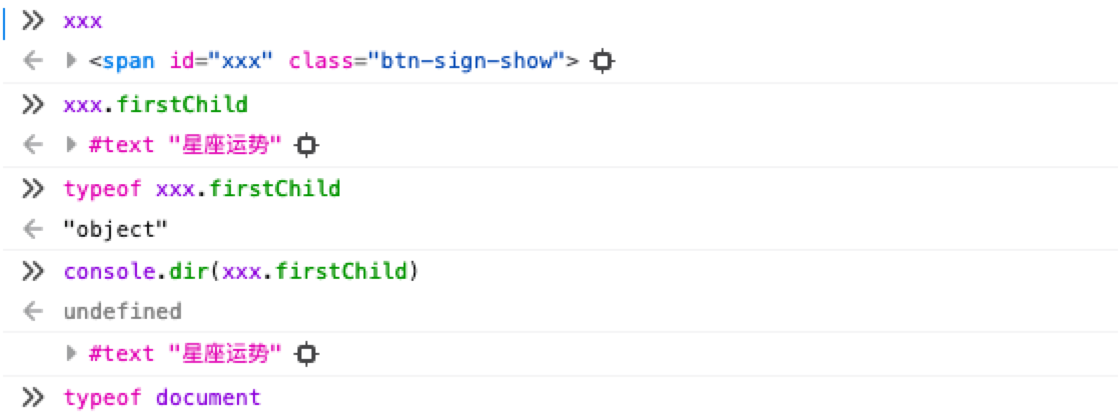

Object.
Object.prototype
Array.prototype
Function.prototype
Node
Element
Text
Document

一个例子

添加属性 `id=xxx`：



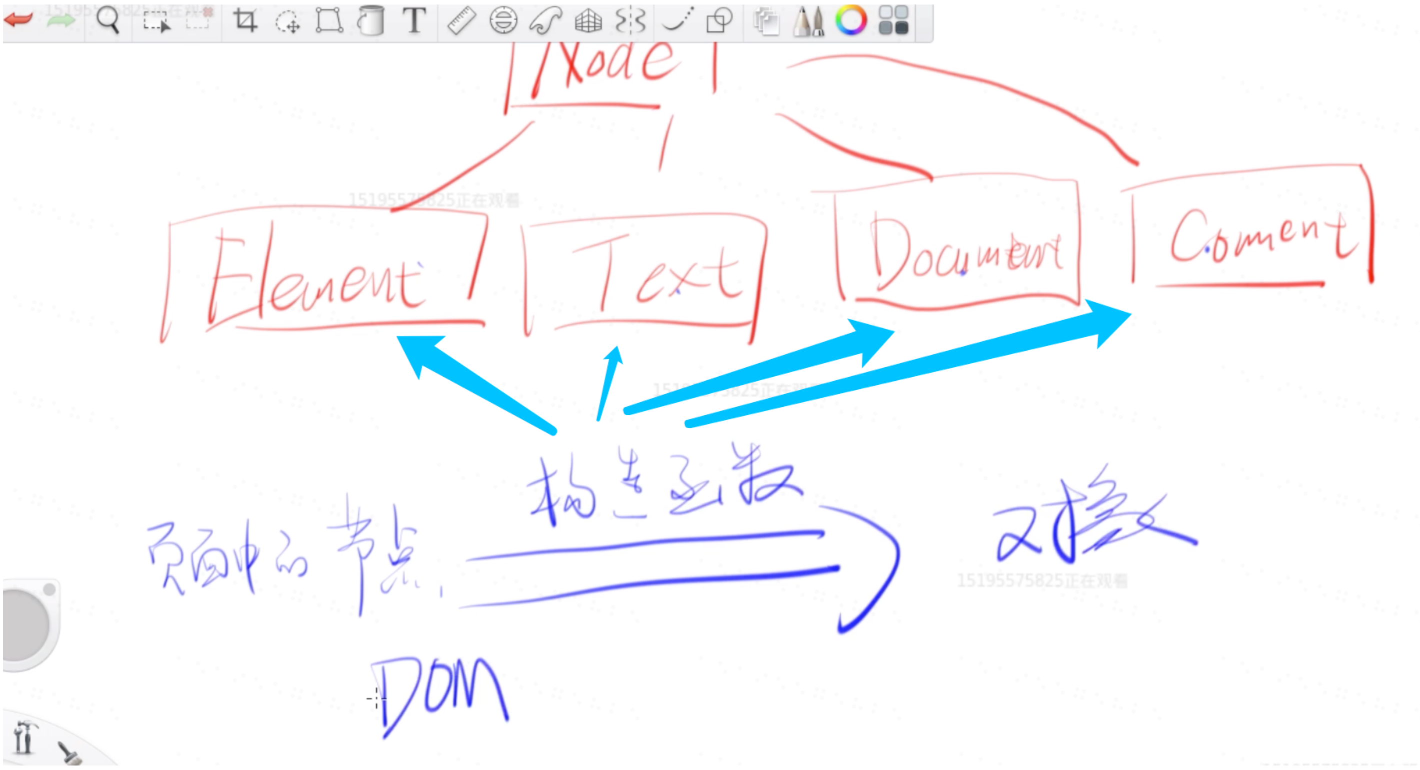
```
<!DOCTYPE html>
<html expire="">
▶<head>⋯</head>
▼<body id="display-wrapper-ver" class="ver-wide">
  ▼<div id="display-wrapper-skin" class="blue theme-tm11-2019" data-priority="theme-tm11-2019">
      <div id="system-error" class="sys-error"></div>
    ▼<div class="wrapper-header">
      ▼<div class="header">
        ▼<div class="layout">
          ▼<div class="fleft">
            ▶<div class="mod logo-firefox" trace-key="head_logo">⋯</div>
            ▼<div class="mod weather-sign-calendar">
              ▼<div class="btn-weather-pop moz-dropdown">
                ▼<a title="点击查看详情">
                  ▼<p id="weather-show-brief" class="btn-weather-show weather-sign">
                    ▼<span>
                        <b class="weather-place">北京</b>
                        晴 16℃~3℃
                      </span>
                    </p>
                  ▶<span id="xxx"                          </span>
                    <i class="arr
                  </a>
                </div>
              ▶<div id="weather-                   -show-pop">⋯</div>
              </div>
            </div>
          ▶<div class="fright">
            </div>
          </div>
        ▶<div id="panel-system-s                    ⋯</div>
        </div>
      ▼<div id="display-wrapper-
        ▼<div class="content-wra
          ▶<div id="thside-left-l                   ar pmt-close-refer theme-side-bar-small theme-s
            left:26px;">⋯</div>
          ▶<div id="menu-side" c                    deal-fixed-bar" trace-key="navigator_side_fixed'
```



