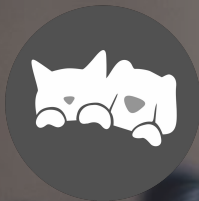


NodeJS





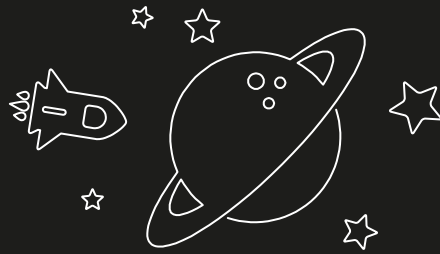
Karl MARQUES BERNARDO

CTO Vetixy

kmarques@vetixy.com

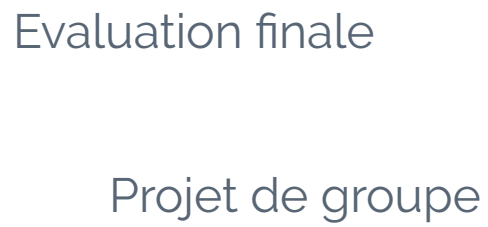
[WSF] [NODE]

<https://github.com/kmarques>

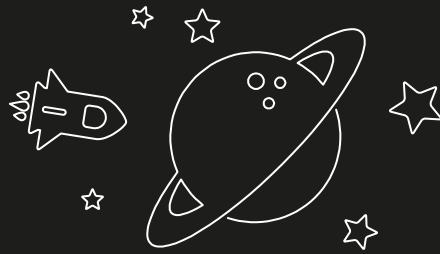


