

聊聊JS的几个新特性

QConShanghai 2019 —— 编程语言

<https://johnhax.net/2019/js-new-features/slide?qcon#0>

GitHub @hax

知乎 @贺师俊

微博 @johnhax

Twitter @haxy

<https://johnhax.net/2019/js-new-features/slide?qcon#1>

1998

<https://johnhax.net/2019/js-new-features/slide?qcon#2>

IE 4

<https://johnhax.net/2019/js-new-features/slide?qcon#3>

一个写了20+年JS的男人

<https://johnhax.net/2019/js-new-features/slide?qcon#4>

- 前ES3时代 1995~1999
- ES3时代 2000~2010
- Harmony时代 2008~2016
- 后ES6时代 2016~2020
- TS/JS生态新时代 2020~

<https://johnhax.net/2019/js-new-features/slide?qcon#5>

360

75team

<https://johnhax.net/2019/js-new-features/slide?qcon#6>

360高级前端架构师

360技术委员会Web前端TC委员

<https://johnhax.net/2019/js-new-features/slide?qcon#7>

2019年7月起 TC39代表

<https://johnhax.net/2019/js-new-features/slide?qcon#8>

ECMA

<https://johnhax.net/2019/js-new-features/slide?qcon#9>

- ECMA-334 — C# Language Specification
- ECMA-335 — Common Language Infrastructure (CLI)
- ECMA-367 — Eiffel programming Language
- ECMA-376 — Office Open XML (ISO/IEC 29500)
- ECMA-388 — Open XML Paper Specification
- ECMA-408 — Dart language specification
- ECMA-410 — NFC-SEC Entity Authentication and Key Agreement using Asymmetric Cryptography
- ECMA-411 — NFC-SEC Entity Authentication and Key Agreement using Symmetric Cryptography

<https://johnhax.net/2019/js-new-features/slide?qcon#10>

China IWNCOMM

西电捷通

2014-6~2018-6?

<https://johnhax.net/2019/js-new-features/slide?qcon#11>

- ECMA-262 — ECMAScript Language Specification
- ECMA-357 — ECMAScript for XML (E4X) (废弃)
- ECMA-402 — ECMAScript Internationalization API
- ECMA-404 — JSON

<https://johnhax.net/2019/js-new-features/slide?qcon#12>

JavaScript

ECMAScript

<https://johnhax.net/2019/js-new-features/slide?qcon#13>

JavaScript®

ECMAScript®

<https://johnhax.net/2019/js-new-features/slide?qcon#14>

TC39

<https://johnhax.net/2019/js-new-features/slide?qcon#15>

- Google
- Apple
- Mozilla
- Microsoft
- Facebook

<https://johnhax.net/2019/js-new-features/slide?qcon#16>

- IBM
- Intel
- Paypal
- Stripe

<https://johnhax.net/2019/js-new-features/slide?qcon#17>

- 360
- Airbnb
- Bloomberg
- GoDaddy
- Salesforce

<https://johnhax.net/2019/js-new-features/slide?qcon#18>

- Evernote
- Igalia
- Meteor
- npm
- OpenJS Foundation

<https://johnhax.net/2019/js-new-features/slide?qcon#19>

建立渠道

<https://johnhax.net/2019/js-new-features/slide?qcon#20>

聊聊JS的 几个新特性

<https://johnhax.net/2019/js-new-features/slide?qcon#21>

已完成的提案

<https://johnhax.net/2019/js-new-features/slide?qcon#22>

每年6-8个提案

<https://johnhax.net/2019/js-new-features/slide?qcon#23>

ES2020

目前5个

<https://johnhax.net/2019/js-new-features/slide?qcon#24>

- `String.prototype.matchAll`
- `import()`
- `BigInt`
- `Promise.allSettled`
- `globalThis`

<https://johnhax.net/2019/js-new-features/slide?qcon#25>

Stage 3

<https://johnhax.net/2019/js-new-features/slide?qcon#26>

- Nullish Coalescing
- Optional Chaining
- `String.prototype.replaceAll` ?

<https://johnhax.net/2019/js-new-features/slide?qcon#27>

Nullish Coalescing Operators

<https://johnhax.net/2019/js-new-features/slide?qcon#28>



<https://johnhax.net/2019/js-new-features/slide?qcon#29>

a ? ? b

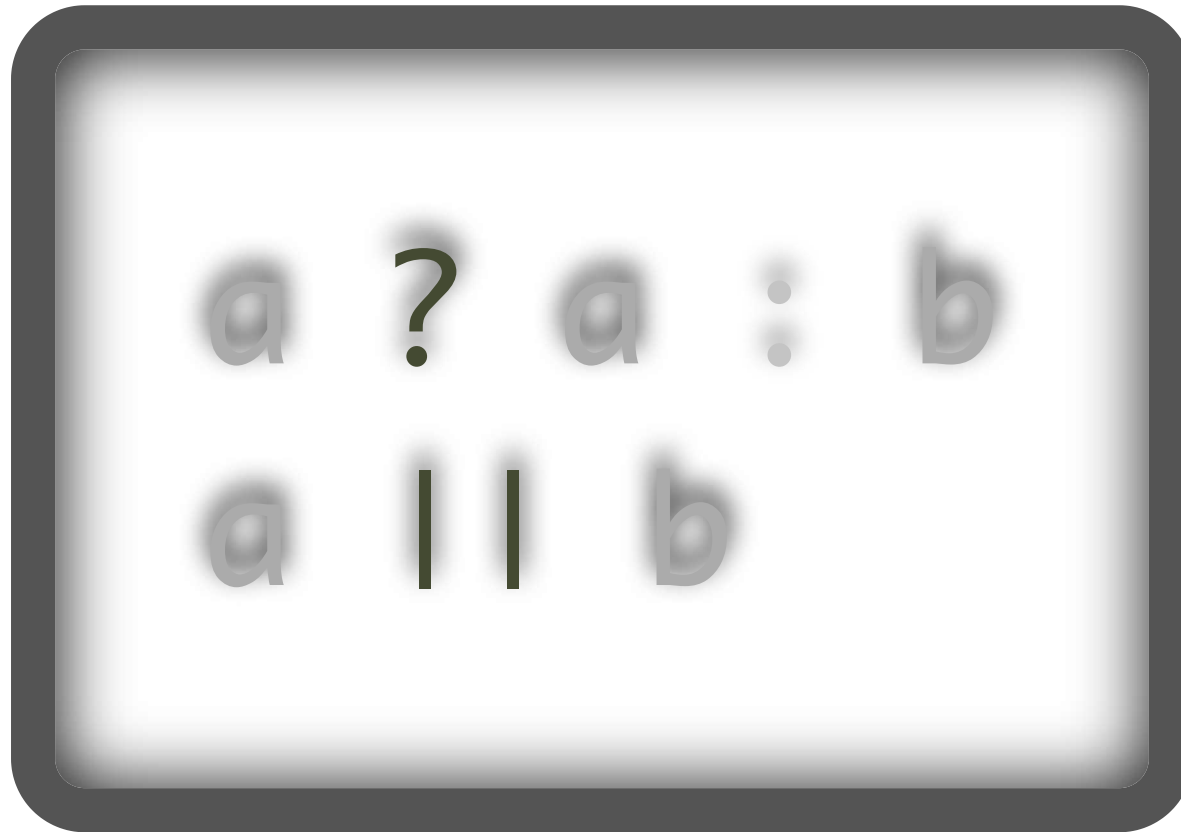
<https://johnhax.net/2019/js-new-features/slide?qcon#30>

```
a !== undefined && a !== null ? a : b
```

