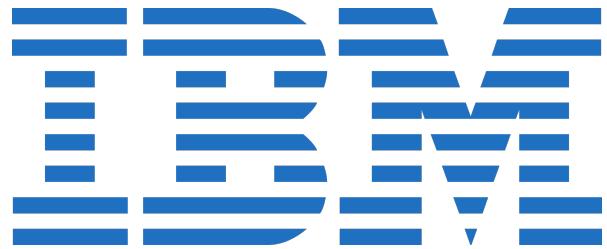


# 前端生产环境部署 ES6 代码

邓钢  
饿了么大前端

2017.thegiac.com

SDN OA  
盛大网络



→ 饿了么



2015 年 6 月

ECMAScript 2015 (ES6)

1999 年 12 月

ECMAScript 3 (ES3)

2009 年 12 月

ECMAScript 5 (ES5)

2015 年 6 月

ECMAScript 2015 (ES6)

1999 年 12 月

ECMAScript 3 (ES3)

2009 年 12 月

ECMAScript 5 (ES5)

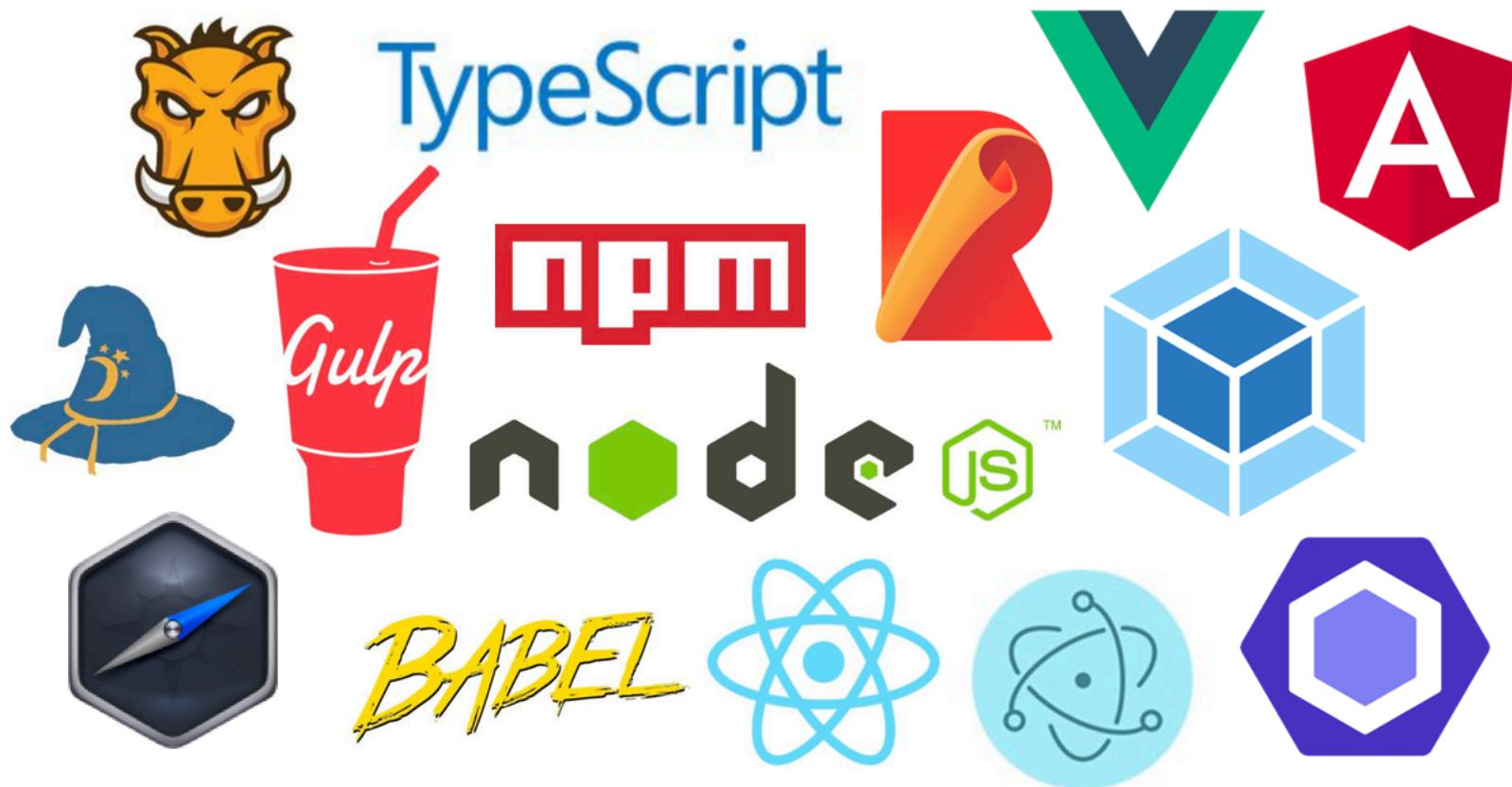
2015 年 6 月

ECMAScript 2015 (ES6)

2016 年 4 月

淘宝和天猫正式停止支持 IE6

## 前端工程化/工具化



## ECMAScript 6 compatibility table

compatibility table

Sort by Engine types ▾ Show obsolete platforms □ Show unstable platforms □

V8 SpiderMonkey JavaScriptCore Chakra Carakan KJS Other  
 Minor difference (1 point) Small feature (2 points) Medium feature (4 points) Large feature (8 points)

