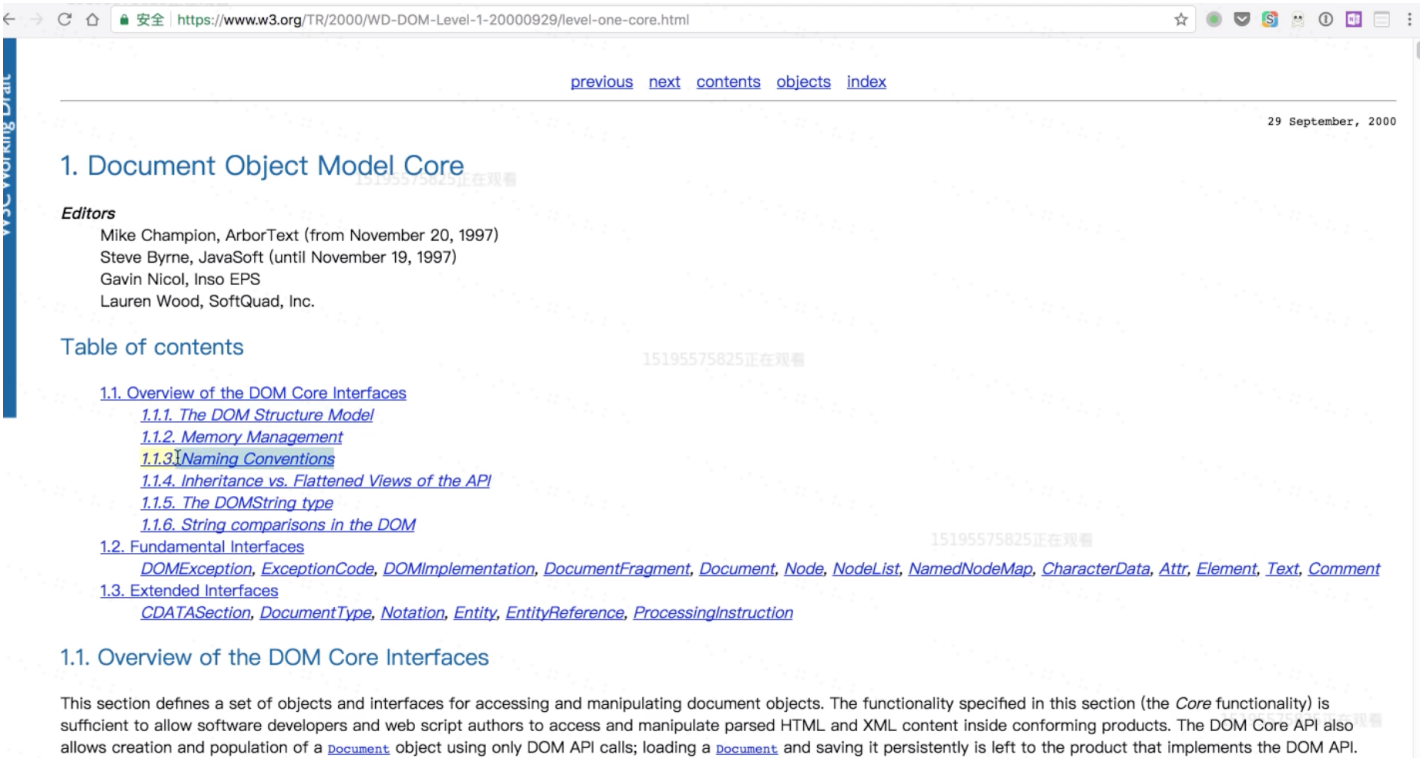
队列(重要)