<https://johnhax.net/2019/js-new-features/slide?qcon#31>

`a !== null ? a : b`

<https://johnhax.net/2019/js-new-features/slide?qcon#32>



<https://johnhax.net/2019/js-new-features/slide?qcon#33>

truthy/falsy

<https://johnhax.net/2019/js-new-features/slide?qcon#34>

- string: "" || v
- boolean: false || v
- nubmer: 0 || v
- number: NaN || v
- Array: [] || v
- Object: ({ valueOf() { return 0 } }) || v

<https://johnhax.net/2019/js-new-features/slide?qcon#35>

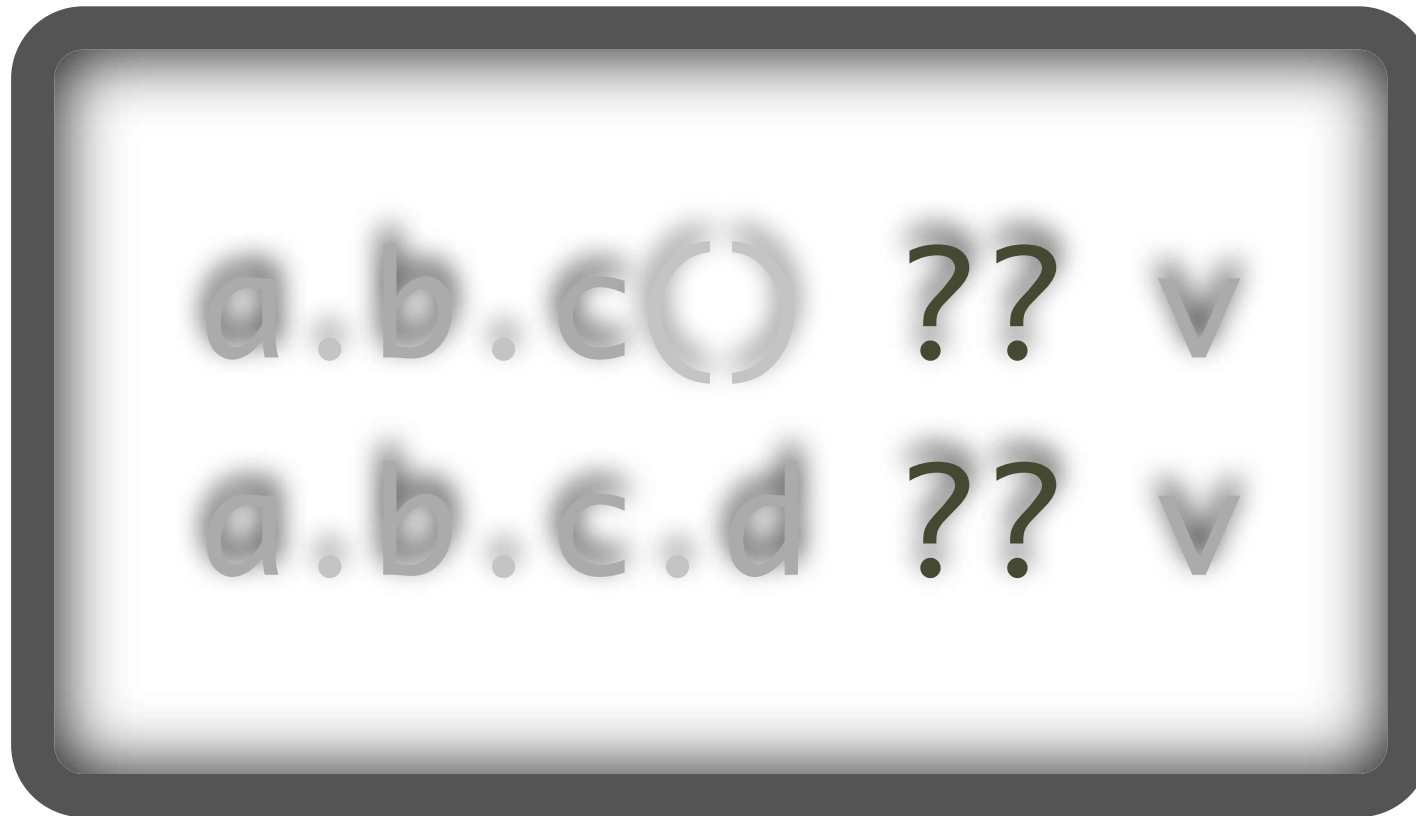
nullish

<https://johnhax.net/2019/js-new-features/slide?qcon#36>

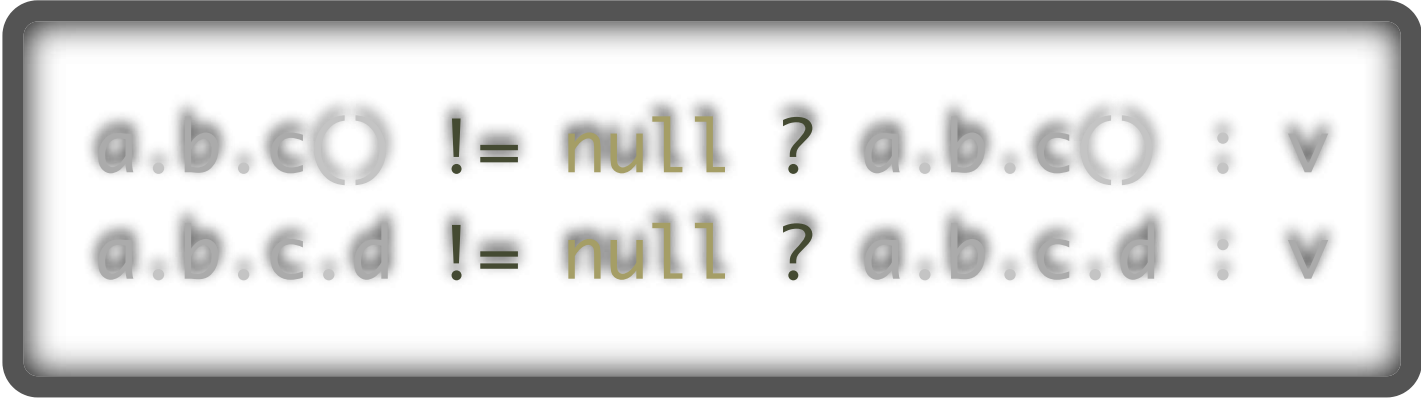


`a != null ? a : b`

<https://johnhax.net/2019/js-new-features/slide?qcon#37>



<https://johnhax.net/2019/js-new-features/slide?qcon#38>



```
a.b.c() != null ? a.b.c() : v  
a.b.c.d != null ? a.b.c.d : v
```

<https://johnhax.net/2019/js-new-features/slide?qcon#39>

Side effects

<https://johnhax.net/2019/js-new-features/slide?qcon#40>

- OAOO
- DRY

