

队列(重要)

one函数

执行顺序

<https://github.com/wojiaofengzhongzhufeng/study/blob/master/blog/B8-DOM.md> 优秀笔记

W3C Working Draft

← → ↺ 🛡️ 安全 <https://www.w3.org/TR/2000/WD-DOM-Level-1-20000929/level-one-core.html> ☆ 🌐 📧 📄 📁 📌 📖 📑 📔 📕 📖 📑 📔 📕

[previous](#) [next](#) [contents](#) [objects](#) [index](#)

29 September, 2000

1. Document Object Model Core

Editors

Mike Champion, ArborText (from November 20, 1997)
Steve Byrne, JavaSoft (until November 19, 1997)
Gavin Nicol, Inso EPS
Lauren Wood, SoftQuad, Inc.

Table of contents

- [1.1. Overview of the DOM Core Interfaces](#)
 - [1.1.1. The DOM Structure Model](#)
 - [1.1.2. Memory Management](#)
 - [1.1.3. Naming Conventions](#)
 - [1.1.4. Inheritance vs. Flattened Views of the API](#)
 - [1.1.5. The DOMString type](#)
 - [1.1.6. String comparisons in the DOM](#)
- [1.2. Fundamental Interfaces](#)
 - [DOMException](#), [ExceptionCode](#), [DOMImplementation](#), [DocumentFragment](#), [Document](#), [Node](#), [NodeList](#), [NamedNodeMap](#), [CharacterData](#), [Attr](#), [Element](#), [Text](#), [Comment](#)
- [1.3. Extended Interfaces](#)
 - [CDATASection](#), [DocumentType](#), [Notation](#), [Entity](#), [EntityReference](#), [ProcessingInstruction](#)

1.1. Overview of the DOM Core Interfaces

This section defines a set of objects and interfaces for accessing and manipulating document objects. The functionality specified in this section (the *Core* functionality) is sufficient to allow software developers and web script authors to access and manipulate parsed HTML and XML content inside conforming products. The DOM Core API also allows creation and population of a [document](#) object using only DOM API calls; loading a [document](#) and saving it persistently is left to the product that implements the DOM API.

着重讲DOM level2

← → ↺ 🛡️ 安全 <https://www.w3.org/DOM/DOMTR#dom1> ☆ 🌐 📧 📄 📁 📌 📖 📑 📔 📕

Document Object Model (DOM) Technical Reports

See also [other W3C Technical Reports](#).

Table of contents

- [1. DOM4](#)
- [2. Document Object Model Level 3](#)
- [3. Document Object Model Level 2](#)
- [4. Document Object Model Level 1](#)
- [5. Others](#)

