Javascript

Introduction

Historique

- Créé par Brendan Eich en 1996
- normalisé par ECMA en 1998
- 3è édition en juin 2002
- 5è édition en juin 2011
- 6è édition en juin 2015
- changement du processus de normalisation en 2016
- 7è édition en juin 2016
- 8è édition en juin 2017
- etc...

Hier

- toutjavascript.com
- francaisfacile.com

Aujourd'hui

- plus360degrees.com
- iss-sim.spacex.com
- earth.nullschool.net

Son nom

Javascript is to Java what hamster is to ham

- Confusion persistente avec Java
- Suffixe "script" laissant croire à un sous-langage

Domaines d'action

Navigateurs évidemment, mais aussi :

- Node.js (javascript côté serveur)
- Extensions des logiciels Mozilla (Firefox, Thunderbird)
- Flash (ActionScript)
- Suite Adobe (Photoshop, Illustrator, etc)
- Windows (**W**indows **S**cripting **H**ost)
- etc

Compatibilité des navigateurs

Problématique à l'époque d'Internet Explorer, beaucoup moins maintenant.

Microsoft Edge s'appuie désormais sur Chromium, la même base que Google Chrome. Les différences entre Chromium et Firefox sont mineures.

Editeur de code

Visual Studio Code s'impose comme le meilleur (open source et portable).

Outils intégrés au navigateur

- Console Web (F12)

Outils de développement

- Node.js pour gérer les dépendances (NPM) et installer d'autres outils.
- Webpack pour empaqueter les fichiers javascript et autres (images, css, etc) de manière optimisée. Parcel est à mentionner quant à sa simplicité d'utilisation.

Aperçu syntaxique

Sensibilité à la casse

```
let toto = "toto";
let Toto = "tata";

//toto et Toto sont 2 variables distinctes.
```

Aperçu syntaxique

Points-virgules

Malgré un mécanisme d' insertion automatique de points-virgules , il est conseillé pour débuter de terminer toutes les instructions par un point virgule.

```
let toto = "toto"
let Toto = "tata"
console.log(Toto)
//ok mais déconseillé
```

Ecriture préconisée

```
let toto = "toto";
let Toto = "tata";
console.log(Toto);
```

Point-virgules
Exemple de piège

```
let toto = "toto";

[1,2,3,4].forEach(function(nb) { console.log(nb); });

//écrit 1, 2, 3 et 4 dans la console de debuggage
```

```
let toto = "toto"

[1,2,3,4].forEach(function(nb) { console.log(nb); });

/*
Exception: "toto"[4] is undefined
le caractère [ n'appelle pas le mécanisme d'insertion automatique
de points-virgule
*/
```

Aperçu syntaxique
commentaires

```
let toto; //commentaire sur une seule ligne

/* commentaire
sur
plusieurs
lignes
*/
```

Mode strict

variante restrictive de javascript

- élimine quelques erreurs silencieuses en les changeant en erreurs explicites (une exception sera levée).
- corrige les erreurs qui font qu'il est difficile pour les moteurs javascript d'effectuer des optimisations.
- interdit les mot-clés susceptibles d'être définis dans les futures versions de ECMAScript.

➡ Référence

Mode strict

Usage

Simple chaîne de caractères, qui sera ignorée par le moteur javascript s'il n'implémente pas le mode strict.

```
"use strict";

toto = "toto";
//Exception: assignment to undeclared variable toto

let tata = {
    nom : "tata",
    age : 25,
    nom : "tata"
};
//Exception: property name nom appears more than once in object literal
```

Mode strict

Où le placer ?

- En dehors d'une fonction, il s'applique à tout le fichier.
  ➡ bugs possibles si on concatène des fichiers pas tous écrits en mode strict.
- En début de fonction, il s'applique au contenu de la fonction (y compris les fonctions imbriquées).
  ➡ c'est la méthode préconisée.

```
function toto() {

    "use strict";

    //je suis en mode strict

    function tata() {
        //je suis en mode strict
    }
}
//je ne suis pas en mode strict
```