<https://johnhax.net/2019/js-new-features/slide?qcon#41>

- eval Once And Only Once
- Don't Repeat Yourself

<https://johnhax.net/2019/js-new-features/slide?qcon#42>

`a !== null ? a : b`

`a !== undefined && a !== null ? a : b`

<https://johnhax.net/2019/js-new-features/slide?qcon#43>

document.all

<https://johnhax.net/2019/js-new-features/slide?qcon#44>

TypeScript

<https://johnhax.net/2019/js-new-features/slide?qcon#45>

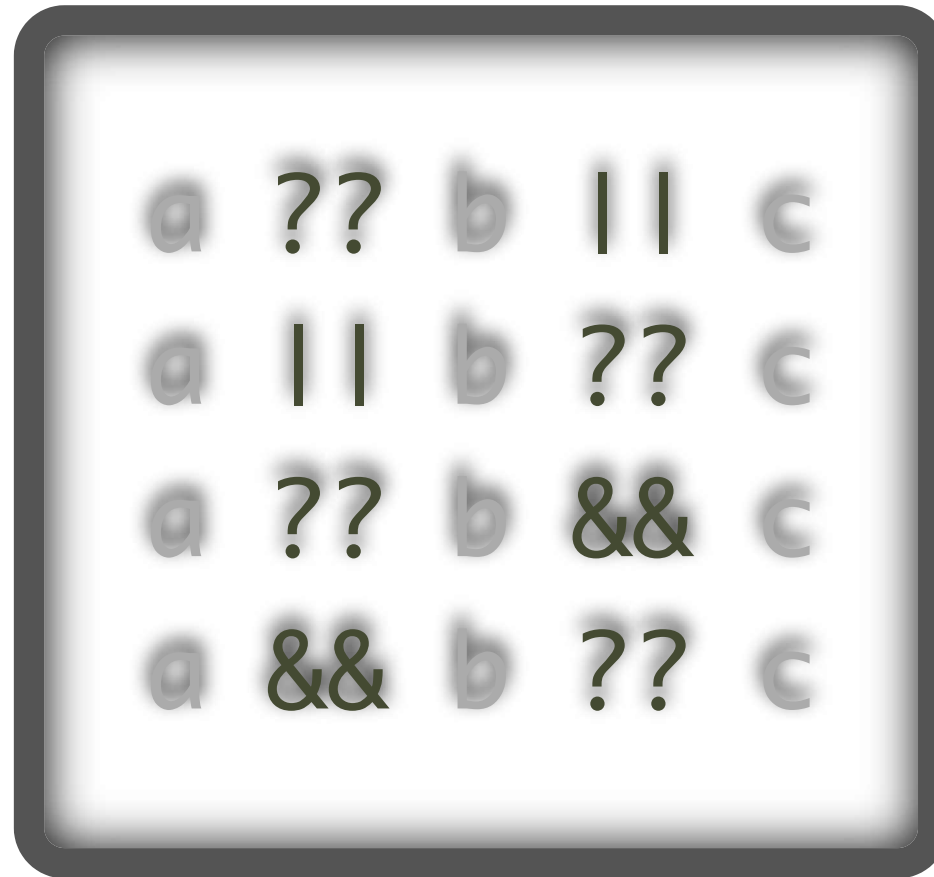
a ?? b

a 必须是 nullish

<https://johnhax.net/2019/js-new-features/slide?qcon#46>

- T | undefined
- T | null
- T | null | undefined

<https://johnhax.net/2019/js-new-features/slide?qcon#47>



<https://johnhax.net/2019/js-new-features/slide?qcon#48>

SyntaxError

<https://johnhax.net/2019/js-new-features/slide?qcon#49>

必须加括号

<https://johnhax.net/2019/js-new-features/slide?qcon#50>

未决问题 优先级

<https://johnhax.net/2019/js-new-features/slide?qcon#51>

- 等于 ?: —— GNU C、Groovy (? :)
- 小于 || —— C#、PHP、Dart、Coffee
- 等于 || —— Perl (//)
- 高于 ==: Swift、Kotlin