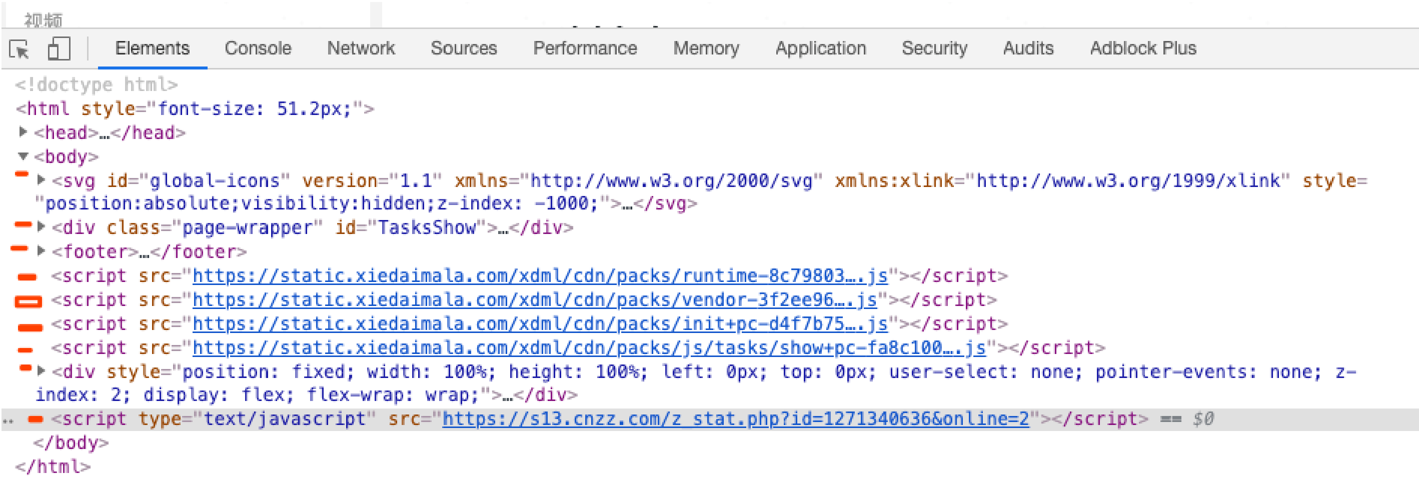
```
>> xxx
← ▶<span id="xxx" class="btn-sign-show">
>> xxx.firstChild
← ▶#text "星座运势"
>> typeof xxx.firstChild
← "object"
>> console.dir(xxx.firstChild)
← undefined
  ▶#text "星座运势"
>> typeof document
```

想要操作节点,该怎么做?

将每个节点变成一个对象,你想操作那个对象,就调用对应的api;

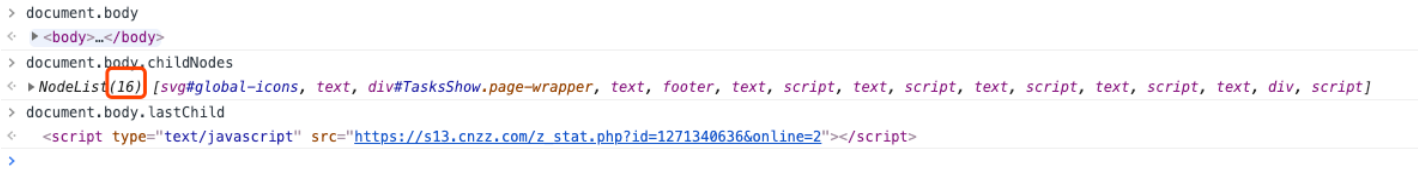


关于节点的一个例子(childNodes(子节点)和children(子标签))



```
<!doctype html>
<html style="font-size: 51.2px;">
 ▶<head>…</head>
 ▼<body>
   ▶<svg id="global-icons" version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" style=
    "position:absolute;visibility:hidden;z-index: -1000;">…</svg>
   ▶<div class="page-wrapper" id="TasksShow">…</div>
   ▶<footer>…</footer>
    <script src="https://static.xiedaimala.com/xdml/cdn/packs/runtime-8c79803….js"></script>
    <script src="https://static.xiedaimala.com/xdml/cdn/packs/vendor-3f2ee96….js"></script>
    <script src="https://static.xiedaimala.com/xdml/cdn/packs/init+pc-d4f7b75….js"></script>
    <script src="https://static.xiedaimala.com/xdml/cdn/packs/js/tasks/show+pc-fa8c100….js"></script>
   ▶<div style="position: fixed; width: 100%; height: 100%; left: 0px; top: 0px; user-select: none; pointer-events: none; z-
   index: 2; display: flex; flex-wrap: wrap;">…</div>
..  <script type="text/javascript" src="https://s13.cnzz.com/z_stat.php?id=1271340636&online=2"></script> == $0
  </body>
</html>
```