Feature name	Current browser	Compilers/polyfills										Desktop browsers										Servers/runtimes										Mobile	
		Traceur	Babel + core-js <sup>[2]</sup>	Closure	Type-Script + core-js	es6-shim	Kong 4.14 <sup>[3]</sup>	IE 11	Edge 15	Edge 16	FF 52 ESR	FF 56	FF 57	CH 62, OP 49 <sup>[1]</sup>	CH 63, OP 50 <sup>[1]</sup>	SF 10.1	SE 11	PJS	XSS	JXA	Node 4 <sup>[5]</sup>	Node ≥ 6.5 < 7 <sup>[5]</sup>	Node ≥ 8.7 < 9 <sup>[5]</sup>	DUK 1.8	DUK 2.2	iOS ≥ 10.3 < 11	iOS 11						
proper tail calls (tail call optimisation)	▶ 0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2						
<b>Syntax</b>																																	
default function parameters	▶ 7/7	4/7	4/7	5/7	5/7	0/7	0/7	0/7	7/7	7/7	6/7	7/7	7/7	7/7	7/7	7/7	0/7	7/7	0/7	0/7	7/7	7/7	0/7	0/7	7/7	7/7							
rest parameters	▶ 5/5	4/5	3/5	2/5	4/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	5/5	0/5	0/5	5/5	5/5	0/5	0/5	5/5	5/5							
spread (...) operator	▶ 15/15	15/15	13/15	12/15	4/15	0/15	0/15	0/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	15/15	0/15	0/15	15/15	15/15							
object literal extensions	▶ 6/6	6/6	6/6	4/6	6/6	0/6	0/6	0/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	6/6	0/6	6/6	0/6	0/6	6/6	6/6	0/6	0/6	4/6	6/6							
for...of loops	▶ 9/9	9/9	9/9	6/9	3/9	0/9	0/9	0/9	9/9	9/9	7/9	9/9	9/9	9/9	9/9	9/9	0/9	9/9	8/9	7/9	9/9	9/9	0/9	0/9	0/9	9/9							
octal and binary literals	▶ 4/4	2/4	4/4	4/4	2/4	0/4	0/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	4/4	0/4	4/4	4/4	4/4	4/4	4/4	0/4	4/4	4/4	4/4							
template literals	▶ 5/5	4/5	4/5	3/5	3/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	5/5	5/5	5/5	5/5	5/5	0/5	0/5	5/5	5/5							
RegExp "y" and "u" flags	▶ 5/5	3/5	3/5	0/5	0/5	0/5	0/5	0/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	5/5	0/5	2/5	0/5	0/5	5/5	5/5	0/5	0/5	5/5	5/5							
destructuring declarations	▶ 22/22	20/22	21/22	20/22	15/22	0/22	0/22	0/22	22/22	22/22	21/22	22/22	22/22	22/22	22/22	22/22	0/22	21/22	19/22	0/22	22/22	22/22	0/22	0/22	22/22	22/22							