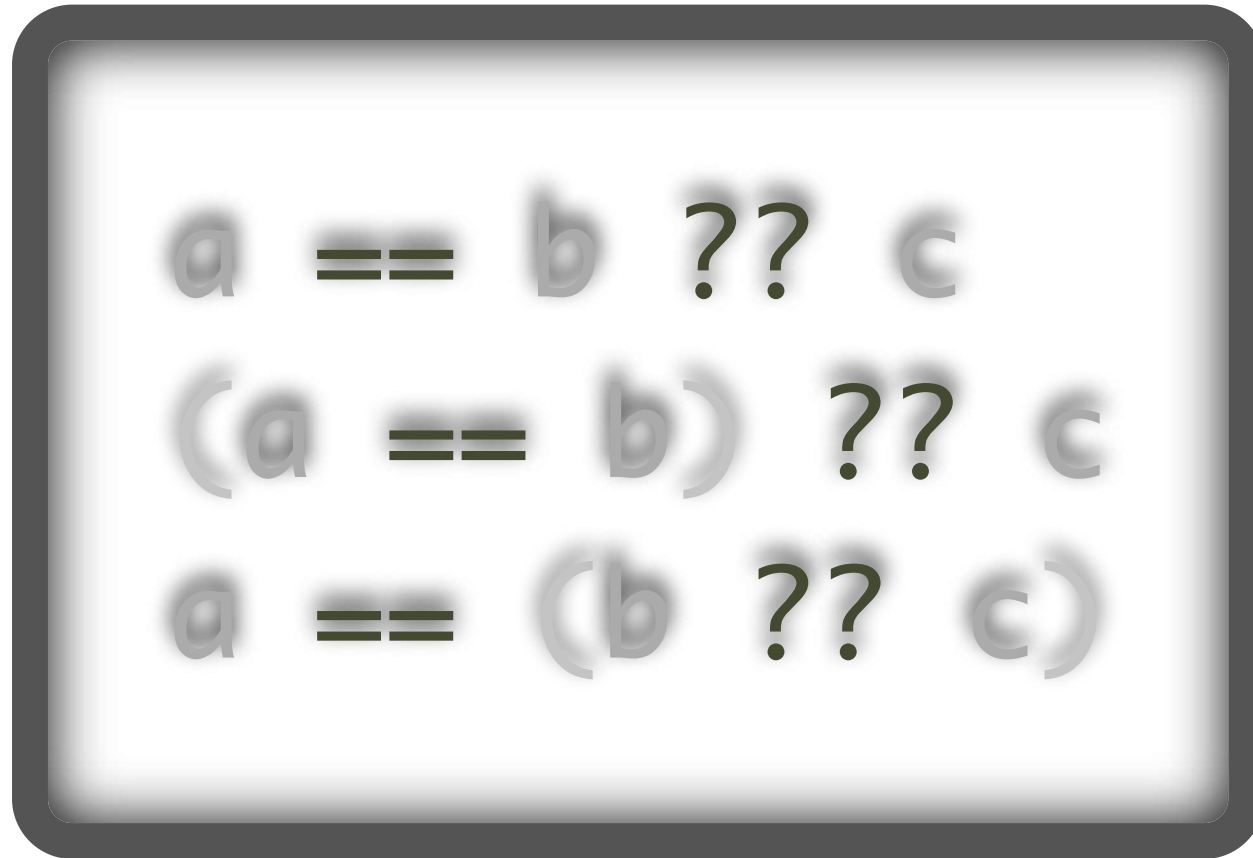
<https://johnhax.net/2019/js-new-features/slide?qcon#52>

v ?? complex_expression

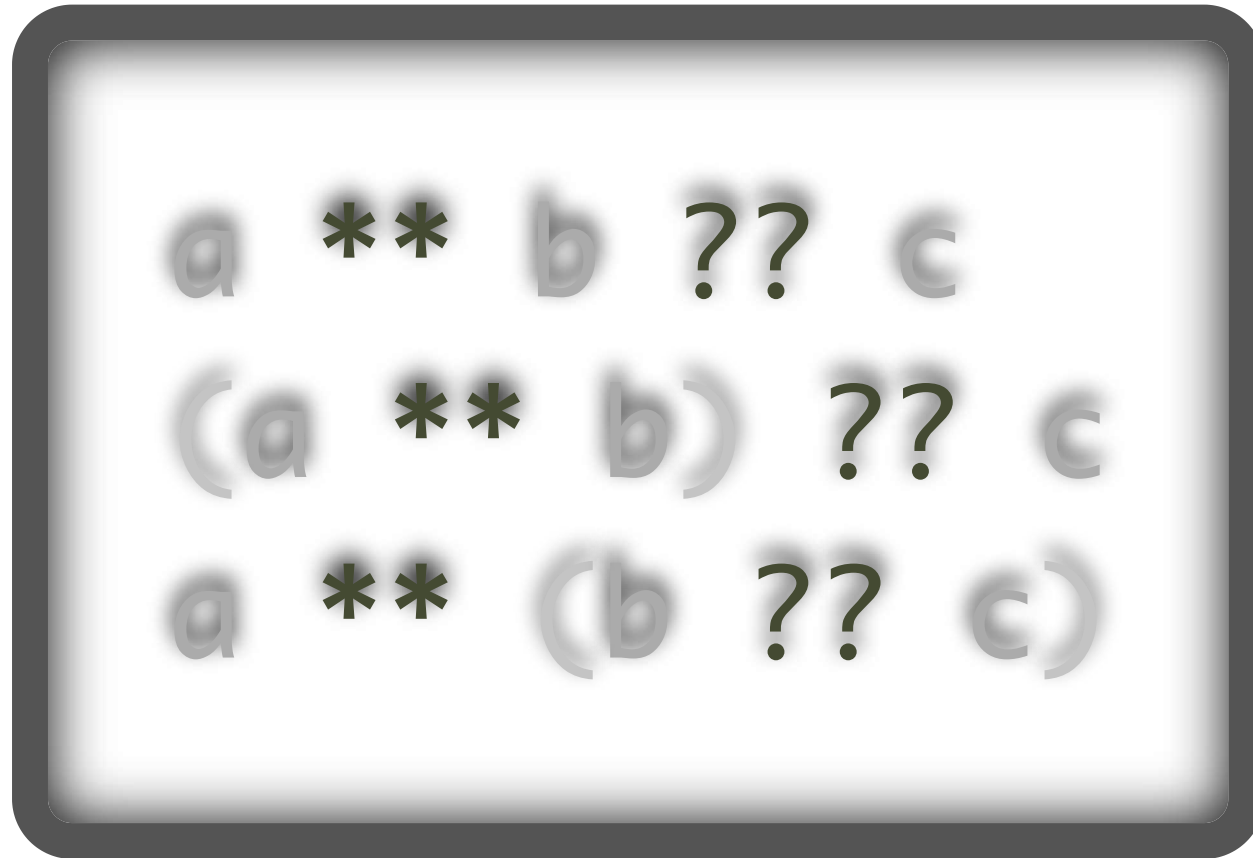
<https://johnhax.net/2019/js-new-features/slide?qcon#53>

complex_expression ?? v

<https://johnhax.net/2019/js-new-features/slide?qcon#54>



<https://johnhax.net/2019/js-new-features/slide?qcon#55>



<https://johnhax.net/2019/js-new-features/slide?qcon#56>

Operator precedence

<https://johnhax.net/2019/js-new-features/slide?qcon#57>

TypeScript

<https://johnhax.net/2019/js-new-features/slide?qcon#58>

a ?? b

a 必须是 nullish

<https://johnhax.net/2019/js-new-features/slide?qcon#59>

$(a \ [OP] \ b) \ \ ? \ ? \ c$
一定都是错误代码

<https://johnhax.net/2019/js-new-features/slide?qcon#60>

a [OP] b ?? c

一定都是错误代码

<https://johnhax.net/2019/js-new-features/slide?qcon#61>

a [OP] (b ?? c)

<https://johnhax.net/2019/js-new-features/slide?qcon#62>

a ?? b [OP] c

理解上存在歧义

<https://johnhax.net/2019/js-new-features/slide?qcon#63>

$a [OP] b \quad ?? \quad c [OP] d$

怎样自动fix?

