

Javascript ES6

Nouveautés ECMAScript 6

Mathieu Lallemand - 01.2016



Maintenant c'est sérieux.

ECMA Wat ?!

- **European Computer Manufacturers Association** - Association Chargé des standards (aujourd'hui internationale)
- Netscape soumet le JS à l'ECMA en 1997. Donne naissance à la norme ECMAScript. Permet d'unifier les langages de scripts orientés prototype (JS et feu ActionScript) sous la norme ECMA-262
- Il y a 6 versions existante à ce jour de l'ECMA-262. C'est pourquoi cette norme est appelée ECMAScript 6.
 - v1: Version de Netscape (l'original)
 - v2: Version remaniée au standard international ISO/IEC 16262
 - v3: Ajout des try/catch, meilleure gestion des chaînes, etc.
 - v4: L'erreur de jeunesse (arrêt du prototype O.o) -> Version abandonnée
 - ~~v3.1~~ v5: L'union des grands (Google, Yahoo, Microsoft, etc.) refondent la v3 vers une version plus propre et moins ambiguë
 - v6: Classes, modules, boucles for/of, générateurs, fonctions fléchées, ...
 - v7: *sync/await, opérateur d'exponentiation **, opérateur de binding ::, objets typés, ...*

Normes ECMAScript 6

- Norme 6 finalisée en Juin 2015
- Intégration progressive dans les navigateurs en cours
- Norme permettant d'envisager une utilisation industrielle du javascript
- Utilisables à l'aide de polyfill/shims comme la librairie Babel.JS ou es6-shim (trop?!) souvent possible après recompilation vers ES5.

Arrow Functions

```
var myValues = [1,2,3,4,5];  
var squared = myValues.map(x => x*x);  
console.log(squared); // -> [1, 4, 9, 16, 25]
```

Rest Parameter

```
function multiply(multiplier, ...theArgs) {  
  return theArgs.map(function (element) {  
    return multiplier * element;  
  });  
}
```

```
var arr = multiply(2, 1, 2, 3);  
console.log(arr); // [2, 4, 6]
```

```
var arr = multiply(2, ...[1, 2, 3]);  
console.log(arr); // [2, 4, 6]
```

Default Parameter

```
function sayMsg(msg='This is a default message.') {  
    console.log(msg);  
}  
sayMsg();  
sayMsg('This is a different message!');
```

Array Comprehension

```
var square = [for (i of [ 1, 2, 3 ]) i*i ];  
// [ 1, 4, 9 ]
```

```
var abc = [ 'A', 'B', 'C' ];  
var lowChar = [for (letters of abc) letters.toLowerCase()];  
// [ 'a', 'b', 'c' ]
```

```
//On peut rajouter des conditions  
[for (i of [1, 2, 3, 4]) if (i > 3) i];  
// [ 1, 2 ]
```

```
//utiliser plusieurs tableaux  
var cross = [for (i of [1, 2, 3]) for (j of [a, b, c]) i+j];  
// [ '1a', '1b', '1c', '2a', '2b', '2c', '3a', '3b', '3c' ]
```


Générateurs

```
function simpleGenerator(){  
  yield "first";  
  yield "second";  
  yield "third";  
  for (var i = 0; i < 3; i++)  
    yield i;  
}
```

```
var g = simpleGenerator();  
print(g.next()); // prints "first"  
print(g.next()); // prints "second"  
print(g.next()); // prints "third"  
print(g.next()); // prints 0  
print(g.next()); // prints 1  
print(g.next()); // prints 2  
print(g.next()); // StopIteration is thrown
```

Template Strings

```
let person = {name: 'Mathieu Lallemand'};
```

```
let tpl = "My name is ${person.name}.";
```

```
console.log(tpl); //My name is Mathieu Lallemand
```

Modules natifs

```
// fichier utils.js
function generateUUID () {
  return 'xxxxxxxx-xxxx-4xxx-yxxx-xxxxxxxxxxxx'.replace(/[xy]/g, function (c) {
    var r = Math.random() * 16 | 0, v = c == 'x' ? r : (r && 0x3 | 0x8);
    return v.toString(16);
  });
};

export { generateUUID }
```

```
import { generateUUID } from 'utils';
var uuid = generateUUID();
ou
import 'utils' as utils;
console.log(utils.generateUUID());
```

Classes

```
class Person {  
  constructor(firstName, lastName, age) { //constructors!  
    this.firstName = firstName;  
    this.lastName = lastName;  
    this.age = age;  
  }  
  
  fullName(){  
    console.log(this.firstName + " " + this.lastName);  
  }  
}
```

```
class Employee extends Person {  
  constructor(firstName, lastName, age, salary) {  
    super(firstName, lastName, age);  
    this.salary = salary;  
  }  
  
  printSalary(){  
    console.log('Salary : ' + this.salary);  
  }  
}  
  
let mat = new Employee('Mathieu', 'Lallemand', '36', 100000);  
mat.fullName();  
mat.printSalary();
```

Objets

//ES5

```
var obj = {  
  myMethod: function () {}  
};
```

//ES6

```
var obj = {  
  myMethod() {}  
};
```

```
let x = 4;
```

```
let y = 1;
```

```
let es5obj = {
```

```
  x : x,
```

```
  y : y
```

```
};
```

```
let es6obj = { x, y };
```

Attention aux pièges de fourbes !

Exemple : Assignment Destructurante

```
let clé = "z";  
let { [clé]: toto } = { z: "truc" };
```

Donc évidemment console.log(toto);

https://developer.mozilla.org/fr/docs/Web/JavaScript/Reference/Op%C3%A9rateurs/Affecter_par_d%C3%A9composition

Table de compatibilité

[illegible]

<https://kangax.github.io/compat-table/es6/>

Pour tout savoir

https://developer.mozilla.org/fr/docs/Web/JavaScript/Nouveaut%C3%A9s_et_historique_de_JavaScript/1.7