EVALUATION

3



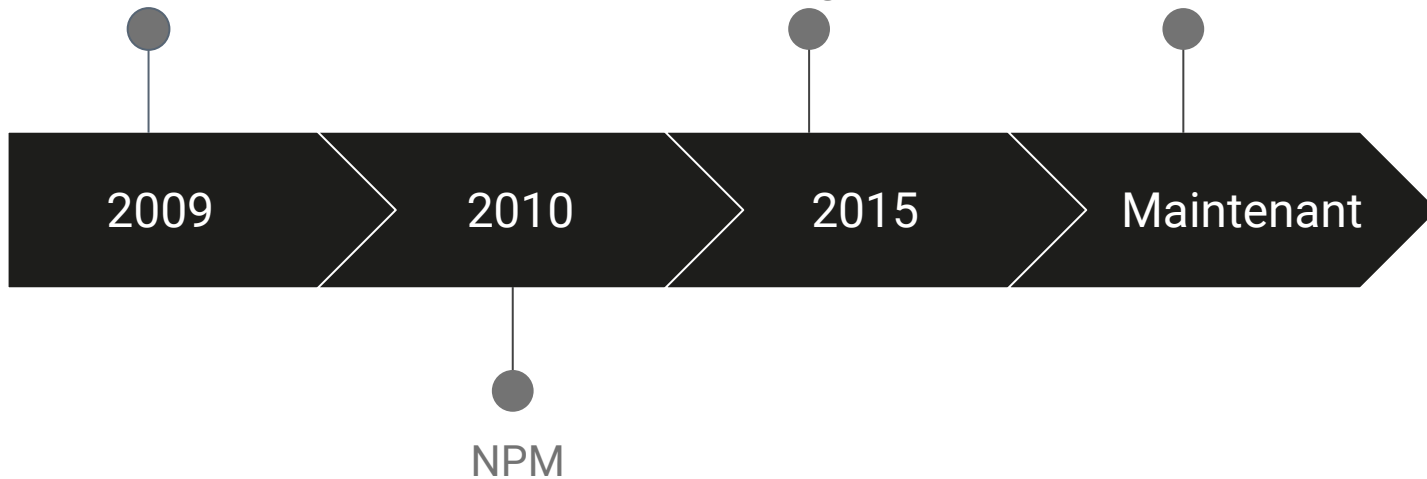
Projet de groupe



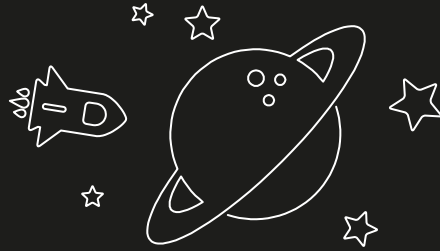
HISTORIQUE

5

Création
Ryan Dahl

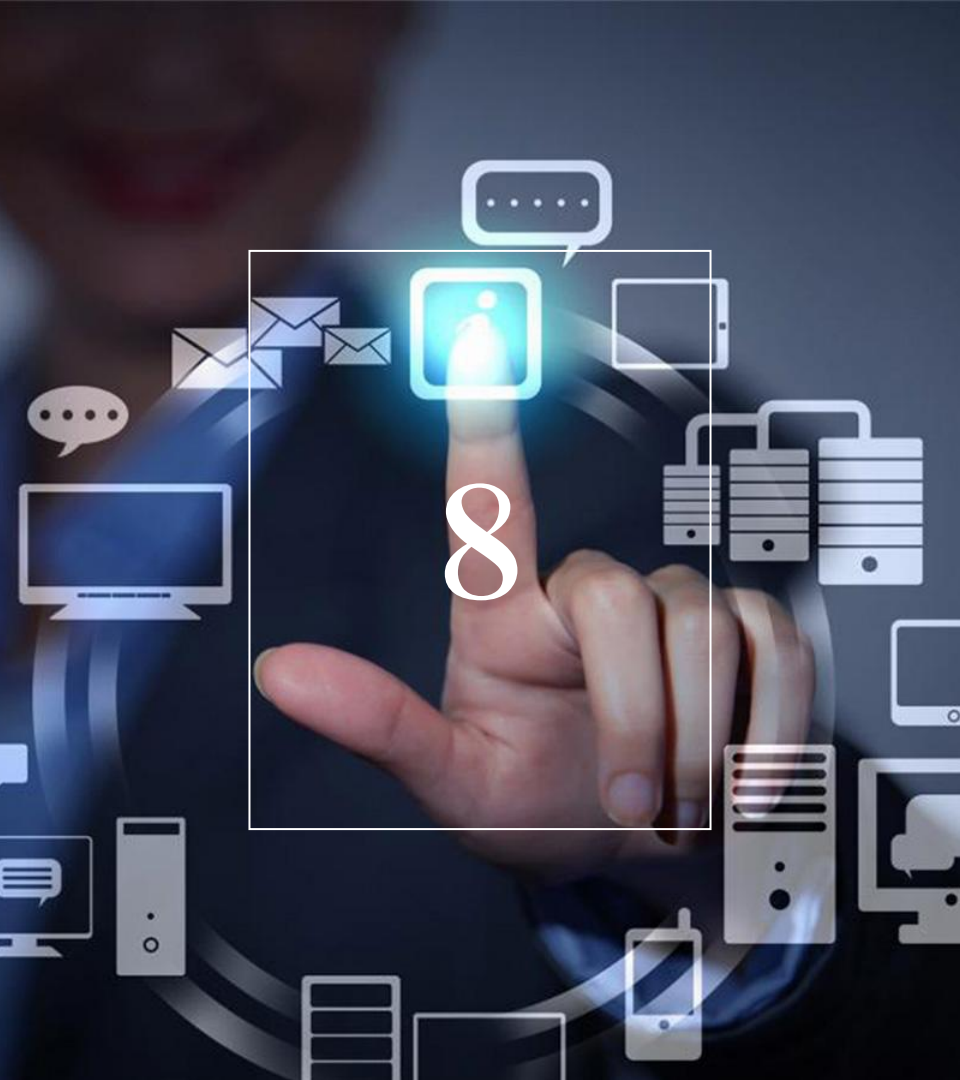


6



ECO-SYSTEME

7



Où exécuter son script ?