这里明明只有9个为什么会显示出16个呢？？

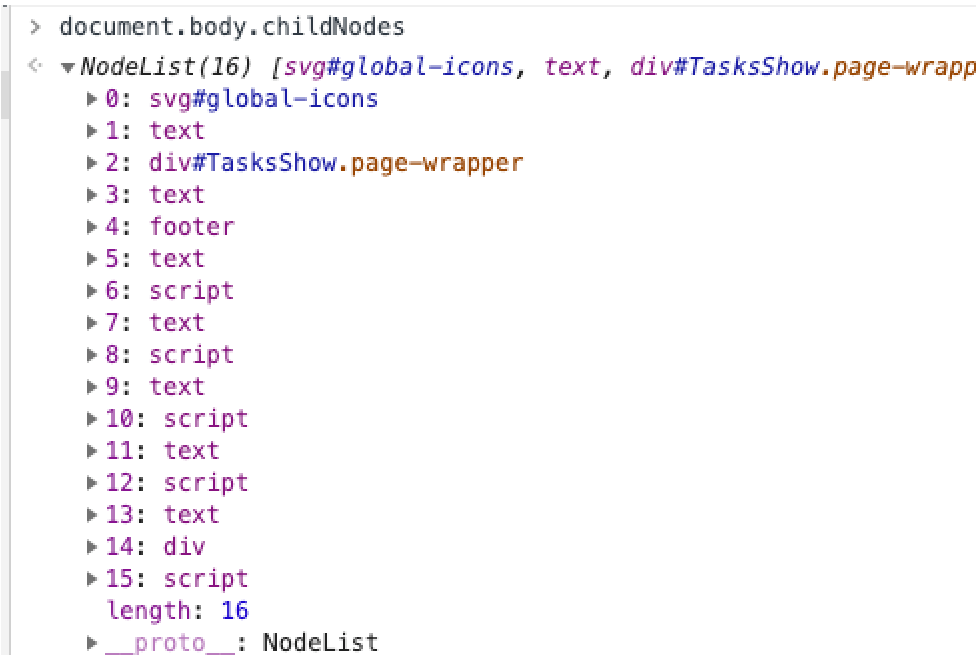


```
> document.body
< ▶<body>…</body>
> document.body.childNodes
< ▶NodeList(16) [svg#global-icons, text, div#TasksShow.page-wrapper, text, footer, text, script, text, script, text, script, text, script, text, div, script]
> document.body.lastChild
<   <script type="text/javascript" src="https://s13.cnzz.com/z_stat.php?id=1271340636&online=2"></script>
>
```

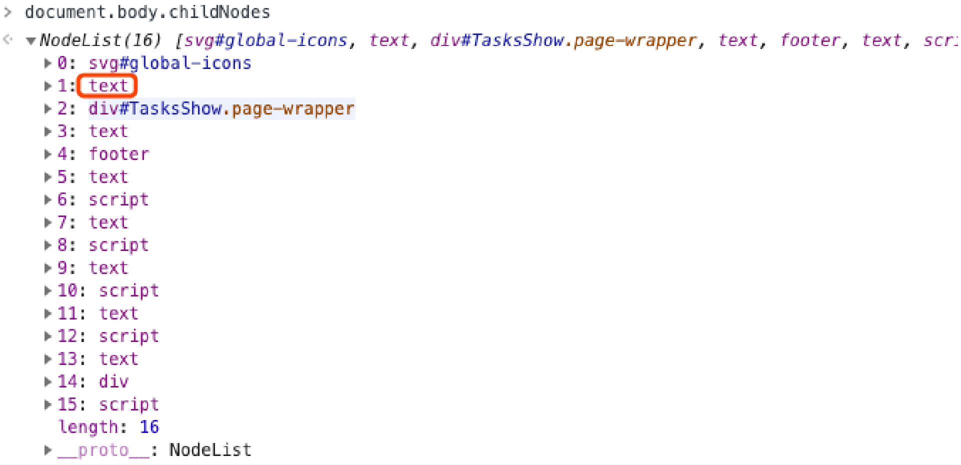
看这里:

如下所示:这里以其中一个地方为例;

0和2之间的text其实是 `svg` 标签和 `div` 标签之间的回车!

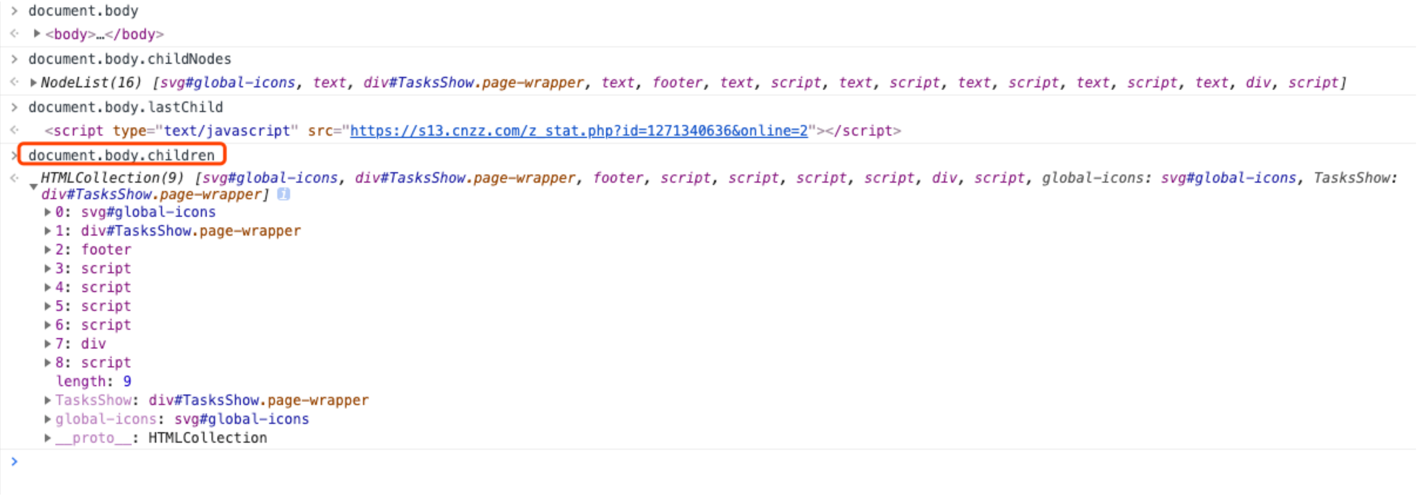


```
> document.body.childNodes
< ▼NodeList(16) [svg#global-icons, text, div#TasksShow.page-wrapp
    ▶0: svg#global-icons
    ▶1: text
    ▶2: div#TasksShow.page-wrapper
    ▶3: text
    ▶4: footer
    ▶5: text
    ▶6: script
    ▶7: text
    ▶8: script
    ▶9: text
    ▶10: script
    ▶11: text
    ▶12: script
    ▶13: text
    ▶14: div
    ▶15: script
     length: 16
    ▶__proto__: NodeList
```

