



ECMAScript 2015

Introduction

- Historique du JavaScript
- Différence entre ECMAScript et JavaScript
- Comment est définie la norme
- Rappels sur NodeJS

Historique du JavaScript

- Inventé en 1989 par Tim Berners Lee au CERN
- Web :
 - HTTP
 - URI
 - HTML
- Diffuser des documents universitaires

- Basé sur le SGML
- Titres, liens, listes et du texte
- Afficher des documents...
- ... et les relier via des liens
- Pas de spécification officielle

- Beaucoup d'évolutions entre 1992 et 1997
- Evolutions amenées par les navigateurs



- Naissance du W3C en 1995
- Aboutissement avec HTML 4.0 en 1997
 - Scripts
 - Css
 - Objects et frames

- 2008 : 1^{er} Draft HTML5
- De 1997 à 2007 : XHTML 1.0 et 1.1
 - Syntaxe XML
 - Remise à plat des balises
 - Trop ambitieux et peu suivi par les développeurs
- Web Sémantique



- Renouveler le langage
- Volonté d'aller + vite que le W3C
- Pose les bases du HTML5
- Ouverture sur l'extérieur

- En 2007, le W3C lance le groupe HTML5
- Premier Draft basé sur les travaux du WHATWG
- Retour à la syntaxe HTML classique
- Abandon de XHTML 2.0
- Garde la rétrocompatibilité HTML 4

- Validé depuis fin 2014
- Recommandé à l'utilisation depuis 2011
- Nouvelle séparation entre W3C et WHATWG
- Travaux sur HTML5.1 et 5.2
- Chaque spécification suit son propre agenda

- 2 normes distinctes
- Toutes les deux standardisées par le W3C
- Mais par deux groupes différents
- Fonctionnement et processus quasi identiques

Et le JavaScript dans tout ça ?



- 1997 : Version 1 *La version initiale*
- 1999 : Version 3 *Le JavaScript tel que vous le connaissez*
- 2009 : Version 5 *Mode strict*
- 2011 : Version 5.1 *Début de l'unification*
- 2015 : Version 6 *Harmony*

- Standardisé par l'ECMA
- Norme ECMA-262
- Groupe TC-39

ECMAScript = le standard

JavaScript = une implémentation de ce standard utilisé par les navigateurs

Harmony ? ES6 ? ES2015 ?

- L'objectif de ES2015 est d'uniformisé le standard dans tous les navigateurs : 1^{er} nom → Harmony
- Historiquement les versions de ES sont numérotés : 2^e nom → ES6
- Début 2015 le TC-39 décide de sortir 1 version par an et de remplacer la version par l'année : 3^e nom → ES2015

Tous ces noms désignent la même chose

Comment est établi le standard ?

- Chacun est libre de proposer des nouveautés
- 4 niveaux (stage) :
 - 0 : La proposition est émise
 - 1 : La proposition suscite de l'intérêt et est en discussion pour rejoindre la norme
 - 2 : Premières expérimentations pour valider la proposition
 - 3 : La proposition est en cours de finalisation
 - 4 : La norme est validée et stabilisée
- Une fois par an toutes les propositions ayant atteint le « stage 4 » à la date choisie sont automatiquement intégrée dans la version courante
- Si une proposition n'est pas prête, elle n'est pas intégrée
- Les navigateurs intègrent les nouveautés en fonction de leurs affinités tant qu'elles n'ont pas atteint le niveau 4



NodeJS

Qu'est ce que c'est ?

- Moteur JavaScript en dehors du navigateur
- Par défaut utilise V8 le moteur JavaScript de Chromium
- Possibilité d'utiliser Chakra le moteur de Edge
- Ajoute des APIs spécifiques pour interagir avec le matériel
- Permet de faire du JavaScript en dehors du navigateur :
 - Server
 - Outillage
 - Applications lourdes
 - Embarqué
- Possède un gestionnaire de dépendance : NPM
- Force l'utilisation de modules via commonJS

- Tout ce que vous codez dans un fichier est isolé du reste

- Pour exposer votre code :

```
module.export = myFunction;
```

```
module.export = {  
  functionToExport: myFunction  
  anotherFunctionToExport: myOtherFunction  
};
```

- Pour importer un autre module :

```
var exportedFunction = require('path/to/module');  
var exportedFunction =  
require('path/to/module').aSpecificFunction;
```

- Permet d'installer des modules fournis par des tiers
- Gère les versions et les dépendances
- Se base sur un fichier package.json

Installer un module :

```
npm install mon_module
```

```
npm install mon_module@version
```

```
npm install --global mon_module
```

```
npm install --save mon_module
```

```
npm install --save-dev mon_module
```

Supprimer un module :

```
npm remove mon_module
```

Installation de tous les modules définis dans le fichier package.json :

`npm install`

Utilisation de scripts :

- Pour déclencher des lignes de commandes lors de l'installation
- Lancer l'application avec `npm start`
- Lancer des tests avec `npm test`
- Définir des scripts personnalisés

