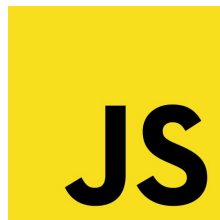
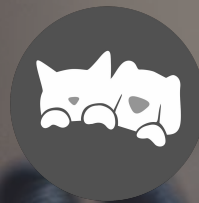


**Introduction**

**Javascript**



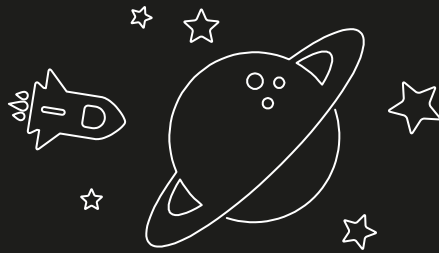


**Karl MARQUES BERNARDO**

CTO Slyvent  
CTO Vetixy

kmarques@vetixy.com  
[ESGI] [JS] [5IW2]

<https://github.com/kmarques>

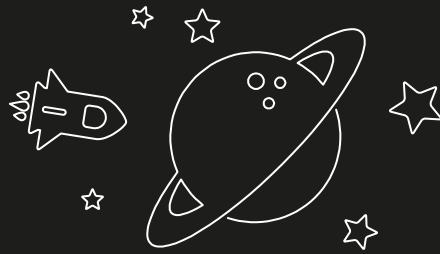


# MISSIONS

3

# 4

- Avoir de la culture générale
- Connaître les bases du Javascript
- Savoir lire et comprendre un code Javascript
- Participer à la communauté Open Source