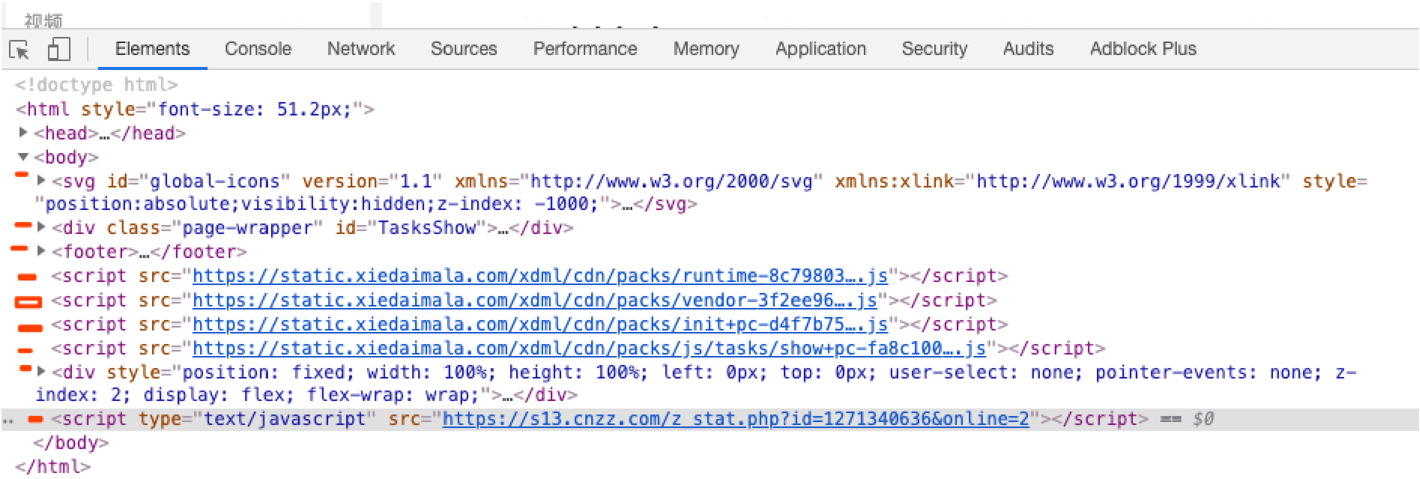


```
> document.body.childNodes
< ▼NodeList(16) [svg#global-icons, text, div#TasksShow.page-wrapper, text, footer, text, scr
    ▶0: svg#global-icons
    ▶1: text
    ▶2: div#TasksShow.page-wrapper
    ▶3: text
    ▶4: footer
    ▶5: text
    ▶6: script
    ▶7: text
    ▶8: script
    ▶9: text
    ▶10: script
    ▶11: text
    ▶12: script
    ▶13: text
    ▶14: div
    ▶15: script
     length: 16
    ▶__proto__: NodeList
```

而children是不会获取回车的!

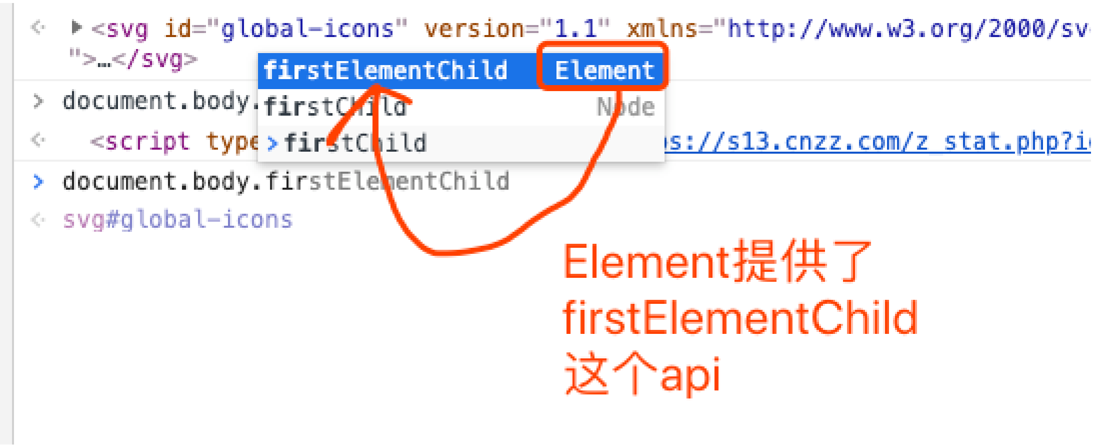


```
> document.body
< ▶<body>…</body>
> document.body.childNodes
< ▶NodeList(16) [svg#global-icons, text, div#TasksShow.page-wrapper, text, footer, text, script, text, script, text, script, text, script, text, div, script]
> document.body.lastChild
<   <script type="text/javascript" src="https://s13.cnzz.com/z_stat.php?id=1271340636&online=2"></script>
> document.body.children
< ▼HTMLCollection(9) [svg#global-icons, div#TasksShow.page-wrapper, footer, script, script, script, script, div, script, global-icons: svg#global-icons, TasksShow:
   div#TasksShow.page-wrapper] ℹ
   ▶0: svg#global-icons
   ▶1: div#TasksShow.page-wrapper
   ▶2: footer
   ▶3: script
   ▶4: script
   ▶5: script
   ▶6: script
   ▶7: div
   ▶8: script
    length: 9
   ▶TasksShow: div#TasksShow.page-wrapper
   ▶global-icons: svg#global-icons
   ▶__proto__: HTMLCollection
>
```