destructuring assignment	▶ 24/24	23/24	24/24	21/24	19/24	0/24	0/24	0/24	24/24	24/24	23/24	24/24	24/24	24/24	24/24	24/24	0/24	24/24	21/24	0/24	24/24	24/24	0/24	0/24	24/24	24/24							
destructuring parameters	▶ 24/24	19/24	21/24	18/24	16/24	0/24	0/24	0/24	23/24	23/24	21/24	24/24	24/24	24/24	24/24	24/24	0/24	24/24	23/24	18/24	0/24	24/24	24/24	0/24	0/24	24/24	24/24						
Unicode code-point escapes	▶ 2/2	1/2	1/2	1/2	0/2	0/2	0/2	2/2	2/2	1/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	2/2	2/2							
new.target	▶ 2/2	0/2	0/2	0/2	0/2	0/2	0/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	0/2	2/2	0/2	0/2	2/2	2/2	0/2	1/2	2/2	2/2							
<b>Bindings</b>																																	
const	▶ 16/16	14/16	14/16	14/16	14/16	0/16	2/16	12/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	16/16	10/16	9/16	16/16	16/16	1/16	2/16	16/16	16/16							
let	▶ 12/12	10/12	10/12	10/12	10/12	0/12	0/12	10/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	12/12	0/12	12/12	0/12	6/12	12/12	12/12	0/12	0/12	12/12	12/12							
block-level function declaration <sup>[15]</sup>	▶ Yes	Yes	Yes	Yes	No	No	No	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No	Yes	Yes	No	No	Yes								
<b>Functions</b>																																	
arrow functions	▶ 13/13	11/13	9/13	10/13	9/13	0/13	0/13	0/13	13/13	13/13	13/13	13/13	13/13	13/13	13/13	0/13	12/13	0/13	9/13	13/13	13/13	0/13	0/13	13/13	13/13								
class	▶ 24/24	17/24	19/24	13/24	19/24	0/24	0/24	0/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	24/24	0/24	24/24	18/24	0/24	24/24	24/24	0/24	0/24	24/24	24/24							
super	▶ 8/8	7/8	4/8	6/8	7/8	0/8	0/8	0/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	8/8	0/8	8/8	8/8	7/8	0/8	8/8	8/8	0/8	0/8	8/8	8/8						

## ES5/6 构建文件尺寸和执行时间对比

Version	Size (minified)	Size (minified + gzipped)
ES2015+	80K	<b>21K</b>
ES5	175K	<b>43K</b>

Version	Parse/eval time (individual runs)	Parse/eval time (avg)
ES2015+	184ms, 164ms, 166ms	<b>172ms</b>
ES5	389ms, 351ms, 360ms	<b>367ms</b>



