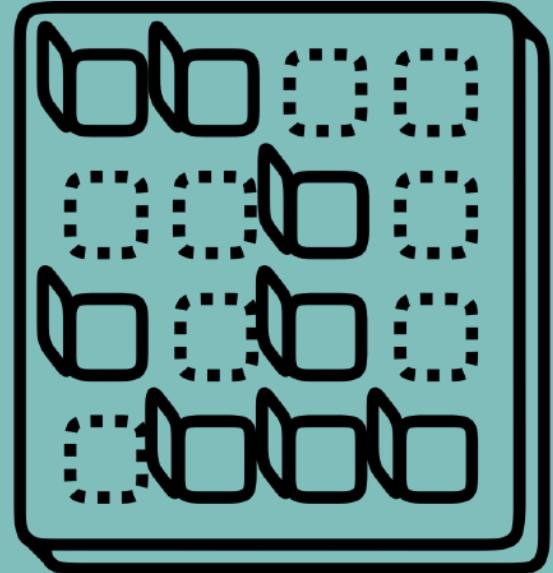


24 features of JavaScript
that made progress in 2019

Advent.js



Szabolcs
Szabolcsi-
Toth

@_Nec

Senior
Engineer at
IBM Cloud



JSConf &
CSSConf
Budapest

Landed in **ES2019**

1	2	
3	4	5

Object fromEntries

1



Object fromEntries



```
const entries = [
  ["foo", "bar"],
  ["baz", 123]
]

const obj = Object.fromEntries(entries)

/*
{
  foo: "bar",
  baz: 123
}
*/
```

String **trimLeft / trimRight**

2



String trimLeft / trimRight



```
const str = "    random text    "

str.trimLeft() // "random text    "
str.trimStart() // "random text    "

str.trimRight() // "    random text"
str.trimEnd()   // "    random text"
```

Array **flatMap**

3



Array flat / flatMap



```
const nested = ['a', 'b', ['c', 'd']]  
nested.flat()  
// ['a', 'b', 'c', 'd']
```

```
const deepNested = [1, [2, [3]]];  
deepNested.flat();  
// [1, 2, [3]]
```

```
deepNested.flat(Infinity);  
// → [1, 2, 3]
```

Optional catch binding

4



Optional catch binding



```
try {
    someActionThrows()
} catch(e) {
    notUsingError() // `e` is unnecessary, unused
}
```

```
try {
    someActionThrows()
} catch {
    notUsingError()
}
```

Stable Array sort

5



Stable Array **sort**



```
const starWars = [
  { title: "The Phantom Menace", rating: 6.5 },
  { title: "Attack of the Clones", rating: 6.5 },
  { title: "Revenge of the Sith", rating: 7.5 },
  { title: "A new hope", rating: 8.5 },
  { title: "The Empire Strikes Back", rating: 8.7 },
  { title: "Return of the Jedi", rating: 8.3 },
]

starWars.sort((a, b) => b.rating - a.rating)

// { title: "The Phantom Menace", rating: 6.5 },
// { title: "Attack of the Clones", rating: 6.5 },

// { title: "Attack of the Clones", rating: 6.5 },
// { title: "The Phantom Menace", rating: 6.5 },
// :(
```

Landed in
ES2019

Made to
Stage 4

Will be released in
ES2020

1	2	
3	4	5

6	7	
8	9	10

Promise allSettled

6



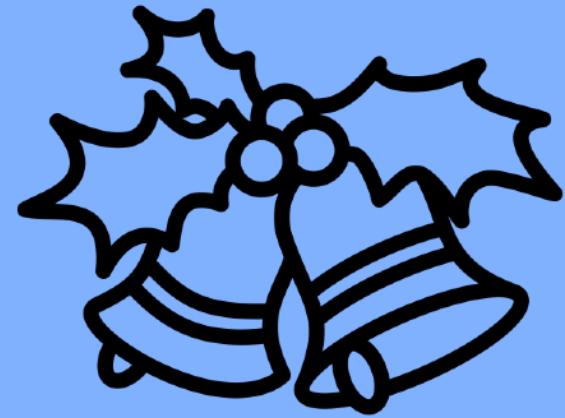
Promise **allSettled**



```
Promise.allSettled([
  Promise.resolve(),
  Promise.reject(),
  fetch('/api/endpoint')
])
  .then((results) => {}) // results.length === 3
  .catch(() => {}) // never gets rejected !!
```