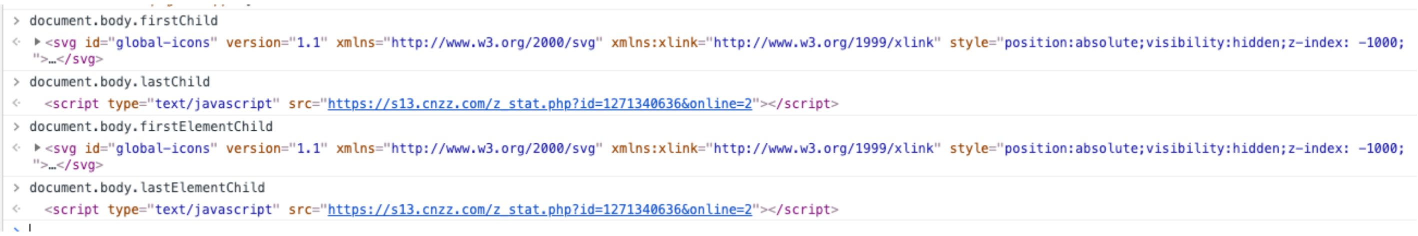
```
<!doctype html>
<html style="font-size: 51.2px;">
▶<head>…</head>
▼<body>
  ▶<svg id="global-icons" version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" style=
  "position:absolute;visibility:hidden;z-index: -1000;">…</svg>
  ▶<div class="page-wrapper" id="TasksShow">…</div>
  ▶<footer>…</footer>
   <script src="https://static.xiedaimala.com/xdml/cdn/packs/runtime-8c79803….js"></script>
   <script src="https://static.xiedaimala.com/xdml/cdn/packs/vendor-3f2ee96….js"></script>
   <script src="https://static.xiedaimala.com/xdml/cdn/packs/init+pc-d4f7b75….js"></script>
   <script src="https://static.xiedaimala.com/xdml/cdn/packs/js/tasks/show+pc-fa8c100….js"></script>
  ▶<div style="position: fixed; width: 100%; height: 100%; left: 0px; top: 0px; user-select: none; pointer-events: none; z-
  index: 2; display: flex; flex-wrap: wrap;">…</div>
…  <script type="text/javascript" src="https://s13.cnzz.com/z_stat.php?id=1271340636&online=2"></script> == $0
  </body>
</html>
```

firstChild,lastChild,firstElementChild



```
<  ▶<svg id="global-icons" version="1.1" xmlns="http://www.w3.org/2000/sv
   ">…</svg>
>  document.body.firstChild
<    <script type
>  document.body.firstElementChild
<  svg#global-icons
```

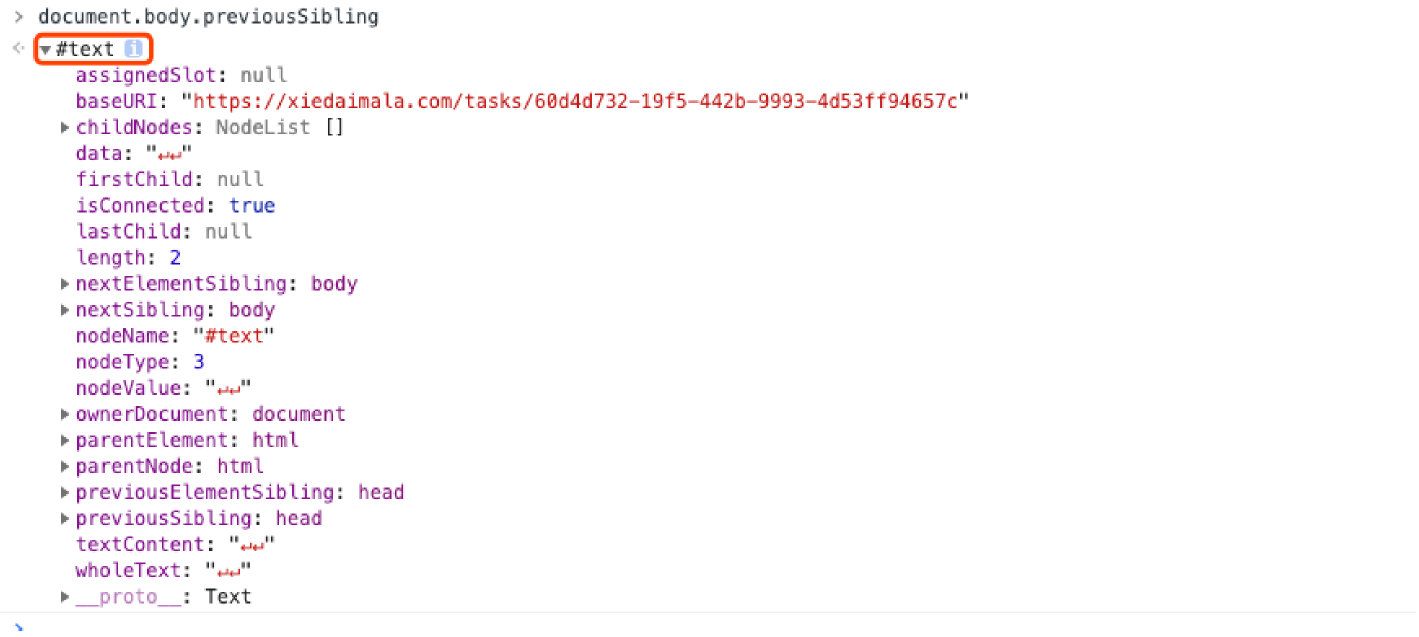
Element提供了
firstElementChild
这个api



```
> document.body.firstChild
<  ▶<svg id="global-icons" version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" style="position:absolute;visibility:hidden;z-index: -1000;
   ">…</svg>
> document.body.lastChild
<    <script type="text/javascript" src="https://s13.cnzz.com/z_stat.php?id=1271340636&online=2"></script>
> document.body.firstElementChild
<  ▶<svg id="global-icons" version="1.1" xmlns="http://www.w3.org/2000/svg" xmlns:xlink="http://www.w3.org/1999/xlink" style="position:absolute;visibility:hidden;z-index: -1000;
   ">…</svg>
> document.body.lastElementChild
<    <script type="text/javascript" src="https://s13.cnzz.com/z_stat.php?id=1271340636&online=2"></script>
```