Terminal client/serveur

Linux: Bash, Sh, zsh

Windows: WSL, Batch, PowerShell

Base de données

MongoDB

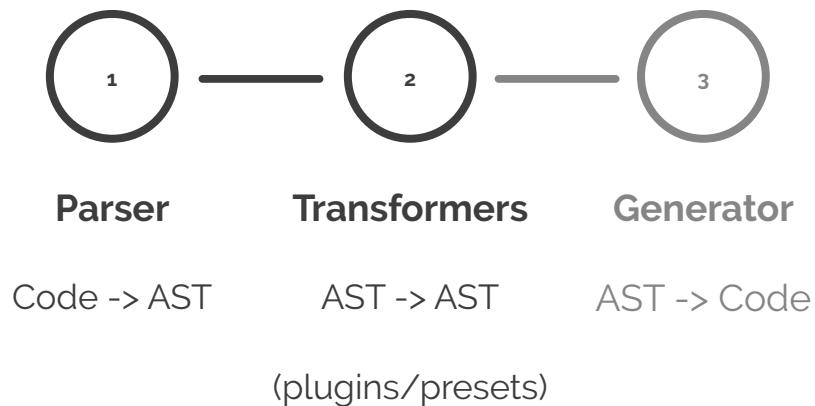


Comment exécuter de l'ES6 ou ultérieur ?