$(a \ * [OP] \ * \ b) \quad ?? \quad (c \ * [OP] \ * \ d)$

$a [OP] (b \quad ?? \quad c [OP] d)$

$a [OP] (b \quad ?? \quad c) [OP] d$

<https://johnhax.net/2019/js-new-features/slide?qcon#64>

高于比较运算符： Swift、 Kotlin

<https://johnhax.net/2019/js-new-features/slide?qcon#65>

高于** (所有二元运算符)

<https://johnhax.net/2019/js-new-features/slide?qcon#66>

a [OP] b ?? c

<https://johnhax.net/2019/js-new-features/slide?qcon#67>

a [OP] b ?? c [OP] d

<https://johnhax.net/2019/js-new-features/slide?qcon#68>

a ?? b *[OP]* c

<https://johnhax.net/2019/js-new-features/slide?qcon#69>

比较保守的团队 可以定制lint规则

<https://johnhax.net/2019/js-new-features/slide?qcon#70>

相对简单的lint规则

只禁止 $a \ ?? \ b \ [OP] \ c$

$a \ ?? \ b$ 或者 $a \ ?? \ (b \ [OP] \ c)$

<https://johnhax.net/2019/js-new-features/slide?qcon#71>

而现在可能需要lint规则

- $a [OP] b \quad ?? \quad c$
- $a \quad ?? \quad b [OP] c$
- $a [OP] b \quad ?? \quad c [OP] d$

<https://johnhax.net/2019/js-new-features/slide?qcon#72>

伏生級
阮元級

<https://johnhax.net/2019/js-new-features/slide?qcon#73>

#15

<https://johnhax.net/2019/js-new-features/slide?qcon#74>

Optional Chaining

<https://johnhax.net/2019/js-new-features/slide?qcon#75>

a?.b

<https://johnhax.net/2019/js-new-features/slide?qcon#76>

`a !== undefined && a !== null ? a.b : undefined`

<https://johnhax.net/2019/js-new-features/slide?qcon#77>

`a !== null ? a.b : undefined`

<https://johnhax.net/2019/js-new-features/slide?qcon#78>

Nullish-aware Operators

<https://johnhax.net/2019/js-new-features/slide?qcon#79>

`a !== null ? a.b : undefined`

<https://johnhax.net/2019/js-new-features/slide?qcon#80>

a && a.b

<https://johnhax.net/2019/js-new-features/slide?qcon#81>

truthy/falsy

nullish

<https://johnhax.net/2019/js-new-features/slide?qcon#82>

`a && a.b && a.b.c`

<https://johnhax.net/2019/js-new-features/slide?qcon#83>

Side effects

<https://johnhax.net/2019/js-new-features/slide?qcon#84>

- OAOO
- DRY

<https://johnhax.net/2019/js-new-features/slide?qcon#85>

document.all

<https://johnhax.net/2019/js-new-features/slide?qcon#86>

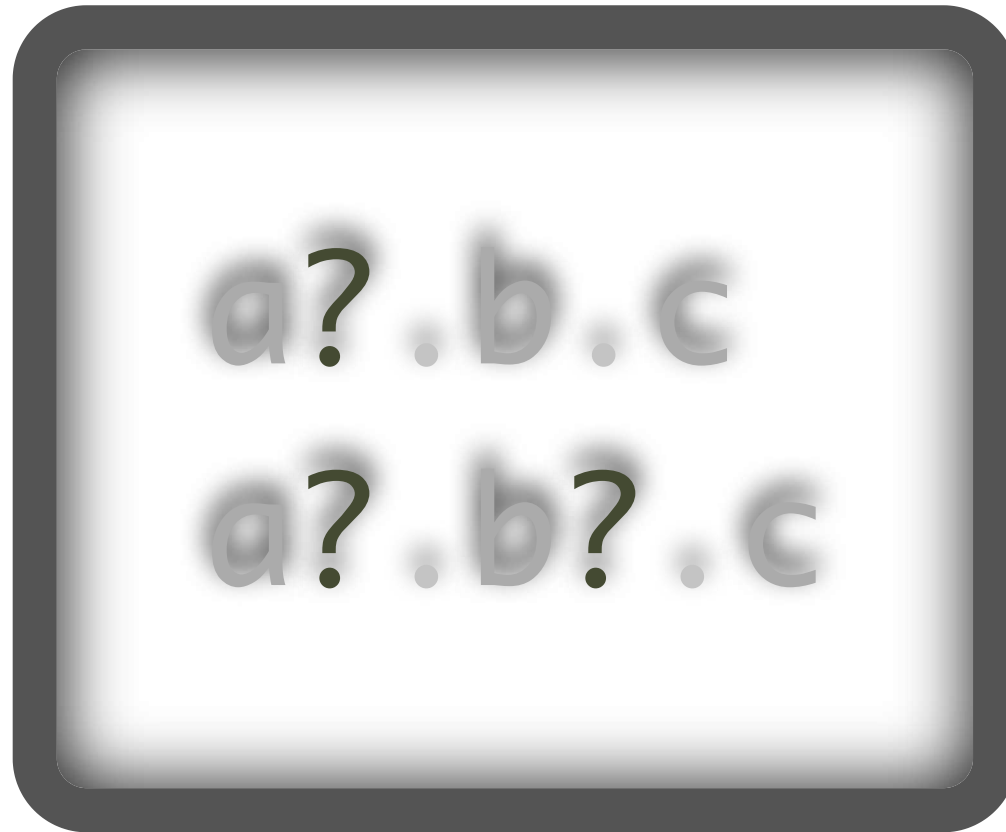
TypeScript

<https://johnhax.net/2019/js-new-features/slide?qcon#87>

a? .b

a 必须是 nullish

<https://johnhax.net/2019/js-new-features/slide?qcon#88>



<https://johnhax.net/2019/js-new-features/slide?qcon#89>

Short-circuiting

<https://johnhax.net/2019/js-new-features/slide?qcon#90>

a?.[+++x]