globalThis

7



globalThis



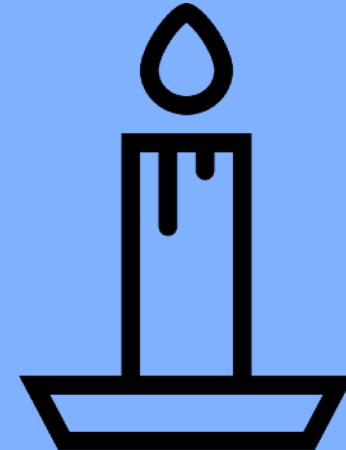
```
// browser environment
console.log(globalThis);    // => Window {...}

// node.js environment
console.log(globalThis);    // => Object [global] {...}

// web worker environment
console.log(globalThis);    // => DedicatedWorkerGlobalScope {...}
```

BigInt

8



BigInt



```
const biig = 9007199254740993 // 253 + 1
// 9007199254740992 <--- !?!?!?!

const huge = BigInt('9007199254740993'); // 253 + 1
// 9007199254740993n

// + - * % ** works the same
4n + 2n      // 6n
4n - 2n      // 2n
4n * 2n      // 8n
4n % 2n      // 0n
4n ** 2n     // 16n

// division: no fraction digits!
2n / 4n      // 0n not 0.5 !!
```

BigInt



```
// loosely equal to Number
2n == 2      // true
2n === 2     // false

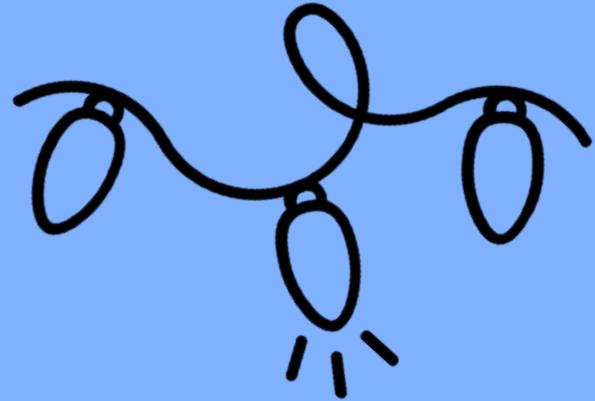
// comparison and boolean conversion is the same
// < > >= <= !

// Can't mix with Number !
2n + 2 // TypeError
2n * 2 // TypeError

// But can be converted to String 😊
4n + '2' // "42"
```

Optional Chaining

9



Optional Chaining



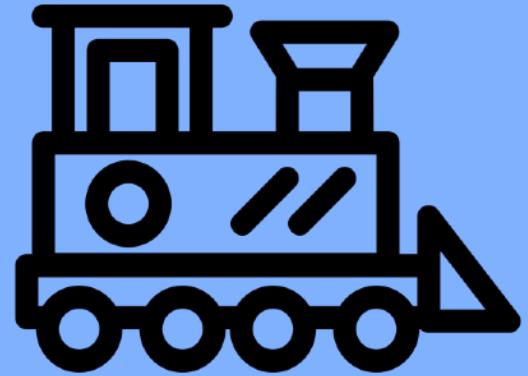
```
// before
const phone = user.contact && user.contact.phone

// after
const phone = user.contact?.phone

// syntax
obj?.prop      // optional static property access
obj?.[expr]    // optional dynamic property access
func?(...args) // optional function or method call
```

Nullish Coalescing

10



Nullish Coalescing



```
// explicit check for `null` and `undefined`

null && 0 // null
null || 0 // 0
null ?? 0 // 0

// 'falsy' values _are_ values

0 || 'fallback value' // "fallback value"
0 ?? 'fallback value' // 0
undefined ?? 'fallback value' // "fallback value"
```