我有个大胆的想法

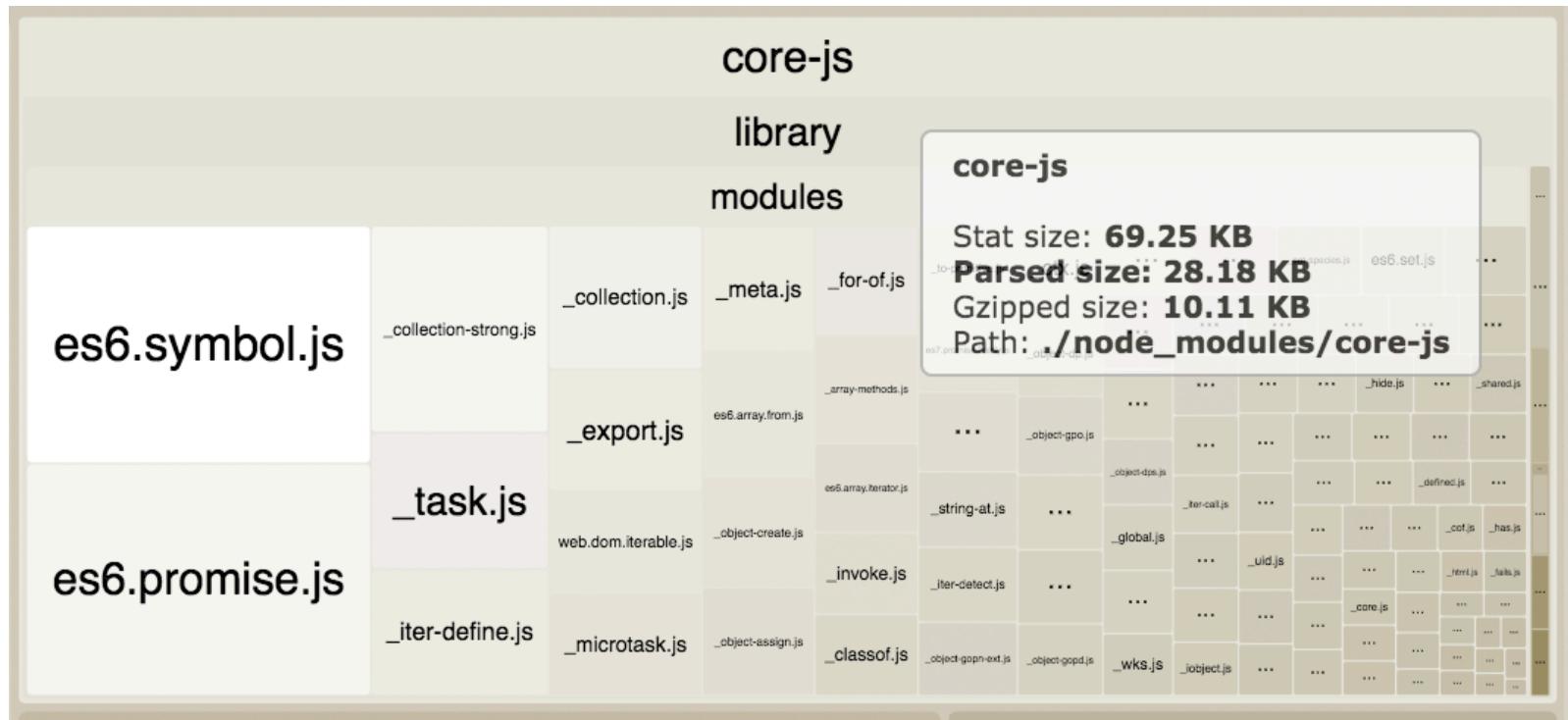
## babel / babel-preset-env / .babelrc

```
B .babelrc
1 {
2   "presets": [
3     ["env", {
4       "modules": false,
5       "useBuiltIns": true,
6       "targets": {
7         "browsers": [
8           "> 1%",
9           "last 2 versions",
10          "not ie <= 8",
11        ]
12      }
13    }]
14  ]
15 }
```



```
B .babelrc
1 {
2   "presets": [
3     ["env", {
4       "modules": false,
5       "useBuiltIns": true,
6       "targets": {
7         "browsers": [
8           "Chrome >= 60",
9           "Safari >= 10.1",
10          "iOS >= 10.3",
11          "Firefox >= 54",
12          "Edge >= 15"
13        ]
14      }
15    }]
16  ]
17 }
```