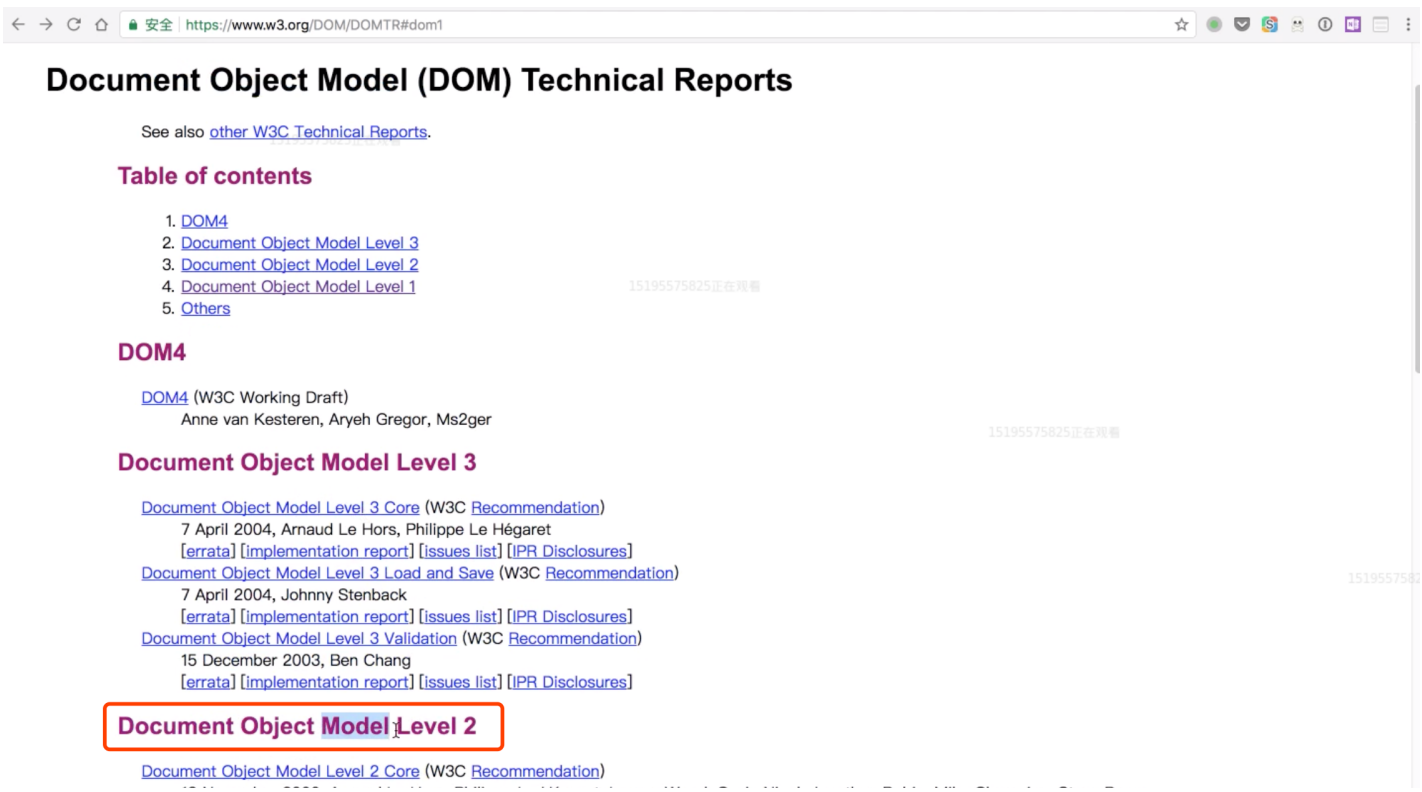
one函数

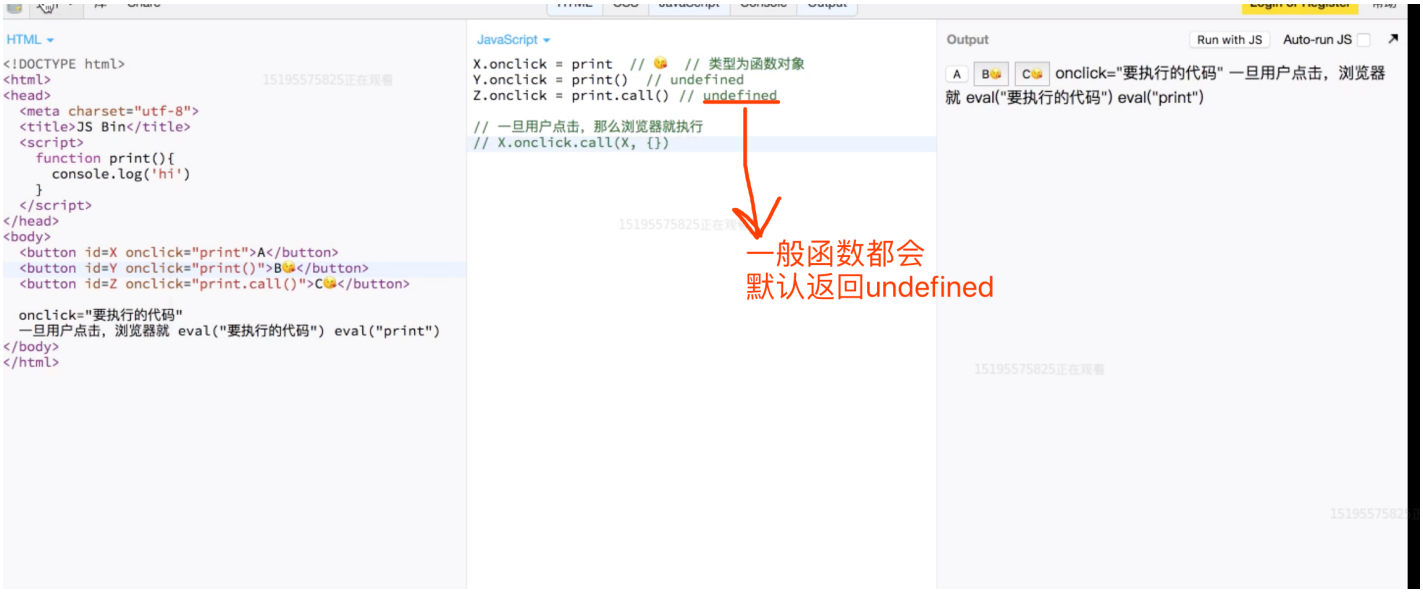
执行顺序

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

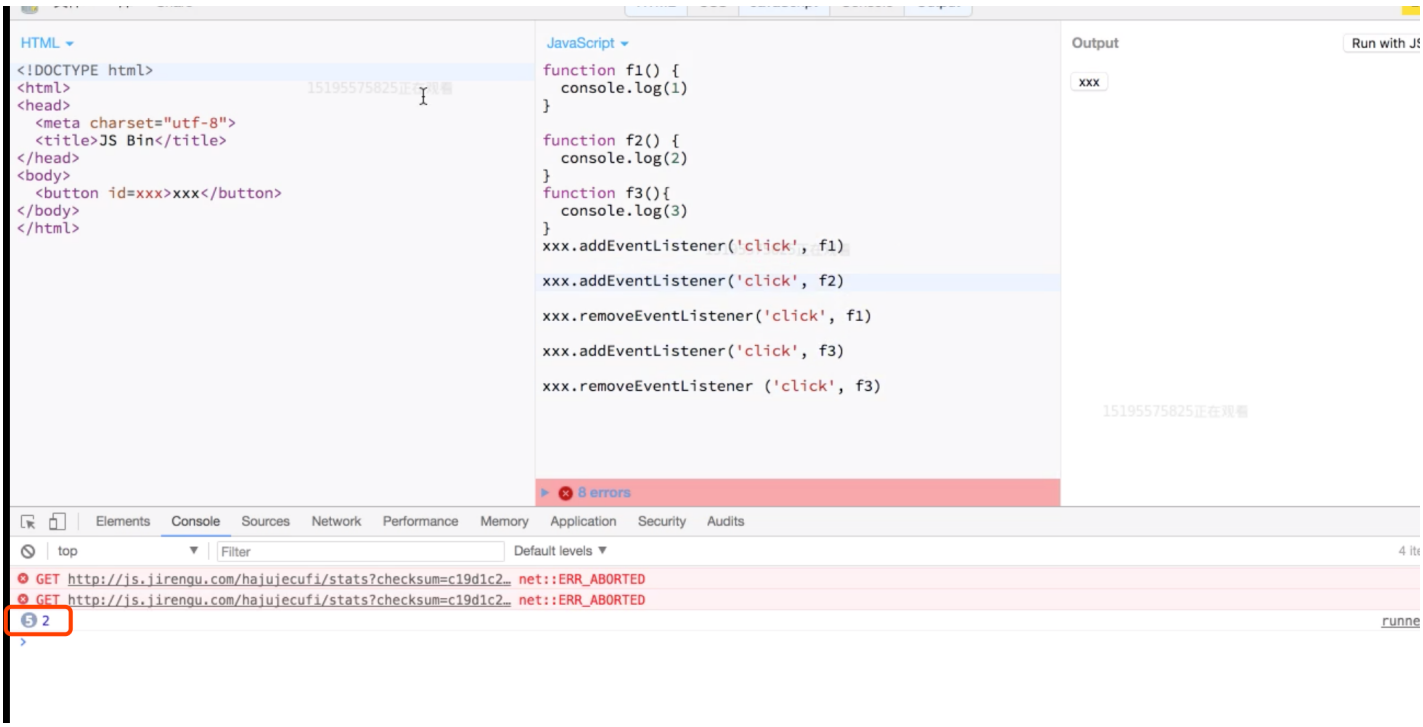



着重讲DOM level2

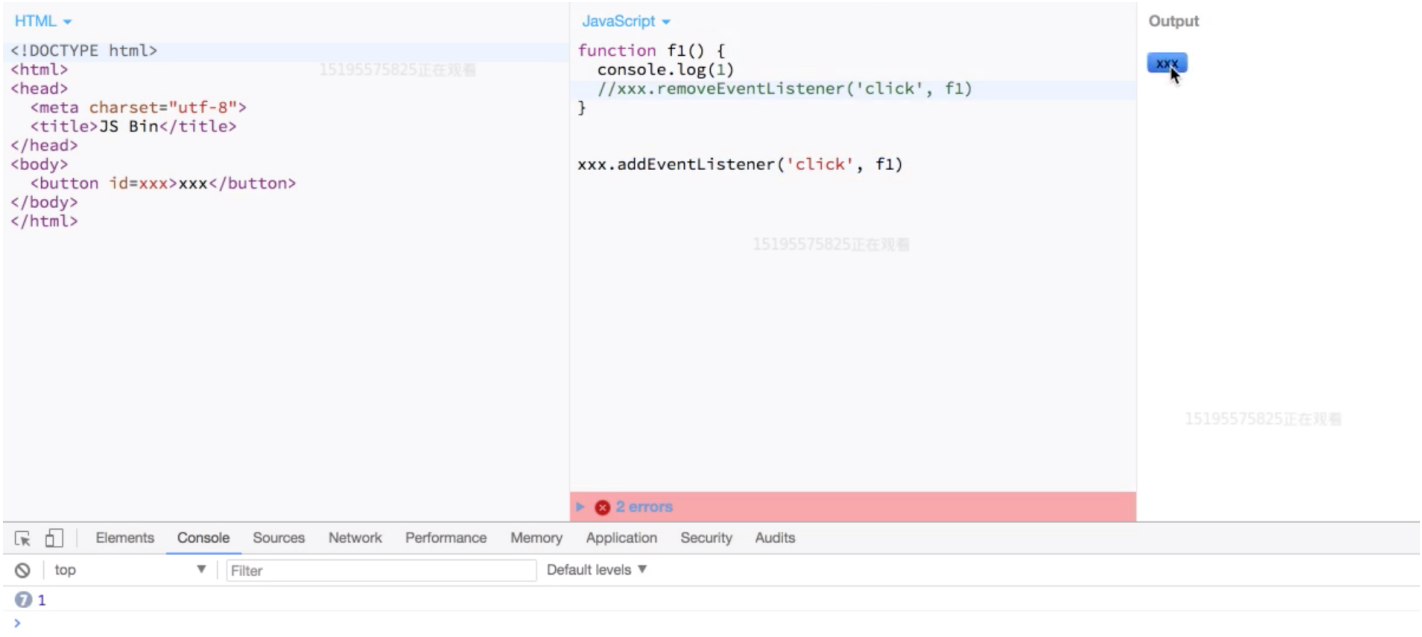



HTML
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>JS Bin</title>
<script>
function print(){
console.log('hi')
}
</script>
</head>
<body>
<button id=X onclick="print">A</button>
<button id=Y onclick="print()">B</button>
<button id=Z onclick="print.call()">C</button>
onclick="要执行的代码"
一旦用户点击，浏览器就 eval("要执行的代码") eval("print")
</body>
</html>
JavaScript
X.onclick = print // // 类型为函数对象
Y.onclick = print() // undefined
Z.onclick = print.call() // undefined
// 一旦用户点击，那么浏览器就执行
// X.onclick.call(X, {})
一般函数都会
默认返回undefined
Output
Run with JS
Auto-run JS
A B C onclick="要执行的代码" 一旦用户点击，浏览器就 eval("要执行的代码") eval("print")