Landed in ES2019

1	2	
3	4	5

Made to Stage 4

Will be released in
ES2020

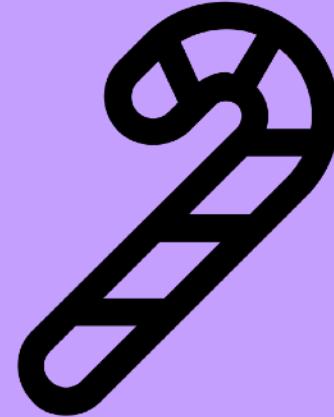
6	7	
8	9	10

Made to Stage 3

11		
12	13	14
15	16	17

Class Field declarations

11



Class Field declarations



```
class MyObject {  
  
    data = 23 // initial data  
  
    increment () {  
        this.data += 1  
    }  
  
    someMethod () {  
        this.increment()  
        return this.data  
    }  
}
```

Class Private methods & fields

12



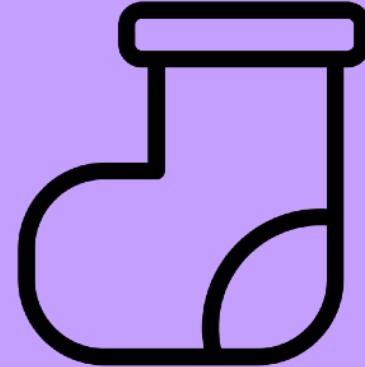
Class Private methods & fields



```
class MyObject {  
  
    #internalData = 23  
  
    #increment () {  
        this.#internalData += 1  
    }  
  
    someMethod () {  
        this.#increment()  
        return this.#internalData  
    }  
}
```

Class Static features

13



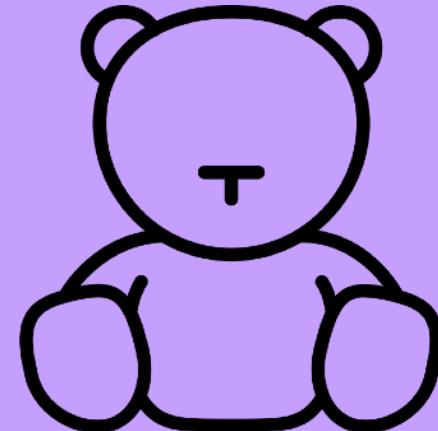
Class Static features



```
1 class VAT {  
2  
3     static version = '1.1.2';      // static field  
4  
5     static #ratio = 1.27;          // private static field  
6  
7     static #calculate (num) {      // private static method  
8         return num * VAT.#ratio   // use private static field  
9     }  
10  
11    gross (num) {  
12        return VAT.#calculate(num) // call private static method  
13    }  
14 }  
15  
16  
17
```

Numeric Separators

14



Numeric Separators



```
const FEE = 12300
// 12,300 EUR?
// 12300 cents so, 123.00 EUR?
```

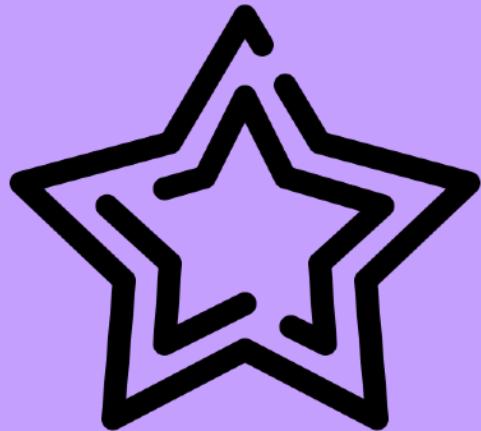
```
const BUDGET = 1234500
// 1,234,500 EUR?
// 1234500 cents so 12,345.00 EUR?
// 123,45 in 4-fixed financial?
```

```
12_300 // 12,300 EUR
123_00 // 123.00 EUR
```

```
1_234_500    // 1,234,500 EUR
12345_00     // means cents, so 12,345.00 EUR
```

Top-level await

15



Top-level `await`



// before: Async IIFE

```
(async () => {
    const data = await fetch(....)
})()
```

// now, you can do these in ECMAScript Modules (ESM)

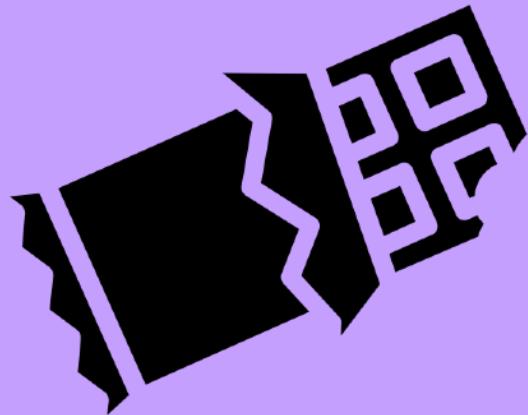
```
const data = await fetch(...)
```

// async dynamic imports & computed modules

```
const translations = await import(`./i18n/${navigator.language}`);
```

