JavaScript 的困境与挑战

GMTC 2019 全球大前端技术大会·深圳站

https://johnhax.net/2020/js-challenges/slide#0

GitHub @hax

知乎 @贺师俊

微博 @johnhax

Twitter @haxy

https://johnhax.net/2020/js-challenges/slide#1

360高级前端架构师

360技术委员会Web前端TC委员

https://johnhax.net/2020/js-challenges/slide#2

2019年7月起
TC39代表

https://johnhax.net/2020/js-challenges/slide#3

1998

https://johnhax.net/2020/js-challenges/slide#4

JavaScript

https://johnhax.net/2020/js-challenges/slide#5

IE 4

https://johnhax.net/2020/js-challenges/slide#6

- 前ES3时代 *1995~1999*
- ES3时代 *2000~2010*

- Harmony时代 *2008~2016*
- ES6/Babel时代 *2014~2020*
- TS/JS生态新时代 *2019~*

https://johnhax.net/2020/js-challenges/slide#7

2011：ES5 — Improve the Safety of JavaScript

https://johnhax.net/2020/js-challenges/slide#8

- API扩展和标准化：JSON、Array.protoype.forEach/map/filter...
- 通用化，可实现平台对象：get/set accessor、Object.defineProperty、etc.

- 适应于PITL（programming-in-the-large）

https://johnhax.net/2020/js-challenges/slide#9

2014：透过ES6看JS未来

https://johnhax.net/2020/js-challenges/slide#10

- Module 使得 JS 生态系统能重新统一
- Module / Class 等让 JS 更适合大型编程

- Promise 等将最佳实践标准化
- Generator / Proxy 等提供了现代语言所需的重要设施
- Arrow function / Destructring 等不仅是语法糖，而且填了长期以来的一些坑

https://johnhax.net/2020/js-challenges/slide#11

2015~2017：JS — 世界第一程式設計語言

https://johnhax.net/2020/js-challenges/slide#12

Babel makes the language

Modular

https://johnhax.net/2020/js-challenges/slide#13

22% stage 4

20% stage 3

25% stage 2

33% stage 0/1

https://johnhax.net/2020/js-challenges/slide#14

Enable language features

incrementally

https://johnhax.net/2020/js-challenges/slide#15

Hardest to upgrade

–> Always use

latest feature!

https://johnhax.net/2020/js-challenges/slide#16

Ecosystem

https://johnhax.net/2020/js-challenges/slide#17

JavaScript have:

https://johnhax.net/2020/js-challenges/slide#18

The most **ubiquitous** platform:

Browser

https://johnhax.net/2020/js-challenges/slide#19

The most **active** platform:

Node.js

https://johnhax.net/2020/js-challenges/slide#20

The **largest** companies:

Google Microsoft

Apple Facebook

https://johnhax.net/2020/js-challenges/slide#21

The most experienced experts
in the language committee
---------- & ----------
The largest developers

Community

https://johnhax.net/2020/js-challenges/slide#22

立了 Flag

https://johnhax.net/2020/js-challenges/slide#23

隐忧浮现

https://johnhax.net/2020/js-challenges/slide#24

JS fatigue

https://johnhax.net/2020/js-challenges/slide#25

学不动了

https://johnhax.net/2020/js-challenges/slide#26

边际效用

性价比

https://johnhax.net/2020/js-challenges/slide#27

2020：JavaScript 的困境与挑战

https://johnhax.net/2020/js-challenges/slide#28

前端开发编程语言的过去、现在和未来

GMTC 2019 全球大前端技术大会·北京站

https://johnhax.net/2020/js-challenges/slide#29

TS/JS未来面临的挑战

https://johnhax.net/2020/js-challenges/slide#30

TS背着JS包袱

JS背着历史包袱

https://johnhax.net/2020/js-challenges/slide#31

社区复杂、差异性
怎么做tradeoff?
委员会语言弊病?

https://johnhax.net/2020/js-challenges/slide#32

历史包袱

https://johnhax.net/2020/js-challenges/slide#33

不能破坏
兼容性

https://johnhax.net/2020/js-challenges/slide#34

只能通过增加特性
来解决以前的问题

https://johnhax.net/2020/js-challenges/slide#35

Arrow Function

https://johnhax.net/2020/js-challenges/slide#36

lexical `this`

https://johnhax.net/2020/js-challenges/slide#37

不可避免地使得

`this`语义更加复杂

https://johnhax.net/2020/js-challenges/slide#38

- 工程方案
- 事实标准

- 真·标准

https://johnhax.net/2020/js-challenges/slide#39

长期共存

https://johnhax.net/2020/js-challenges/slide#40

- CommonJS
- ES module

- Bundle

https://johnhax.net/2020/js-challenges/slide#41

CJS/ESM互操作性问题

https://johnhax.net/2020/js-challenges/slide#42

```
--experimental-modules
```

https://johnhax.net/2020/js-challenges/slide#43

包入口可不可以

分别指定CJS/ESM?