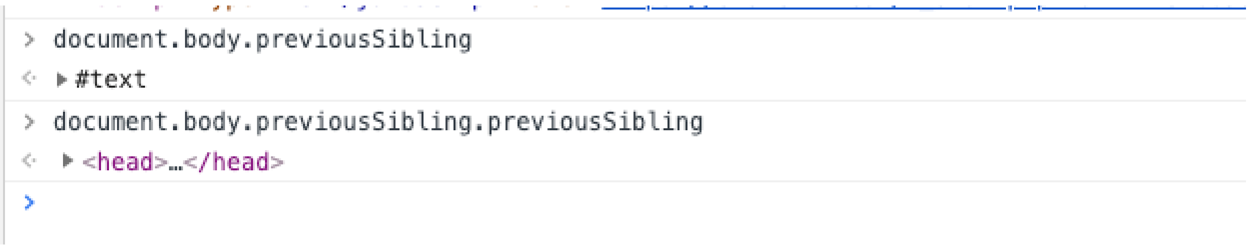
previousSibling

按道理来说这里body的哥哥应该是head,这里为什么是text呢?



```
> document.body.previousSibling
< ▾#text
    assignedSlot: null
    baseURI: "https://xiedaimala.com/tasks/60d4d732-19f5-442b-9993-4d53ff94657c"
  ▸ childNodes: NodeList []
    data: "↵↵"
    firstChild: null
    isConnected: true
    lastChild: null
    length: 2
  ▸ nextElementSibling: body
  ▸ nextSibling: body
    nodeName: "#text"
    nodeType: 3
    nodeValue: "↵↵"
  ▸ ownerDocument: document
  ▸ parentElement: html
  ▸ parentNode: html
  ▸ previousElementSibling: head
  ▸ previousSibling: head
    textContent: "↵↵"
    wholeText: "↵↵"
  ▸ __proto__: Text
>
```

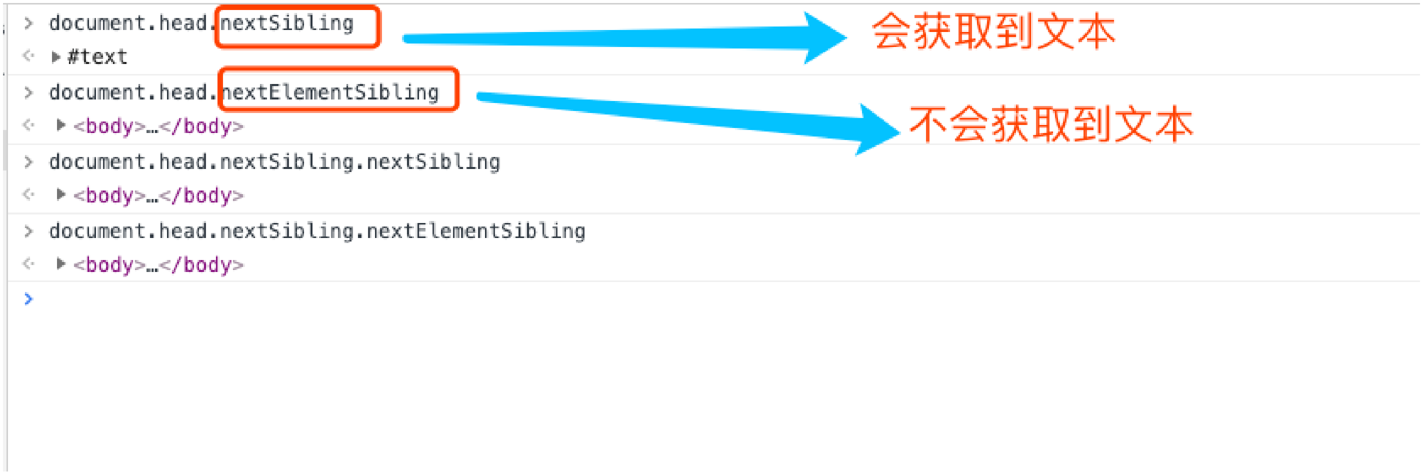
因为head和body之间有回车啊!!!只要再加一个previousSibling即可!



```
> document.body.previousSibling
< ▸#text
> document.body.previousSibling.previousSibling
< ▸<head>…</head>
>
```