# 队列(重要)

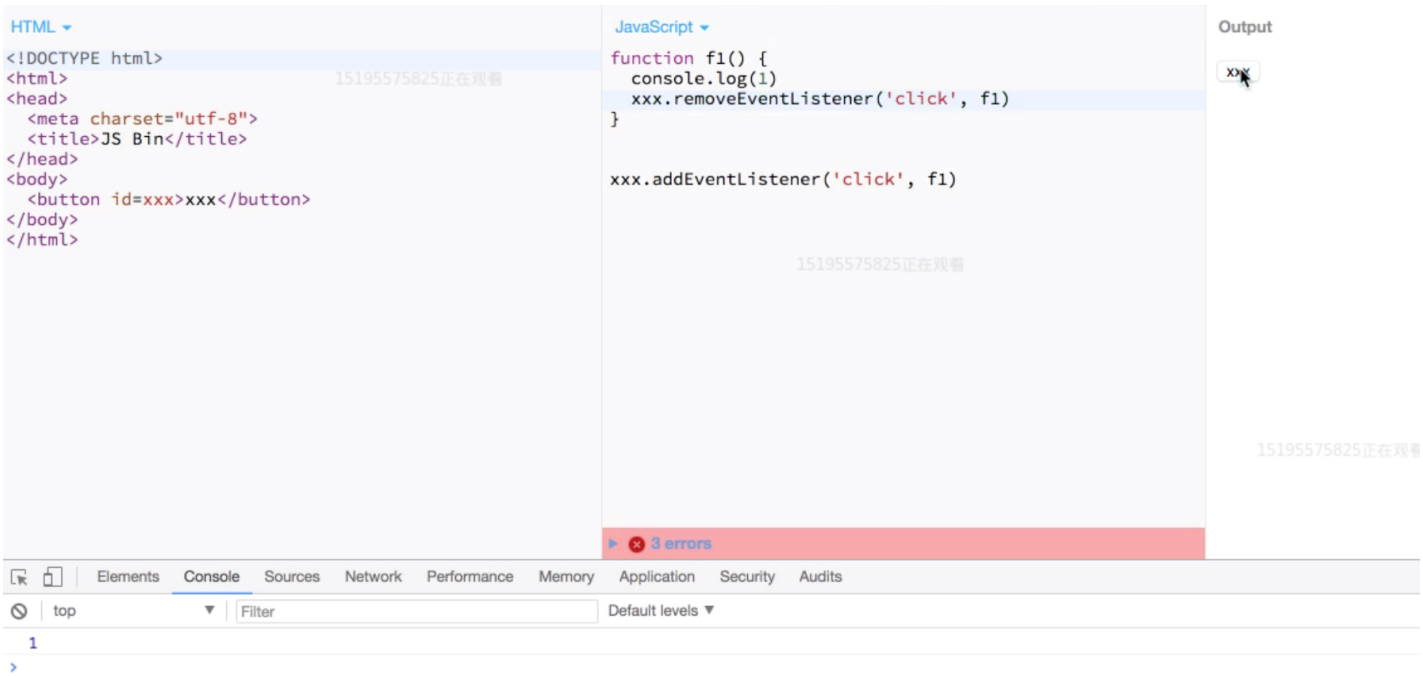

HTML
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>JS Bin</title>
</head>
<body>
<button id=xxx>xxx</button>
</body>
</html>
JavaScript
function f1() {
console.log(1)
}
function f2() {
console.log(2)
}
function f3(){
console.log(3)
}
xxx.addEventListener('click', f1)
xxx.addEventListener('click', f2)
xxx.removeEventListener('click', f1)
xxx.addEventListener('click', f3)
xxx.removeEventListener ('click', f3)
8 errors
Output
Run with JS
xxx
Elements Console Sources Network Performance Memory Application Security Audits
top
Filter
Default levels
GET http://js.jirengu.com/hajujecufi/stats?checksum=c19d1c2… net::ERR_ABORTED
GET http://js.jirengu.com/hajujecufi/stats?checksum=c19d1c2… net::ERR_ABORTED
2

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

```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>
```

JavaScript

```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