https://johnhax.net/2020/js-challenges/slide#44

```
<script></script>
<script type=module></script>
<script type=module async></script>
```

https://johnhax.net/2020/js-challenges/slide#45

Polyfill+Transpiler

https://johnhax.net/2020/js-challenges/slide#46

Polyfill

https://johnhax.net/2020/js-challenges/slide#47

狭义Polyfill

广义Polyfill

https://johnhax.net/2020/js-challenges/slide#48

- Global
- Prototype

https://johnhax.net/2020/js-challenges/slide#49

- Array.prototype.flatten (MooTools)
- Array.prototype.flat (HighChart)

https://johnhax.net/2020/js-challenges/slide#50

- Polyfill
- Experimental implementation

https://johnhax.net/2020/js-challenges/slide#51

Babel

https://johnhax.net/2020/js-challenges/slide#52

Stage-x Preset

https://johnhax.net/2020/js-challenges/slide#53

22% stage 4

20% stage 3

25% stage 2

33% stage 0/1

https://johnhax.net/2020/js-challenges/slide#54

无意中在Production中
使用了unstable features

https://johnhax.net/2020/js-challenges/slide#55

JavaScript

BabelScript

https://johnhax.net/2020/js-challenges/slide#56

babel-plugin-macros

https://johnhax.net/2020/js-challenges/slide#57

长得像函数
其实并不是

https://johnhax.net/2020/js-challenges/slide#58

TypeScript

https://johnhax.net/2020/js-challenges/slide#59

Stage 3

https://johnhax.net/2020/js-challenges/slide#60

2012

https://johnhax.net/2020/js-challenges/slide#61

- arrow function
- class

- es module
- decorator
- private property / private field

- public property / public field

https://johnhax.net/2020/js-challenges/slide#62

ESLint

https://johnhax.net/2020/js-challenges/slide#63

Stage 4

https://johnhax.net/2020/js-challenges/slide#64

- 早期stage也/更需要保护
- 标准缺乏lint社区的反馈

https://johnhax.net/2020/js-challenges/slide#65

Maximally Minimal

https://johnhax.net/2020/js-challenges/slide#66

先解決温饱

再考虑小康

https://johnhax.net/2020/js-challenges/slide#67

每个人的需求不一样

谁能代表开发者?

https://johnhax.net/2020/js-challenges/slide#68

`Map.prototype.upsert()`

https://johnhax.net/2020/js-challenges/slide#69

如何衡量成本和收益?

https://johnhax.net/2020/js-challenges/slide#70

有争议的提案

https://johnhax.net/2020/js-challenges/slide#71

Top-level Await

Class fields

https://johnhax.net/2020/js-challenges/slide#72

『改革进入深水区』

https://johnhax.net/2020/js-challenges/slide#73

好做的都已做完了
剩下的都是难搞的

https://johnhax.net/2020/js-challenges/slide#74

Pipeline Operator

https://johnhax.net/2020/js-challenges/slide#75

```
value |> f
```

```
f(value)
```

https://johnhax.net/2020/js-challenges/slide#76

```
value |> f(...args)
```

https://johnhax.net/2020/js-challenges/slide#77

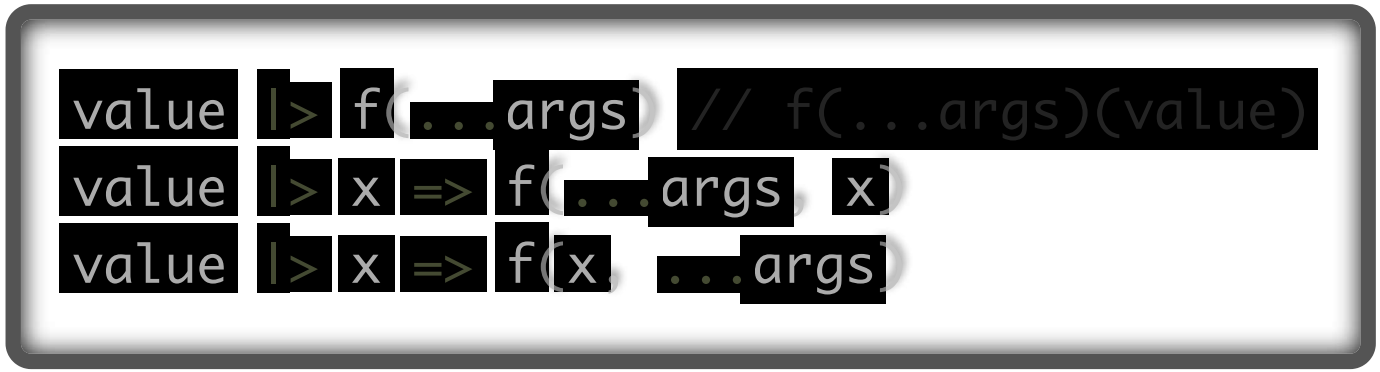
- `f(...args)(value)`
- `f(...args, value)`