nextElementSibling和nextSibling

nextSibling本来是为XML服务的,不是为html服务的;后来加了nextElementSibling



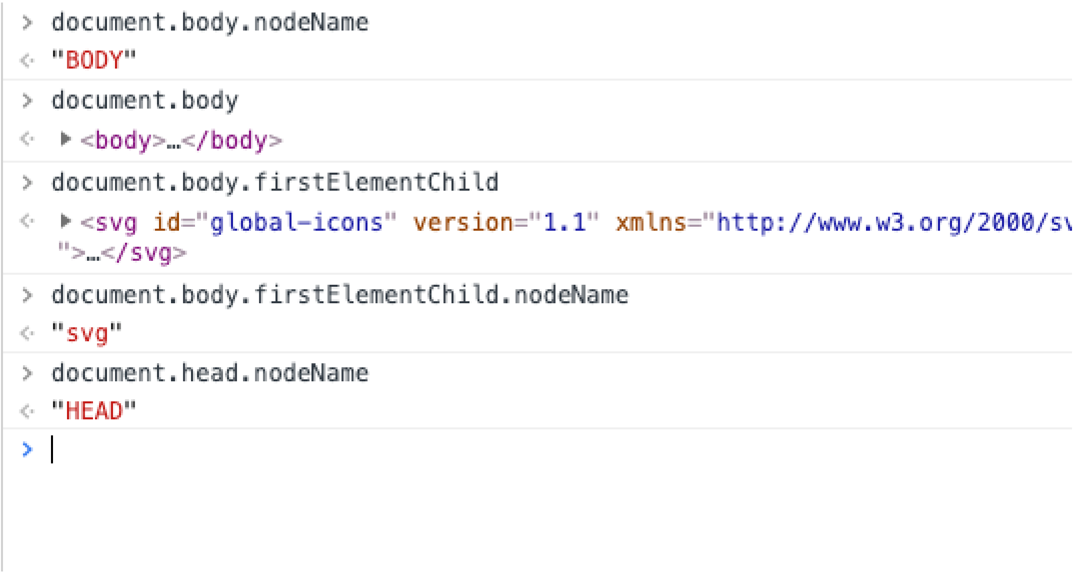
```
> document.head.nextSibling
< ▸#text
> document.head.nextElementSibling
< ▸<body>…</body>
> document.head.nextSibling.nextSibling
< ▸<body>…</body>
> document.head.nextSibling.nextElementSibling
< ▸<body>…</body>
>
```

会获取到文本

不会获取到文本

nodeName

注意点:只有svg是小写,其他都大写!!

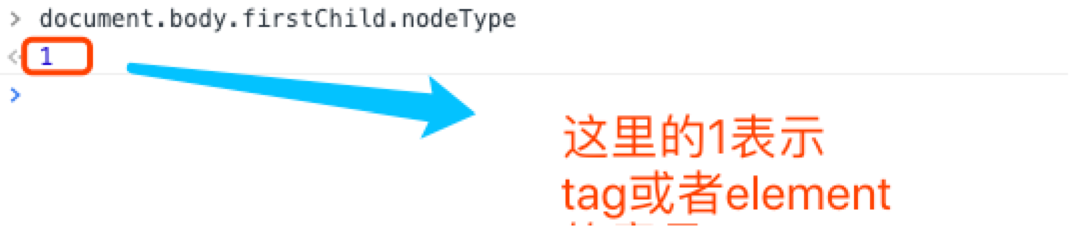


```
> document.body.nodeName
< "BODY"
> document.body
< ▸<body>…</body>
> document.body.firstElementChild
< ▸<svg id="global-icons" version="1.1" xmlns="http://www.w3.org/2000/sv
  ">…</svg>
> document.body.firstElementChild.nodeName
< "svg"
> document.head.nodeName
< "HEAD"
> |
```

nodeType

这里的1表示tag(element的意思),所以说js很奇葩!

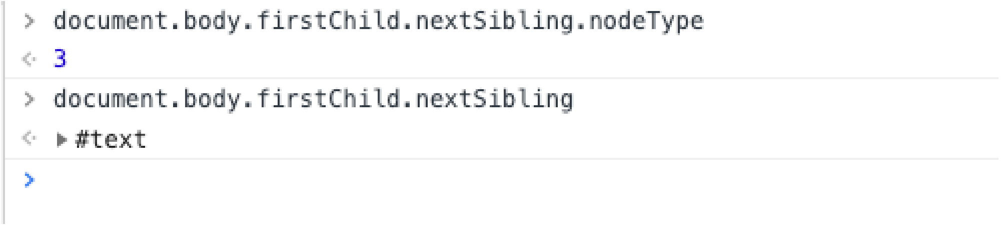
其实这里是考虑内存的原因!



```
> document.body.firstChild.nodeType
< 1
>
```

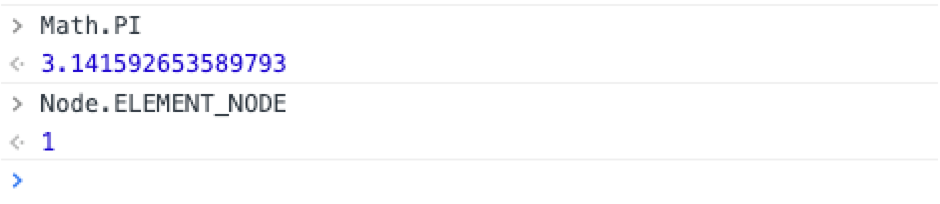
这里的1表示
tag或者element

3表示文本标签!



```
> document.body.firstChild.nextSibling.nodeType
< 3
> document.body.firstChild.nextSibling
< ▸#text
>
```