String replaceAll

16



String `replaceAll`



```
const queryString = "santa=ho+ho+ho"
```

```
// replace all `+` to ``  
queryString.replace(/\+/g, '')
```

```
// proposal  
queryString.replaceAll('+', '')
```

```
// using a non-global regexp will throw an exception!  
queryString.replaceAll(/\+/g, '')
```

Promise any

17



Promise any



```
// stops when an input promise is fulfilled
Promise.any([
  fetch(url),
  fetch(backupUrl),
])
  .then((data) => {}) // fulfills if one of the inputs is fulfilled
  .catch((e) => {}) // rejects if _all_ inputs are rejected
```

Landed in ES2019

1	2	
3	4	5

Made to Stage 4

Will be released in
ES2020

6	7	
8	9	10

Made to Stage 3

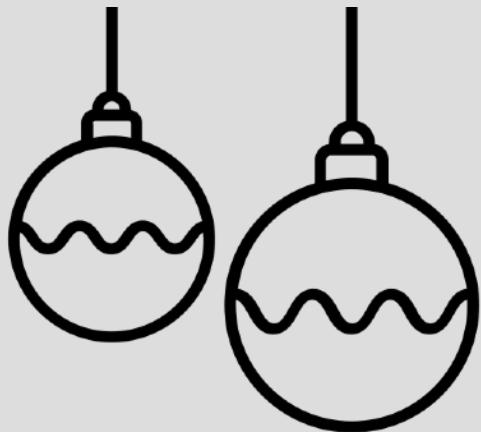
11		
12	13	14
15	16	17

Made to Stage 2

18		
19	20	21
22	23	24

Decorators

18



Decorators



```
// internal decorators

@wrap
@register
@initialize
@expose

// user created decorators

decorator @customDecorator(...args) {
    // decorator logic
}
```

Decorators



```
// internal decorator
@wrap

class C {
  @wrap(f) method() { }
}

// same as

class C {
  method() { }
}
C.prototype.method = f(C.prototype.method);
```

Decorators



```
// internal decorator
@register

class C {
    @register(f) method() { }
}
```

// same as

```
class C {
    method() { }
}
f(C, "method");
```

Decorators



```
// internal decorator  
@initialize
```

```
@initialize(f)  
class C { }
```

// is the same as

```
class C {  
    constructor() {  
        f(this);  
    }  
}
```

Decorators



```
// internal decorator
@expose

class C {
    @expose(f) #x;
}
```

Decorators



```
// user created decorators

export decorator @customDecorator(...args) {
    // decorator logic
}

// usage

import { @customDecorator } from 'customDecorator.mjs'

// decorate Class
@customDecorator
class C {
    method() { }
}

// decorate method
class C {
    @customDecorator method() { }
}
```

New methods for Set

19



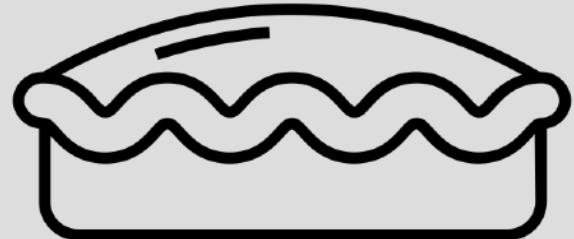
New methods for Set



```
Set.prototype.intersection(iterable)
// method creates new Set instance by set intersection operation.
Set.prototype.union(iterable)
// method creates new Set instance by set union operation.
Set.prototype.difference(iterable)
// method creates new Set without elements present in iterable.
Set.prototype.symmetricDifference(iterable)
// returns Set of elements found only in either this or in iterable.
Set.prototype.isSubsetOf(iterable)
// determines whether a set is a subset of a specified iterable.
Set.prototype.isDisjointFrom(iterable)
// determines whether the set has no members in common with the specified iterable.
Set.prototype.isSupersetOf(iterable)
// determines whether a set is a superset of a specified iterable.
```