# EVALUATION

5

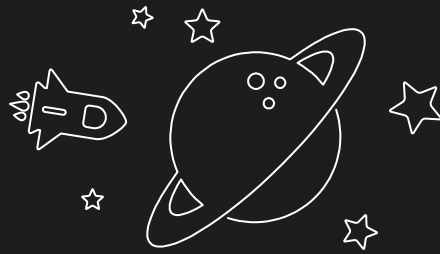


# Contrôle continu

## QCM + Question ouverte

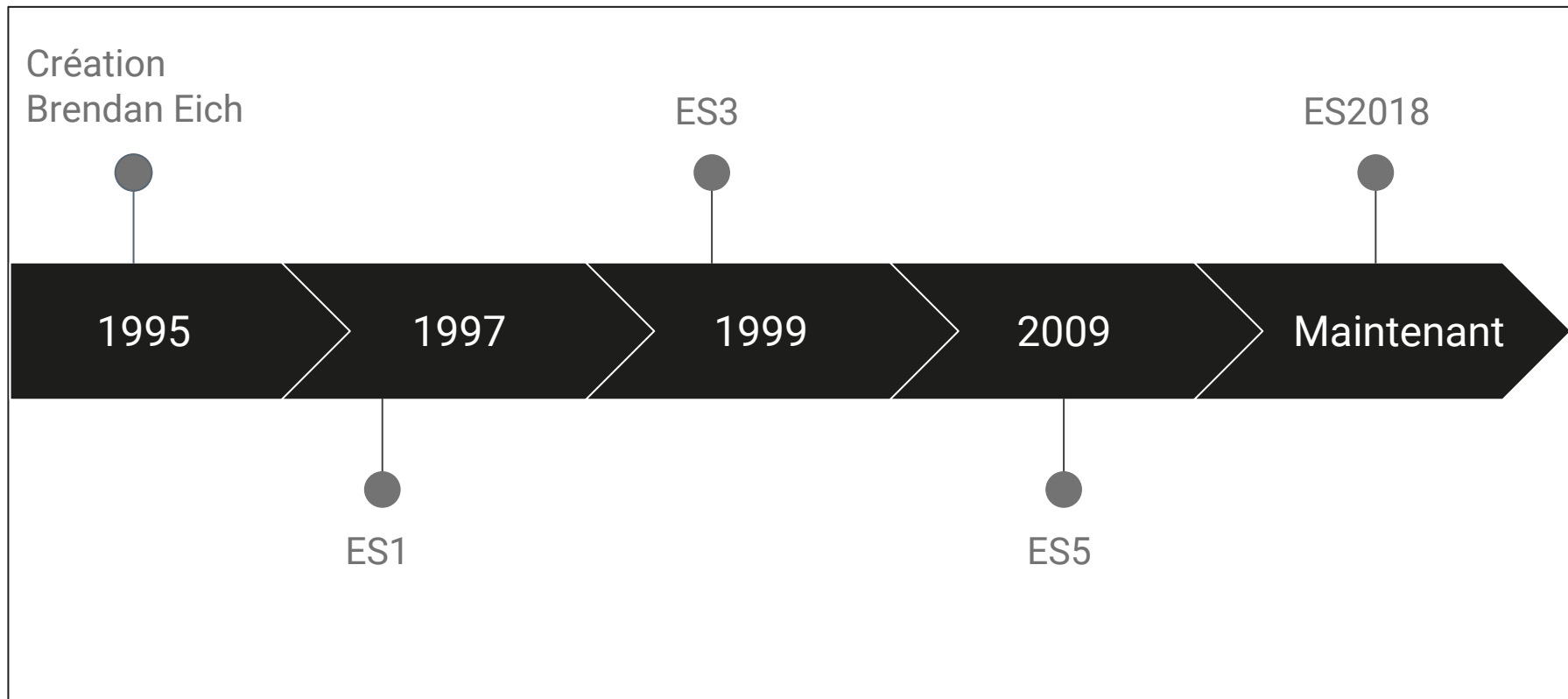
## Evaluation finale

## Projet Individuel



# HISTORIQUE

7

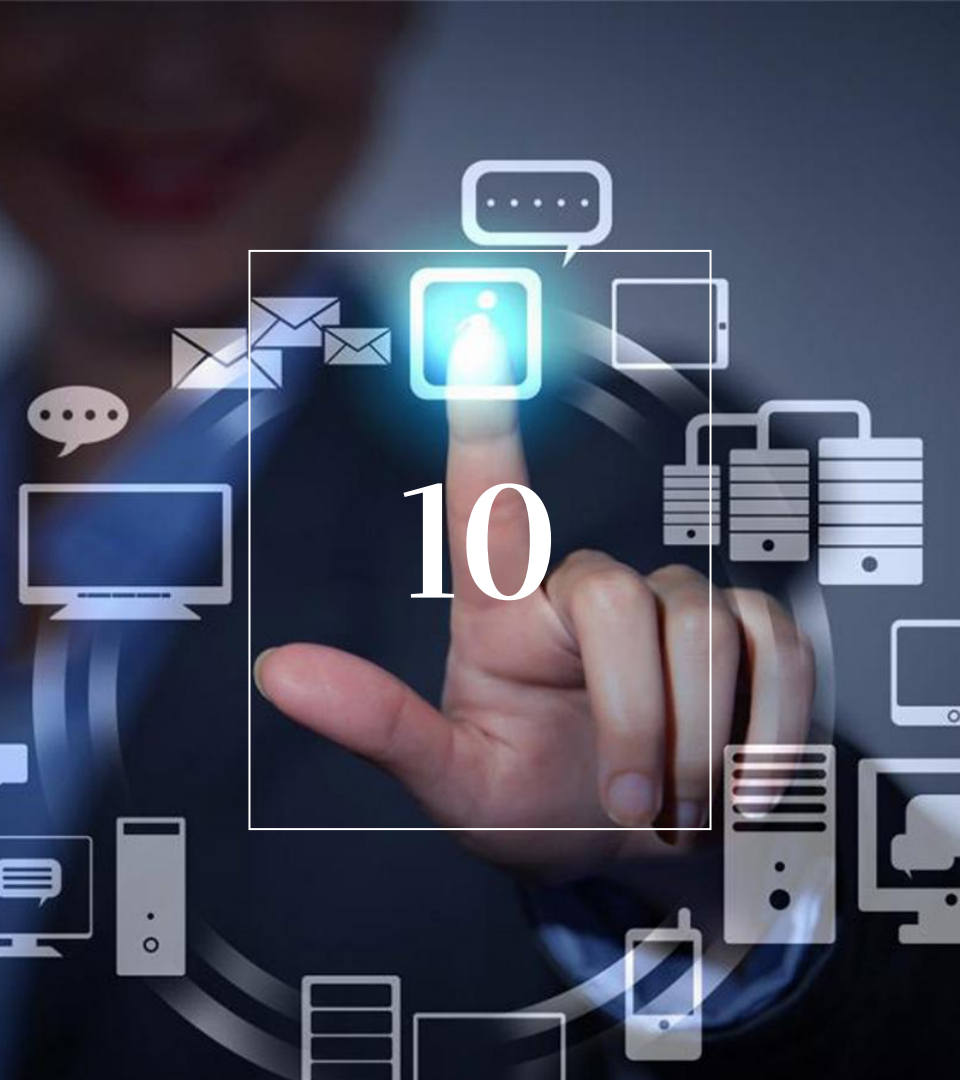






**ECO-SYSTEME**

9



## Où exécuter son script ?

### Navigateur WEB

Page Web

Balise `<script>`

Fichier .js

Console

Extensions

Chrome

Tamper(Grease)Monkey

### Terminal

NodeJS

### Base de données

MongoDB

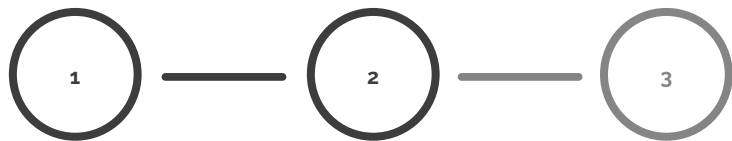
# 11

**Comment exécuter de l'ES6 ou ultérieur ?**

**LES TRANSPILLERS ou "compilateurs"**  
(Babel / Browserify / ...)

Transformer du code Javascript

Next-Gen en javascript multi-plateformes



**Parser**

**Transformers**

**Generator**

Code -> AST

AST -> AST

AST -> Code

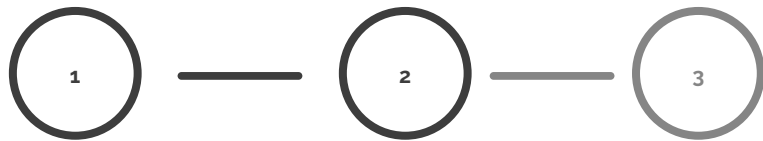
(plugins/presets)

# 12

## Les toolchains ou task-runners

Même concept que pour les TRANSPILLERS  
mais appliqué à l'ensemble des assets d'une  
application

(Gulp / Grunt / WebPack)



**Inputs**

**Transformers**

**Generator**

**Collecte des  
entrées**

**Modification des  
entrées**

**Sorties**

fichiers  
var Env  
string

plugins



# 13

## Les gestionnaires de paquets

### NPM

Gestion automatique des dépendances

### YARN

Mode Hors-ligne

Meilleure gestion des dépendances

Plus sécurisé

### BOWER

Gestion manuelle des dépendances

Plateforme annexe de paquets

Mort et enterré



# INSTALLATION

14

1. Installation de Docker et Docker Compose
2. Enregistrer le docker-compose.yml

```
version: '3'
services:
  node:
    image: 'node:latest'
    volumes:
      - './:/home/node/app'
    working_dir: '/home/node/app'
    ports:
      - '3000:3000'
    command: 'npm install && npm start'
```
3. Créer un repo github *esgi-javascript*
4. Envoyer le lien à l'adresse  
<https://goo.gl/forms/3QhLdmaaaXxrKUnh1>
5. Structure du dossier

```
./docker-compose.yml
./README.txt
./intro/
./exercise-{num}/
./project/
```
6. Créer un fichier *./intro/hello.js*  
*console.log("Hello World!");*
7. Exécuter la commande  
*docker-compose run node node ./intro/hello.js*
8. Commit & Push  
*[DONE] intro*