

101 – Introduction

JavaScript

WIK-JS-101

Durée estimée : 1h

Intervenant : Anthony Domingue <anthony@wikodit.fr>

Support de cours : Jeremy Trufier <jeremy@wikodit.fr>



WIK-JS

Programme JavaScript

101 – Introduction

102 – Les bases

103 – Le prototypage et la POO

104 – Callback, asynchrones, évènements

105 – L'asynchrone avec les promesses

1XX – découverte de la technologie

2XX – approfondissement de la technologie

3XX – rappels et pratique, niveau moyen d'algorithmie

4XX – concepts avancés, niveau avancé d'algorithmie

5XX – approfondissement experts

Au préalable

Préparation

Cette série de cours WIK-JS-1xx sera basé sur l'apprentissage du langage JavaScript pur, sans aucune notion de navigateur ou de Web.

Pour ce faire, nous utiliserons NodeJS qui permet de lancer des scripts JavaScript depuis votre ordinateur.

Si NodeJS n'est pas installé, installez-le (au moins v10.16.3), cf. slide suivant.

Terminal & Shell

Les notions d'utilisation d'un shell, comme bash, sont requises.

Sur macOS ou Linux, ouvrez simplement un terminal.

Sur Windows, il est possible d'activer un Linux intégré, sous-système Linux, il est grandement recommandé de le faire -> <https://docs.microsoft.com/en-us/windows/wsl/install-win10>

Je recommande (personnellement) de se familiariser avec Linux ou Unix en général.

Sous macOS, d'utiliser au maximum le terminal, et d'installer un Linux pour ceux qui ont un PC (Manjaro, Ubuntu... libre à vous de trouver votre distribution).

90% des projets Web sont hébergés sur du Unix et utilisent des utilitaires Unix.

Terminal & Shell

Quelques ressources se familiariser et progresser avec l'utilisation de Linux

- <http://juliend.github.io/linux-cheatsheet/>
- <https://overthewire.org/wargames/bandit/>

Installer NodeJS

Pour installer NodeJS:

- sous macOS (téléchargez depuis le site <https://nodejs.org>, version Current)
- pour Ubuntu/Debian ou Windows avec le bash Ubuntu:

```
curl -sL https://deb.nodesource.com/setup_12.x | sudo -E bash -  
sudo apt-get install -y nodejs
```

Fermez et Ré-ouvrez le terminal et vérifiez que tout fonctionne

```
$ node  
> console.log('hello')  
hello  
undefined
```

La commande node démarre un interpréteur JavaScript
L'instruction JavaScript `console.log('hello')` permet d'afficher du texte

Préparation

Tout d'abord, créez un dossier pour les cours de JavaScript :

- Ouvrir un terminal
- Naviguez vers ce dossier (avec la commande `cd dossier1/sousdossier/...`)
- Créez un dossier appelé js-101 (avec la commande `mkdir js-101`)
- Naviguez vers ce dossier (`cd js-101`)

Tout ce qui se passera dans ce cours devra se passer dans ce dossier "js-101".

Essayez de créer au moins un dossier de travail par cours pour vous y retrouver.

Introduction

Le JavaScript

C'est un langage de programmation

Il est Orienté Objet à Prototype

Il est principalement utilisé pour rendre dynamique des pages Web

Il est aussi beaucoup utilisé côté serveur avec NodeJS

Histoire

Mai 1995 – Création du LiveScript par Brendan Eich, pour le compte de Netscape qui voulait un langage de programmation facile d'accès, il s'inspire du Java mais aussi du C

Décembre 1995 – Sun et Netscape annoncent la sortie du JavaScript. Ils ont changé le nom pour des raisons marketing et également par leur partenariat avec Sun Microsystems qui édite Java, l'idée étant que la popularité de l'un fasse monter celle de l'autre. => Confusion entre Java et Javascript

Mars 1996 – Le moteur JavaScript est intégré à Netscape Navigator 2.0, gros succès

Août 1996 – Microsoft réagit en apportant un langage similaire à Internet Explorer 3.0, c'est le JScript

1996 - 1997 – Netscape fait standardiser le langage par Ecma International. Ecma crée donc une édition ECMA-262 qui spécifie le langage qu'ils décident d'appeler ECMAScript

Depuis c'est la fondation Mozilla qui gère officiellement l'implémentation du langage JavaScript selon la spécification de l'ECMAScript.

Microsoft maintient le JScript en se basant également sur la spécification ECMAScript, la dernière version du JScript se trouve sur IE9

ECMAScript / JavaScript

Juin 1997 – ECMAScript 1

Juin 1998 – ECMAScript 2

Décembre 1999 – ECMAScript 3 (nombreux changements)

Décembre 2009 – ECMAScript 5 (clarifications et améliorations du langage dans l'objectif d'une standardisation, poussé par l'engouement autour de NodeJS)

Juin 2011 – ECMAScript 5.1 (changement mineurs)

Juin 2015 – ECMAScript 6 (changements majeurs, notamment au niveau de la syntaxe, introduction des classes, refonte progressive du langage)

Juin 2016 – ECMAScript 7, ES2016 (continuation de la refonte, des efforts sur la propreté et la qualité)

Juin 2017 – ECMAScript 8, ES2017 (atomicité, async/await, etc...)

Juin 2018 – ECMAScript 9, ES2018 (iterator synchrone et asynchrone, rest/spread operator...)

Juin 2019 – ECMAScript 10, ES2019 (Array.flat...)

Juin 2020 ? – ESNext (futur du langage)

Version ES ?

Pour des compatibilité navigateurs, les développeurs étaient obligés d'écrire leurs programme en ES3 pendant très longtemps, la transition vers ES5 fut très dure.

Nous allons utiliser ES6... puis même ES7, et finalement pourquoi pas ES8 et ES9 ou ES10 ?

Il vaut mieux profiter des langages modernes, et trouver une solution de rétro-compatibilité (ex: Babel).

Avantages

- Le langage le plus Objet orienté, presque tout est un objet
- Un langage très simple pour commencer la programmation
- Possible de créer un web-server, backend/api avec NodeJS
- Le Javascript peut manipuler le DOM (la structure) des pages Web, et donc changer très simplement l'apparence ou la structure de façon dynamique
- JavaScript est théoriquement non-bloquant, et asynchrone, il est donc possible d'exécuter plusieurs tâches simultanément
- Syntaxe inspirée de Java et du C
- Langage grandissant

CLI

Command Line Interface

Commande `node`

Terminal ou PowerShell

Evaluation de script

Mode interactif

Code javascript

Output

Return

Pour quitter

Execution d'un fichier JS

```
$ node -e "console.log('hello')"  
$ node  
> console.log('hello')  
hello  
undefined  
> 1+3  
4  
> .exit  
  
$ node ./hello.js  
Hello World  
$
```


Félicitations !!

Cours WIK-JS-101 burned :)