Realms

20



Realms

20

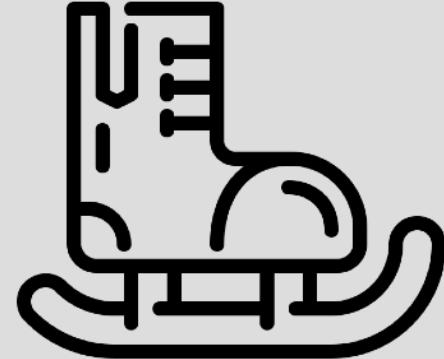
HTML ➡ IFRAME

JS ➡ Realms

- security isolation
- plugins
- in-browser code editors
- server-side rendering
- testing/mocking (e.g., jsdom)
- in-browser transpilation

ArrayBuffer transfer

21



ArrayBuffer transfer



```
1 // working with large ArrayBuffer might be slow
2 await validate(arrayBuffer);
3 // ..
4 await fs.writeFile("data.bin", arrayBuffer);
5
6 // possible solution: copy (but it eats up memory)
7 const copy = arrayBuffer.slice();
8 await validate(copy);
9 await fs.writeFile("data.bin", copy);
10
11 // proposed solution: transfer()
12 const transferred = arrayBuffer.transfer();
13 // `transferred` points the same memory space as `arrayBuffer`
14 // but `arrayBuffer` is "detached" from it's data, cannot be modified
15 await validate(transferred);
16 await fs.writeFile("data.bin", transferred);
17
18 // note: SharedArrayBuffer cannot be detached!
19
```

Temporal

22



Temporal

22

Top-level namespace for
Date & Time related features
(like Math)

“Fixing JavaScript Date”

Iterator helpers

23



Iterator helpers



```
1 // provide all iterators (async & not) these lazy methods
2 map( fn )
3 filter( fn )
4 take( limit )
5 drop( limit )
6 asIndexedPairs( )
7 flatMap( fn )
8 reduce( reducer [ , initialValue ] )
9 toArray( )
10 forEach( fn )
11 some( fn )
12 every( fn )
13 find( fn )
14
15
16 function* naturals() {
17     //...
18 }
19
20 const evens = naturals()
21     .filter((n) => n % 2 === 0);
22
23
```

Map upsert

24



Map upsert



```
1 // provide all iterators (async & not) these lazy methods
2 map( fn )
3 filter( fn )
4 take( limit )
5 drop( limit )
6 asIndexedPairs( )
7 flatMap( fn )
8 reduce( reducer [ , initialValue ] )
9 toArray( )
10 forEach( fn )
11 some( fn )
12 every( fn )
13 find( fn )
14
15
16 function* naturals() {
17     //...
18 }
19
20 const evens = naturals()
21     .filter((n) => n % 2 === 0);
22
23
```



Made to ES2019

- <https://github.com/tc39/proposal-object-from-entries>
- <https://github.com/tc39/proposal-string-left-right-trim>
- <https://github.com/tc39/proposal-flatMap>
- <https://github.com/tc39/proposal-optional-catch-binding>
- <https://github.com/tc39/ecma262/pull/1340>

Made to Stage 4, will be ES2020

- <https://github.com/tc39/proposal-promise-allSettled>
- <https://github.com/tc39/proposal-global>
- <https://github.com/tc39/proposal-bigint>
- <https://github.com/tc39/proposal-optional-chaining>
- <https://github.com/tc39/proposal-nullish-coalescing>

Reached Stage 3

- <https://github.com/tc39/proposal-private-methods>
- <https://github.com/tc39/proposal-class-fields>
- <http://github.com/tc39/proposal-static-class-features/>
- <https://github.com/tc39/proposal-numeric-separator>
- <https://github.com/tc39/proposal-top-level-await>
- <https://github.com/tc39/proposal-string-replaceall>
- <https://github.com/tc39/proposal-promise-any>

Reached Stage 2

- <https://github.com/tc39/proposal-decorators>
- <https://github.com/tc39/set-methods>
- <https://github.com/tc39/proposal-realms>
- <https://github.com/domenic/proposal-arraybuffer-transfer/>
- <https://github.com/tc39/proposal-temporal>
- <https://github.com/tc39/proposal-iterator-helpers>
- <https://github.com/thumbsupep/proposal-upsert>

Thank you!

Merry
Xmas