## core-js: 10.11 KB gzipped



regenerator: 2.37 KB gzipped

# regenerator-runtime

regenerator-runtime

Stat size: **24.61 KB**

**Parsed size: 6.43 KB**

Gzipped size: **2.37 KB**

Path: **./node\_modules/regenerator-runtime**



## 可降级的方案

- <script type="module"> 和 <script nomodule>

```
<!-- 支持 ES6 模块, 加载并执行 -->
<!-- 不支持 ES6 模块, 加载但不执行 -->
<script type="module" src="es2015.js"></script>

<!-- 支持 ES6 模块, 不加载 -->
<!-- 不支持 ES6 模块, 加载并执行 -->
<script nomodule src="es5.js"></script>
```

完美！



## ( 已知的 ) 坑

- 不支持 ES6 模块的浏览器会加载两份代码；
- <script type="module"> 和 <script type="text/javascript"> 的区别；
- webpack.optimize.UglifyJsPlugin 不认识 ES6 代码；
- <script type="module"> 自带 defer 效果；
- Safari 10 不支持 nomodule 属性；
- 无形中提高了准入门槛；

## (已知的) 坑：内置 UglifyJsPlugin 不认识 ES6 代码

- 使用更新的 UglifyJS 插件

```
const UglifyJsPlugin = require('uglifyjs-webpack-plugin')

module.exports = {
  plugins: [
    new UglifyJsPlugin()
  ]
}
```

## (已知的) 坑：`<script type="module">`自带 defer 效果

- `<link rel="preload">`

```
<link rel="preload" as="script" href="es2015.js" crossorigin>
```

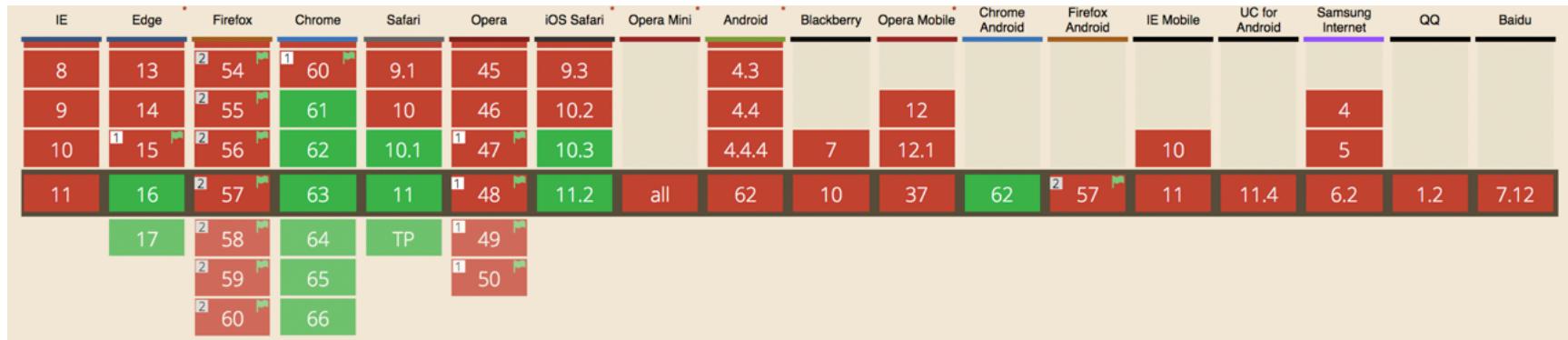
## (已知的) 坑 : Safari 10 不支持 nomodule 属性

```
(function () {
    var check = document.createElement('script')
    if (!('noModule' in check) && 'onbeforeload' in check) {
        var support = false
        document.addEventListener('beforeload', function (e) {
            if (e.target === check) {
                support = true
            } else if (!e.target.hasAttribute('nomodule') || !support) {
                return
            }
            e.preventDefault()
        }, true)

        check.type = 'module'
        check.src = '.'
        document.head.appendChild(check)
        check.remove()
    }
})()
```