<https://johnhax.net/2019/js-new-features/slide?qcon#91>

`a == null ? undefined : a[++x]`

<https://johnhax.net/2019/js-new-features/slide?qcon#92>

`a?.b.c(++x).d`

`a == null ? undefined : a.b.c(++x).d`

<https://johnhax.net/2019/js-new-features/slide?qcon#93>

Long short-circuiting

<https://johnhax.net/2019/js-new-features/slide?qcon#94>



`a?.b.c`

`(a?.b).c`

<https://johnhax.net/2019/js-new-features/slide?qcon#95>


```
a == null ? undefined : a.b.c  
(a == null ? undefined : a.b).c
```

<https://johnhax.net/2019/js-new-features/slide?qcon#96>



a?.b.c.d

a?.b?.c?.d

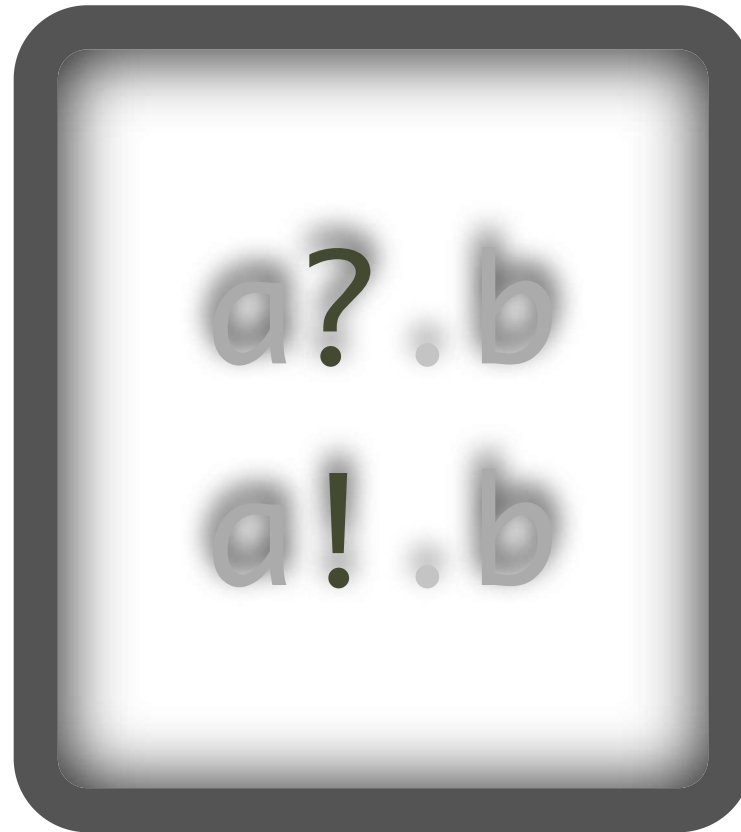
<https://johnhax.net/2019/js-new-features/slide?qcon#97>

TypeScript静态类型检查
可以严格地确定每个部分
应该用`a.b`还是`a?.b`

<https://johnhax.net/2019/js-new-features/slide?qcon#98>

a? .b .c? .d

<https://johnhax.net/2019/js-new-features/slide?qcon#99>



<https://johnhax.net/2019/js-new-features/slide?qcon#100>

Non-null assertion

<https://johnhax.net/2019/js-new-features/slide?qcon#101>

a!

<https://johnhax.net/2019/js-new-features/slide?qcon#102>

```
const H = '🇭🇰'.codePointAt(0).toString(16)
const K = '🇭🇰'.codePointAt(2).toString(16)
console.log(H, K)
```

<https://johnhax.net/2019/js-new-features/slide?qcon#103>


```
const H = '🍀'.codePointAt(0)!.toString(16)
const K = '🍀'.codePointAt(2)!.toString(16)
console.log(H, K)
```

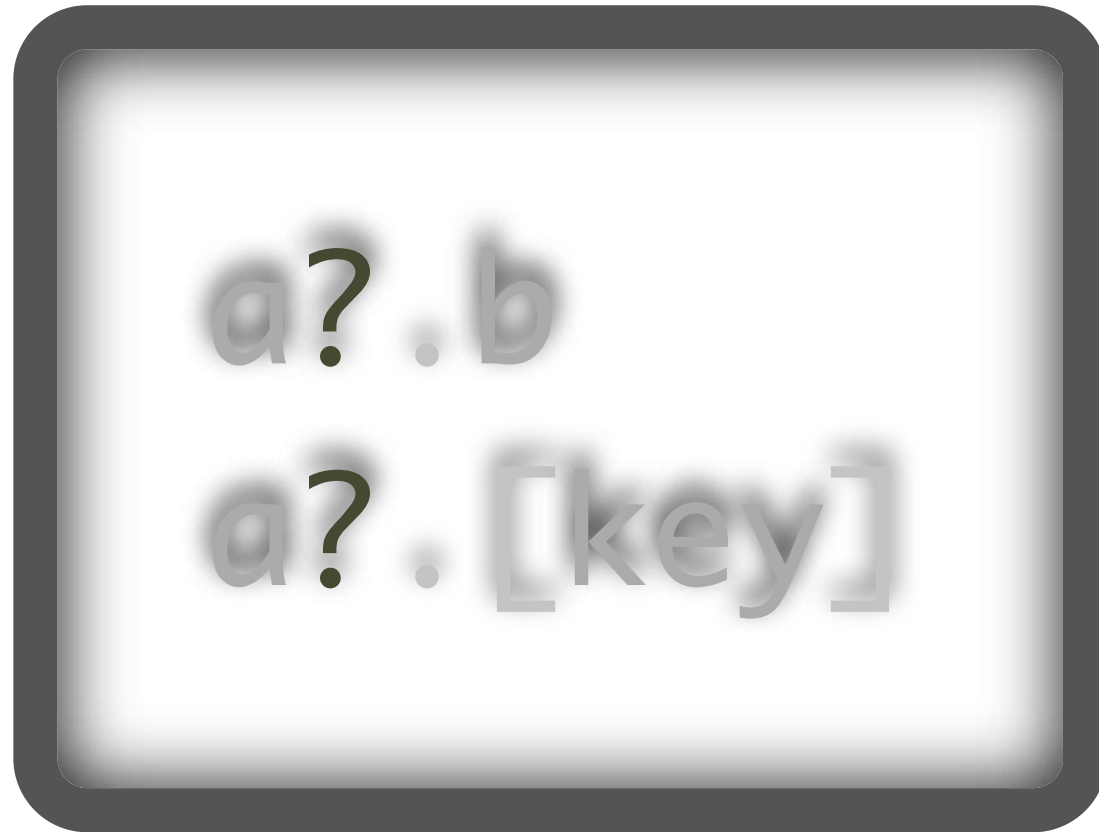
<https://johnhax.net/2019/js-new-features/slide?qcon#104>

```
const H = '🇰🇵'.codePointAt(0)?.toString(16)
const K = '🇰🇵'.codePointAt(2)?.toString(16)
console.log(H, K)
```