常量

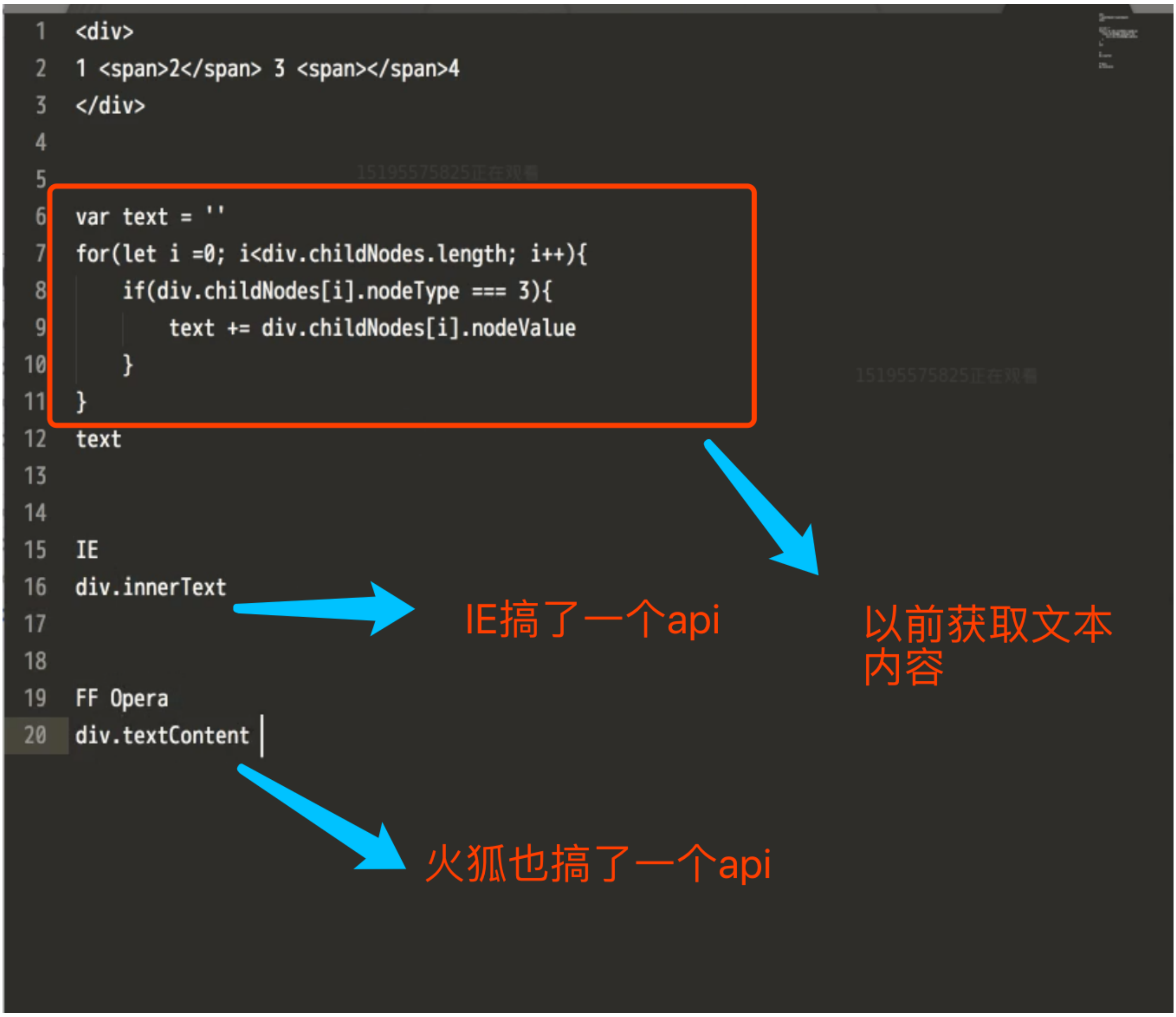


```
> Math.PI
< 3.141592653589793
> Node.ELEMENT_NODE
< 1
>
```

https://developer.mozilla.org/zh-CN/docs/Web/API/Node/nodeType

其他

获取文本内容的api



```
<div>
1 <span>2</span> 3 <span></span>4
</div>


var text = ''
for(let i =0; i<div.childNodes.length; i++){
    if(div.childNodes[i].nodeType === 3){
        text += div.childNodes[i].nodeValue
    }
}
text


IE
div.innerText


FF Opera
div.textContent
```

IE搞了一个api

以前获取文本内容

火狐也搞了一个api

两者之间的区别:

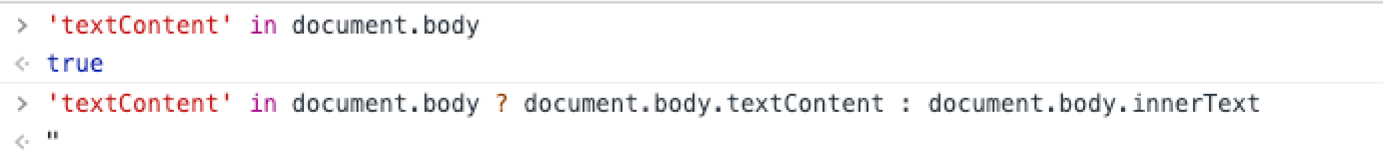
https://developer.mozilla.org/zh-CN/docs/Web/API/Node/textContent

与**innerText**的区别

Internet Explorer 引入了 `node.innerText`。意图类似，但有以下区别：

- `textContent` 会获取所有元素的内容，包括 `<script>` 和 `<style>` 元素，然而 **innerText** 不会。
- `innerText` 受 CSS 样式的影响，并且不会返回隐藏元素的文本，而textContent会。
- 由于 `innerText` 受 CSS 样式的影响，它会触发重排（reflow），但`textContent` 不会。
- 与 `textContent` 不同的是, 在 Internet Explorer (对于小于等于 IE11 的版本) 中对 innerText 进行修改， 不仅会移除当前元素的子节点，而且还会永久性地破坏所有后代文本节点（所以不可能再次将节点再次插入到任何其他元素或同一元素中）。

有 `textContent` 就用它，没有就用innerText!



```
> 'textContent' in document.body
< true
> 'textContent' in document.body ? document.body.textContent : document.body.innerText
< "
```