## ( 已知的 ) 坑 : 无形中提高了准入门槛

- <script type="module"> 的支持程度



( 惨不忍睹 )

## 终于上了生产

- [https://p\\*\\*\\*\\*i.ele.me/](https://p****i.ele.me/)

版本	命中占比		生产依赖			业务代码		
	预期	实际	打包尺寸	Gzipped	执行时间	打包尺寸	Gzipped	执行时间
ES5	40-%	67%	307K	94.6K	75ms	288K	96.7K	636ms
ES6	60+%	33%	298K	92.3K	67ms	274K	105K	233ms

如果生产依赖也提供 ES6 版本.....

vuejs / vue

Code Issues 61 Pull requests 17 Projects 0

## Provide ES2015 bundle? #6937

Open myst729 opened this issue on Oct 27 • 2 comments

myst729 commented on Oct 27 • edited

What problem does this feature solve?

## ( 后来才知道的 ) 坑

- [Safari] SyntaxError: Cannot declare a let variable twice.
- 各种魔改浏览器、WebView.....

## (后来才知道的) 坑 : Safari SyntaxError

- SyntaxError: Cannot declare a let variable twice.

```
let e = e => {
  console.log(e)
  for (let e of [1, 2, 3]) // Cannot declare a let variable twice: 'e'.
    console.log(e)
}
```

## (后来才知道的) 坑 : Safari SyntaxError

- UglifyJS: { mangle: { safari10: true } }

```
const UglifyJsPlugin = require('uglifyjs-webpack-plugin')

module.exports = {
  plugins: [
    new UglifyJsPlugin({
      mangle: {
        safari10: true
      }
    })
  ]
}
```

- babel-minify

## ( 后来才知道的 ) 坑 : 各种魔改浏览器、WebView.....

- 没有遇到过的新问题.....
- 占比太少，暂时不管了.....

**还是回到 ES5 吧.....**

**还有救吗？**

## 动态服务

- 服务器根据用户代理区分客户端对 ES6 的支持程度，返回不同的内容
- Vary: User-Agent

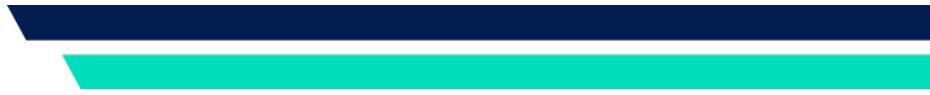
## 动态服务

- 好处：ES6 门槛降低；
- 好处：ES5 用户不需要下载 ES6 代码；
- 好处：让浏览器尽情地暴力优化；
- 风险：误判可能造成服务无法使用！
- 难点：移动客户端分裂严重，白名单维护成本高

**编写 ES6 只是程序员的胜利  
部署 ES6 才是与用户的双赢**

## 参考资料

- <https://philipwalton.com/articles/deploying-es2015-code-in-production-today/>
- <https://kangax.github.io/compat-table/es6/>
- <https://webpack.js.org/plugins/uglifyjs-webpack-plugin/>
- <https://gist.github.com/samthor/64b114e4a4f539915a95b91ffd340acc>
- <https://www.w3.org/TR/preload/>
- <https://www.caniuse.com/#feat=es6-module>
- <https://github.com/webpack-contrib/uglifyjs-webpack-plugin/issues/92>
- <https://github.com/mishoo/UglifyJS2/tree/harmony#mangle-options>
- [https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Vary#Dynamic\\_serving](https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Vary#Dynamic_serving)
- <https://www.fastly.com/blog/best-practices-using-vary-header/>



# ARIA 实践指南

<https://github.com/ElemeFE/WAI-ARIA-Practices>

GIAC | 全球互联网架构大会  
GLOBAL INTERNET ARCHITECTURE CONFERENCE

GIAC

# 全球互联网架构大会

GLOBAL INTERNET ARCHITECTURE CONFERENCE



扫码关注GIAC公众号

2017.thegiac.com