<https://johnhax.net/2019/js-new-features/slide?qcon#105>

```
const lang = localStorage.getItem('lang') ?? 'en-US'  
const labels = i18n.load(lang)  
const user = JSON.parse(localStorage.getItem('user'))  
document.querySelector('bdi#user-name')!.textContent =  
  user?.name ?? labels?['non-login-user-name'] ?? 'anonymous'
```

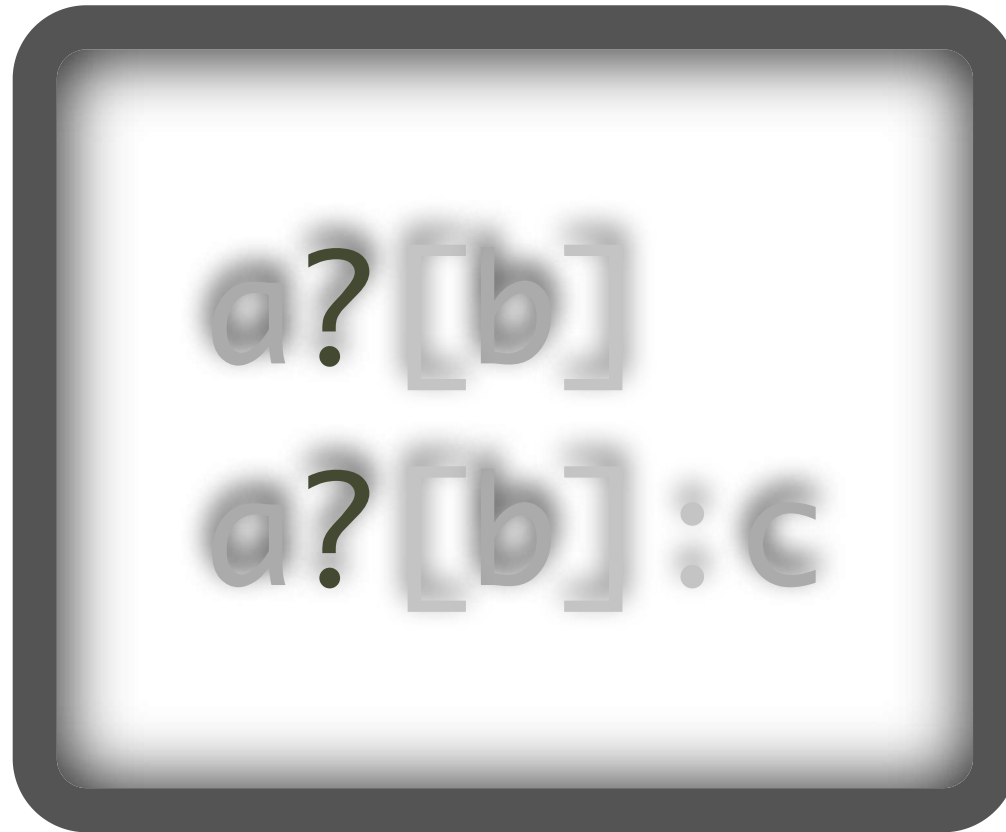
<https://johnhax.net/2019/js-new-features/slide?qcon#106>



<https://johnhax.net/2019/js-new-features/slide?qcon#107>

$a . b \rightarrow a [\text{key}]$
 $a ? . b \rightarrow a ? [\text{key}]$

<https://johnhax.net/2019/js-new-features/slide?qcon#108>



<https://johnhax.net/2019/js-new-features/slide?qcon#109>

`a?[b]?[c]:d`

<https://johnhax.net/2019/js-new-features/slide?qcon#110>

其他语言?

<https://johnhax.net/2019/js-new-features/slide?qcon#111>

C#: 数组是 {...}

<https://johnhax.net/2019/js-new-features/slide?qcon#112>

CoffeeScript: 没有?: 三元

<https://johnhax.net/2019/js-new-features/slide?qcon#113>

Swift: 用空格区分!

<https://johnhax.net/2019/js-new-features/slide?qcon#114>

try a? .b? [c]

<https://johnhax.net/2019/js-new-features/slide?qcon#115>



`a?.b`

`a?.[prop]`

`a?.(...args)`

<https://johnhax.net/2019/js-new-features/slide?qcon#116>

`a !== null ? a(...args) : undefined`

<https://johnhax.net/2019/js-new-features/slide?qcon#117>

a?({...})

a?.({...})

<https://johnhax.net/2019/js-new-features/slide?qcon#118>

C#: `a?.invoke(...)`

<https://johnhax.net/2019/js-new-features/slide?qcon#119>

- `a.b?.(...)`
- `a.b?.invoke(...)`

<https://johnhax.net/2019/js-new-features/slide?qcon#120>

Stage 2

<https://johnhax.net/2019/js-new-features/slide?qcon#121>



```
a?? .b
```

```
a??[key]
```

```
a??(...args)
```

<https://johnhax.net/2019/js-new-features/slide?qcon#122>

`a?? . b`

`a?? [key]`

`a?? (...args)`

`a ??? b`

<https://johnhax.net/2019/js-new-features/slide?qcon#123>

```

a?? .b a??[key] a??(...args) a ??? b
a?&b a?&[key] a?&(...args) a ?? b
a?>b a?>[key] a?>(...args) a ?? b
a?.b a?\[key] a?\(...args) a ?? b

a\.b a\[key] a\(...args) a \\ b
a?.b a?.[key] a?.(...args) a ?? b

```

<https://johnhax.net/2019/js-new-features/slide?qcon#124>

主要use case

a?.b 80%+

a ?? b

<https://johnhax.net/2019/js-new-features/slide?qcon#125>

delete a?.b

<https://johnhax.net/2019/js-new-features/slide?qcon#126>

```
a == null ? true : delete a.b
```

<https://johnhax.net/2019/js-new-features/slide?qcon#127>

不支持的

<https://johnhax.net/2019/js-new-features/slide?qcon#128>

- `new?.target`
- `import?.('foo')`

- `new a?.()`
- `new a?.b()`
- `super?.()`

- `super?.foo`
- `a?.string`
- `a?.bstring`

- `a?.b = c`

<https://johnhax.net/2019/js-new-features/slide?qcon#129>

a?.b = c

<https://johnhax.net/2019/js-new-features/slide?qcon#130>

```
// syntax error  
document.querySelector('#test')?.innerHTML += '...'
```