DOM4

[DOM4](#) (W3C Working Draft)
Anne van Kesteren, Aryeh Gregor, Ms2ger

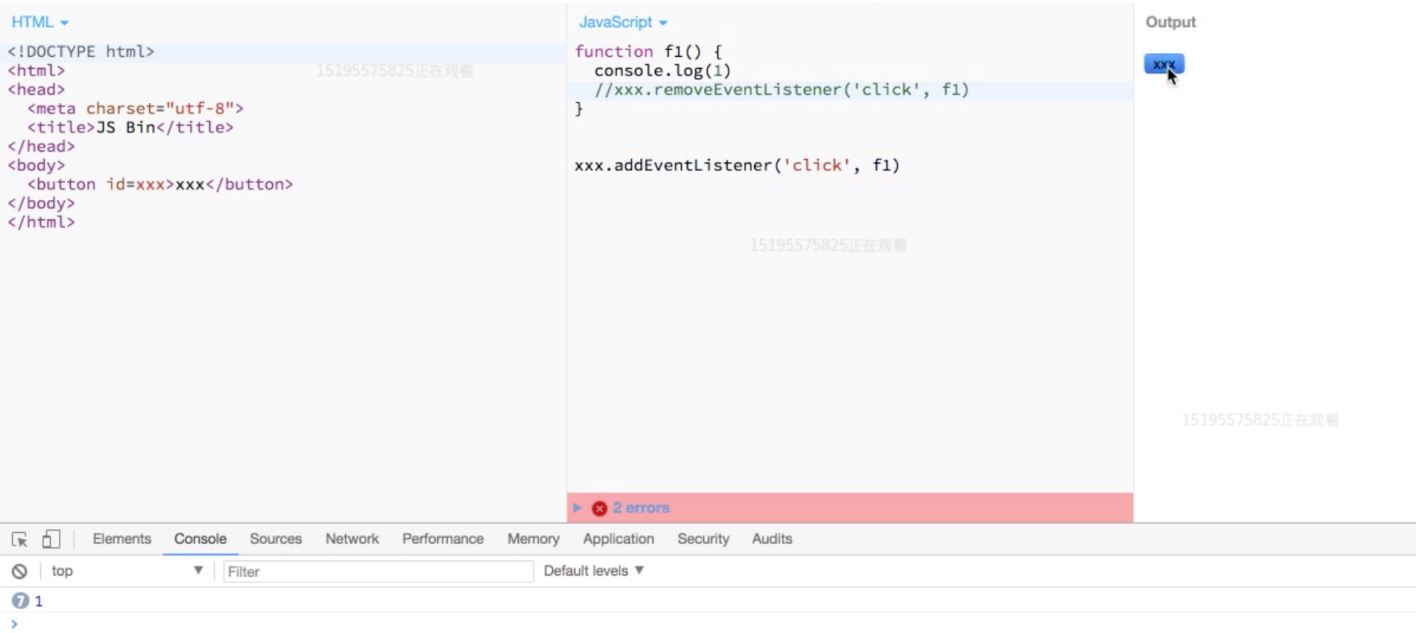
Document Object Model Level 3

[Document Object Model Level 3 Core](#) (W3C [Recommendation](#))
7 April 2004, Arnaud Le Hors, Philippe Le Hégarret
[\[errata\]](#) [\[implementation report\]](#) [\[issues list\]](#) [\[IPR Disclosures\]](#)
[Document Object Model Level 3 Load and Save](#) (W3C [Recommendation](#))
7 April 2004, Johnny Stenback
[\[errata\]](#) [\[implementation report\]](#) [\[issues list\]](#) [\[IPR Disclosures\]](#)
[Document Object Model Level 3 Validation](#) (W3C [Recommendation](#))
15 December 2003, Ben Chang
[\[errata\]](#) [\[implementation report\]](#) [\[issues list\]](#) [\[IPR Disclosures\]](#)

Document Object Model Level 2

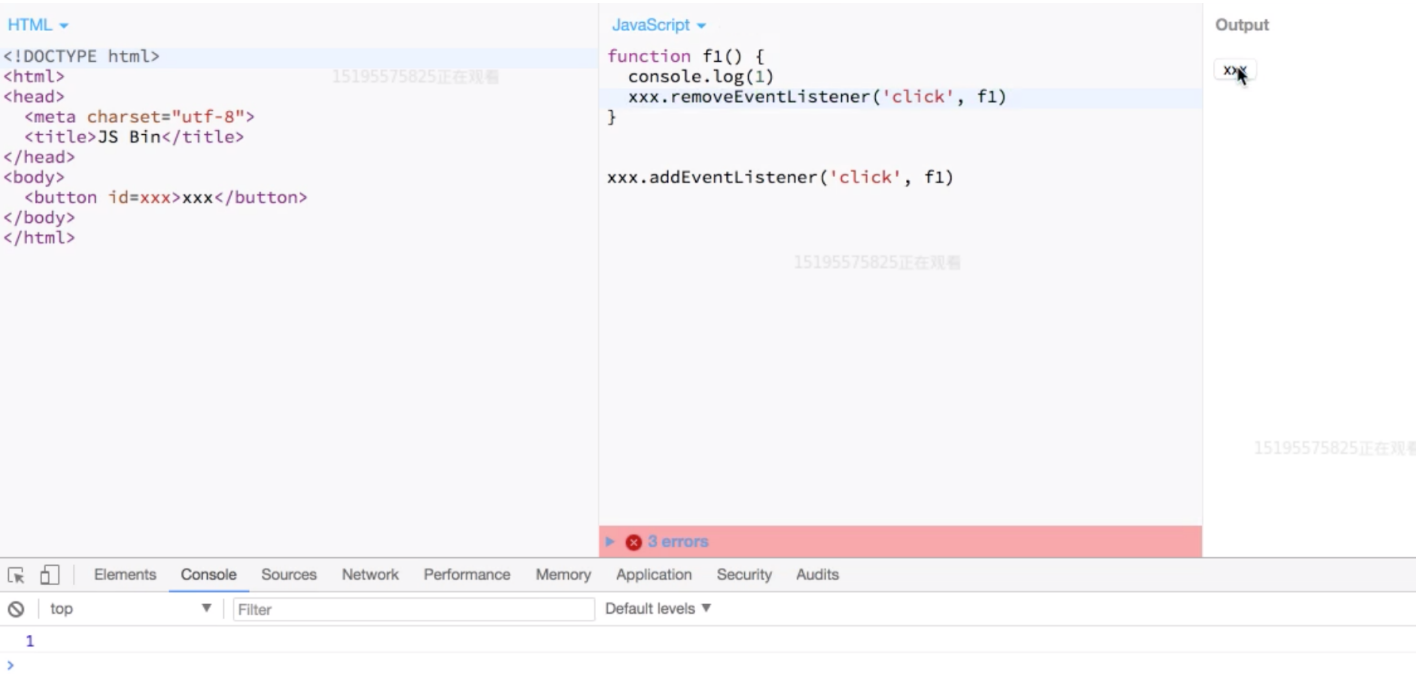
[Document Object Model Level 2 Core](#) (W3C [Recommendation](#))

这里会不断的输出1是因为一直在队列里面;



one函数

这里只是输出一一次1是因为把f1从队列里面拿走了;(One这个函数就是一次事件监听)



执行顺序

HTML ▾

```
<!DOCTYPE html>
<html>

<head>
  <meta charset="utf-8">
  <title>JS Bin</title>
</head>

<body>
  <div id="grand1">
    爷爷
    <div id="parent1">
      父亲
      <div id="child1">
        儿子
      </div>
    </div>
  </div>

</body>

</html>
```

15195575825正在观看

JavaScript ▾

```
// 1 当我点击儿子的时候，我是否点击了父亲和爷爷
// yes

// 2 当我点击儿子的时候，三个函数是否调用
grand1.addEventListener('click', function fn1(){
  console.log('爷爷')
})
parent1.addEventListener('click', function fn2(){
  console.log('爸爸')
})
child1.addEventListener('click', function fn3(){
  console.log('儿子')
})
// yes

// 3 请问 fn1 fn2 fn3 的执行顺序
// 1 2 3 or 3 2 1

// W3C: 都可以

// false: undefined null 0 '' NaN
// 不传第三个参数/ 传false 儿子爸爸爷爷
// 传第三个参数为 true 爷爷爸爸儿子
```

Output

Run with JS Auto-run JS

爷爷

父亲

儿子

15195575825正在观看

151955758