- `f(value, ...args)`
- `f.call(value, ...args)`

https://johnhax.net/2020/js-challenges/slide#78

F# Style

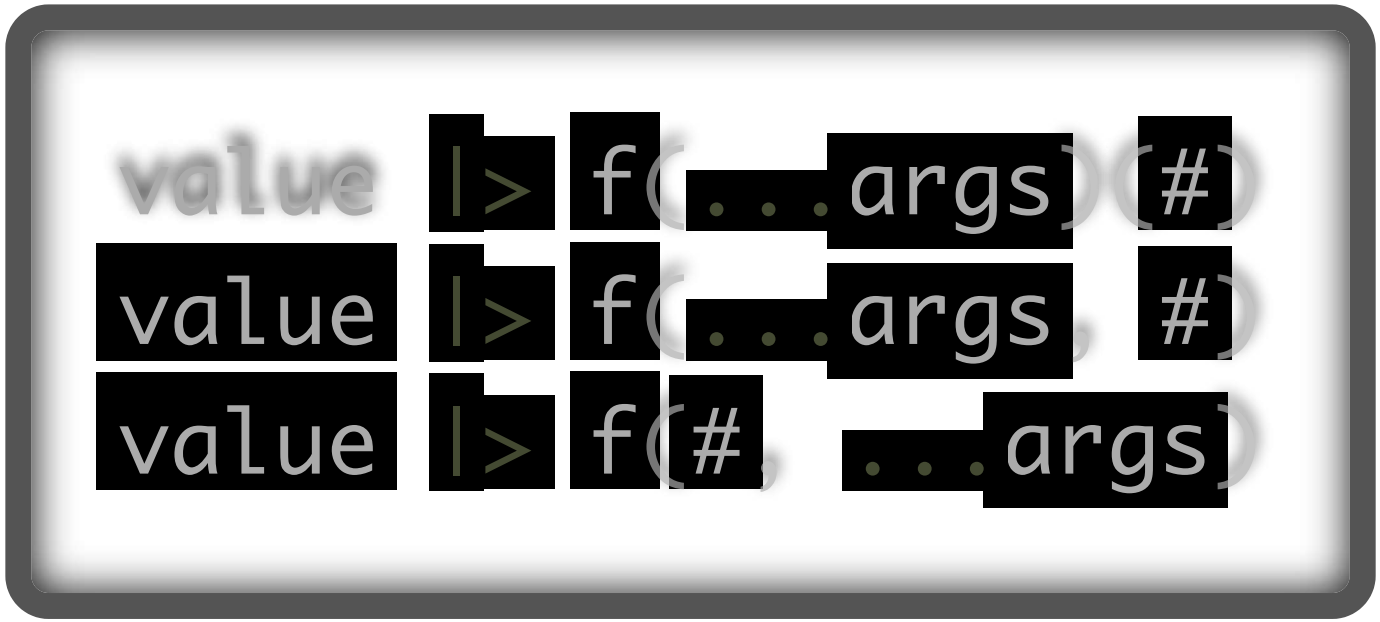
https://johnhax.net/2020/js-challenges/slide#79



https://johnhax.net/2020/js-challenges/slide#80

Smart Style

https://johnhax.net/2020/js-challenges/slide#81



https://johnhax.net/2020/js-challenges/slide#82

- 更符合FP主流?
- 更普适JS?

https://johnhax.net/2020/js-challenges/slide#83

Binary AST

https://johnhax.net/2020/js-challenges/slide#84

- 不同平台的需求
- 性能和动态性的矛盾

https://johnhax.net/2020/js-challenges/slide#85

Roadmap?

https://johnhax.net/2020/js-challenges/slide#86

主导者?

全局观?

https://johnhax.net/2020/js-challenges/slide#87

总结

https://johnhax.net/2020/js-challenges/slide#88

JavaScript 的困境与挑战

https://johnhax.net/2020/js-challenges/slide#89

QA

https://johnhax.net/2020/js-challenges/slide#90