<https://johnhax.net/2019/js-new-features/slide?qcon#131>

```
// ok  
document.querySelector('#test')?.insertAdjacentHTML('beforeend', '...')
```

<https://johnhax.net/2019/js-new-features/slide?qcon#132>

#18

<https://johnhax.net/2019/js-new-features/slide?qcon#133>

QA

<https://johnhax.net/2019/js-new-features/slide?qcon#134>

`String.prototype.replaceAll`

<https://johnhax.net/2019/js-new-features/slide?qcon#135>


```
'abaa'.replace('a', 'A') // Abaa  
'abaa'.replace(/a/, 'A') // Abaa  
'abaa'.replace(/a/g, 'A') // AbAA
```

```
'abaa'.replaceAll('a', 'A') // AbAA  
'abaa'.replaceAll(/a/, 'A') // Ab??  
'abaa'.replaceAll(/a/g, 'A') // AbAA
```

<https://johnhax.net/2019/js-new-features/slide?qcon#136>

- 'AbAA' (auto g)
- 'Abaa' (consistency?)
- throw `TypeError`
- do not allow `RegExp` at all (change API name to `substitute`)

<https://johnhax.net/2019/js-new-features/slide?qcon#137>

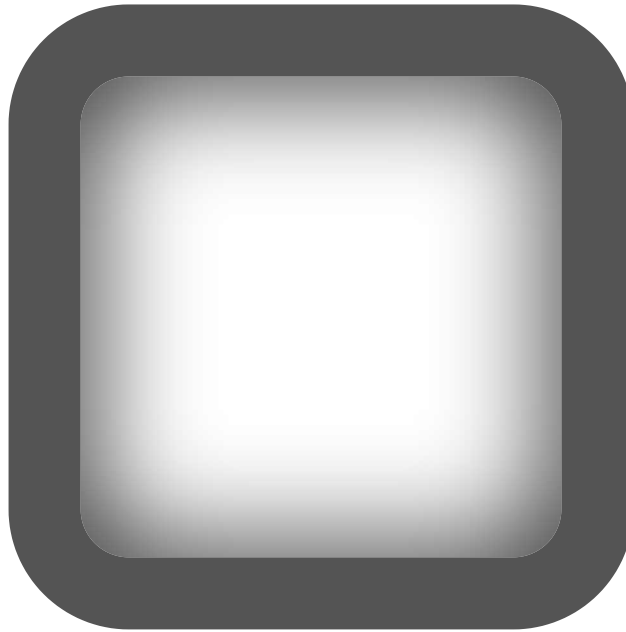
String.prototype.matchAll

<https://johnhax.net/2019/js-new-features/slide?qcon#138>

```
'abaa'.match('a') // ['a', index: 0, input: 'abaa']  
'abaa'.match(/a/) // ['a', index: 0, input: 'abaa']  
'abaa'.match(/a/g) // ['a', 'a', 'a']
```

<https://johnhax.net/2019/js-new-features/slide?qcon#139>

```
'abaa'.matchAll('a') // [['a', index: 0, input: 'abaa']]  
'abaa'.matchAll(/a/) // ?  
'abaa'.matchAll(/a/g) // [['a', index: 0], ['a', index: 2], ['a', index: 3]]
```



<https://johnhax.net/2019/js-new-features/slide?qcon#140>

Stage 4!

<https://johnhax.net/2019/js-new-features/slide?qcon#141>

```
'abaa'.replaceAll(/a/, 'A') // TypeError!  
'abaa'.matchAll(/a/)       // TypeError!
```

<https://johnhax.net/2019/js-new-features/slide?qcon#142>

globalThis

<https://johnhax.net/2019/js-new-features/slide?qcon#143>

- window
- self
- global

<https://johnhax.net/2019/js-new-features/slide?qcon#144>

Why globalThis

<https://johnhax.net/2019/js-new-features/slide?qcon#145>

global

<https://github.com/tc39/proposal-global>

<https://johnhax.net/2019/js-new-features/slide?qcon#146>

```
if (typeof global === 'object') {  
  // code for node.js  
} else {  
  // code for browser  
}
```

<https://johnhax.net/2019/js-new-features/slide?qcon#147>

- self
- globalObject
- globe
- Global

<https://johnhax.net/2019/js-new-features/slide?qcon#148>

globalThis

<https://johnhax.net/2019/js-new-features/slide?qcon#149>

安慰一下自己

<https://johnhax.net/2019/js-new-features/slide?qcon#150>

只是一个
名字而已

<https://johnhax.net/2019/js-new-features/slide?qcon#151>

QA

<https://johnhax.net/2019/js-new-features/slide?qcon#152>

One more thing

<https://johnhax.net/2019/js-new-features/slide?qcon#153>

class fields

<https://johnhax.net/2019/js-new-features/slide?qcon#154>

Stage 3

<https://johnhax.net/2019/js-new-features/slide?qcon#155>

Chrome已经实现

<https://johnhax.net/2019/js-new-features/slide?qcon#156>

```
class Test {  
  name = 'hax'  
  greeeting = `Hello ${name}`  
}  
console.log(new Test().greeting)
```

<https://johnhax.net/2019/js-new-features/slide?qcon#157>

```
class Test {  
  #privateField  
  #privateMethod() {}  
  async #privateAsyncMethod() {}  
  get #privateGetter() {}  
  *#privateGenerator() {}  
  async *#privateAsyncGenerator() {}  
}
```

<https://johnhax.net/2019/js-new-features/slide?qcon#158>

```
class Test {  
  #privateField  
  #privateMethod() {}  
  async #privateAsyncMethod() {}  
  get #privateGetter() {}  
  *#privateGenerator() {}  
  async *#privateStaticAsyncGenerator() {}  
  static #privateStaticField  
  
  static #privateStaticMethod() {}  
  static async #privateStaticAsyncMethod() {}  
  static get #privateStaticGetter() {}  
  
  static *#privateStaticGenerator() {}  
  static async *#privateStaticAsyncGenerator() {}  
}
```

<https://johnhax.net/2019/js-new-features/slide?qcon#159>

9月8日class fields提案闭门研讨会

<https://johnhax.net/2019/js-new-features/slide?qcon#160>

参会名单

<https://johnhax.net/2019/js-new-features/slide?qcon#161>