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

Transformer du code Javascript

Next-Gen en javascript multi-plateformes

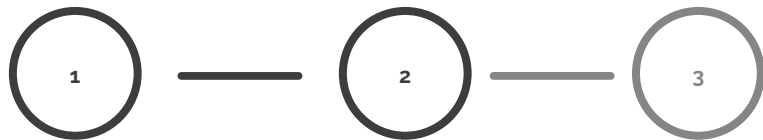


10

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

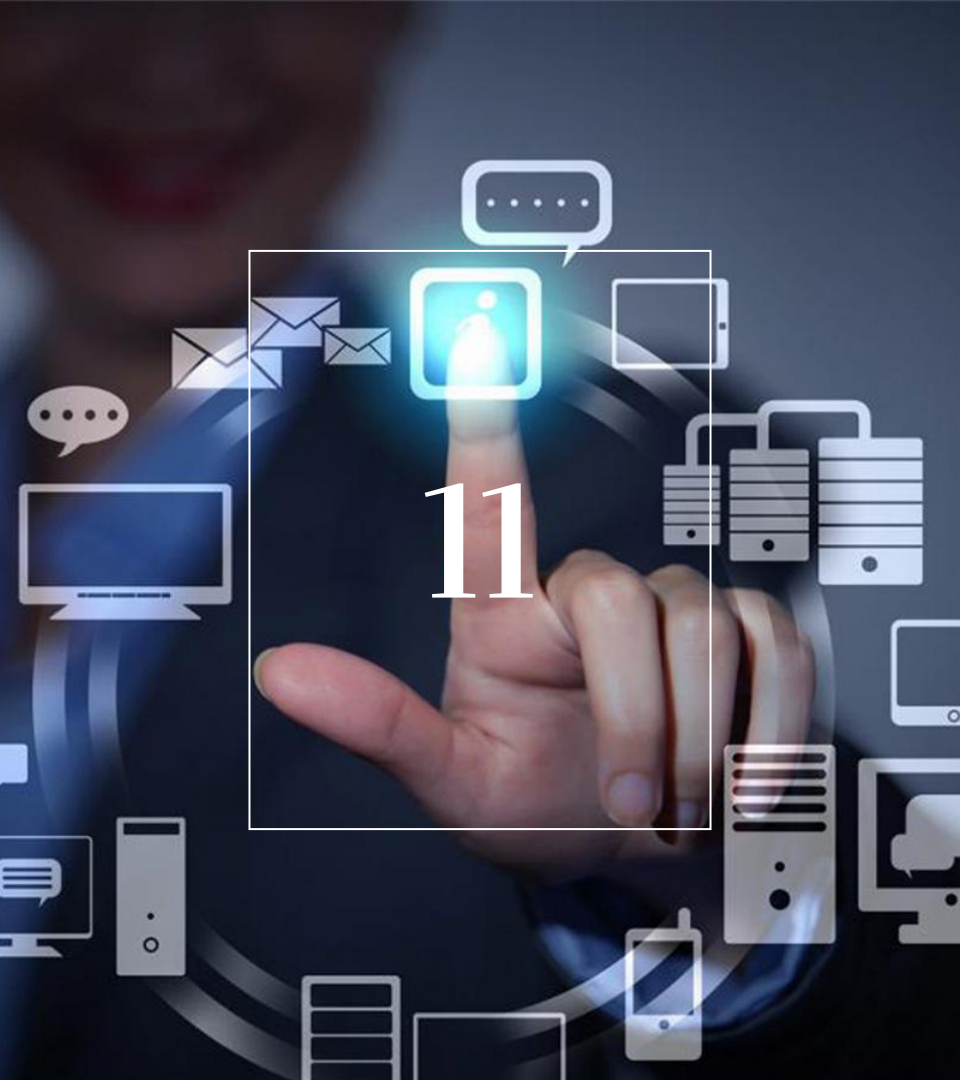
**Collecte des
entrées**

**Modification des
entrées**

Sorties

fichiers
var Env
string

plugins



Les gestionnaires de paquets

NPM

Gestion automatique des dépendances

YARN

Mode Hors-ligne

Meilleure gestion des dépendances

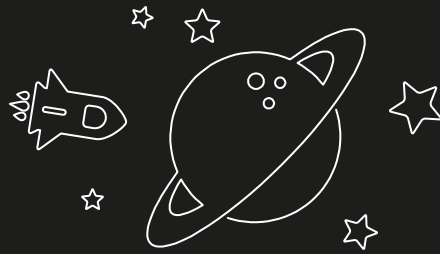
Plus sécurisé

BOWER

Gestion manuelle des dépendances

Plateforme annexe de paquets

Mort et enterré



INSTALLATION

12

1. Docker : docker-compose.yml

```
version: '3'
services:
  node:
    image: 'node:13-alpine'
    volumes:
      - './:/home/node/app'
    working_dir: '/home/node/app'
    ports:
      - '3000:3000'
    command: 'npm install && npm start'
```

2. Mac : Homebrew

```
brew install nodejs
```

3. Linux

```
curl -sL https://deb.nodesource.com/setup_13.x | sudo -E bash -
sudo apt-get install -y nodejs
```

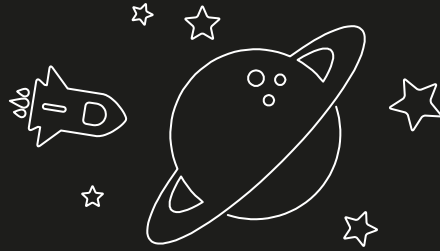
4. Windows : Télécharger l'installeur

WSL : utiliser la méthode Linux

Autres:

<https://nodejs.org/en/download/>

Pensez à ajouter au PATH !!!



Rappel des bases

15

Les types de données

Types primitifs

String => "message"

Number => 2 2.3

Boolean => true false

Undefined => undefined

Types complexes

Object

object => { prop1: true, prop2: "yes" }

null => null

array => [1, 2, 3]

Function

16

Portées des variables

foo = 3

Déclaration globale

var foo = 3

Déclaration locale

let foo = 3

Déclaration au block

const foo = 3

Let + constante de référence

Attention !!! Phénomène de Hoisting

Javascript remonte automatiquement les déclarations de variables sans les initialiser en haut du scope courant pour le keyword **var**

Exemples

17

```
var foo = 1;

(function() {

    console.log(foo);      Undefined

    var foo = 2;

    var baz = 3;

    bar = 4;

})();

console.log(foo);      1
console.log(bar);      4
console.log(baz);      erreur
```



Les closures

Fonction dans une fonction

Javascript se souvient de l'environnement de chaque fonction.

Une variable qui ne devrait plus exister peut donc continuer à être appelée.

```
function creerFonction() {  
  var nom = "Mozilla";  
  function afficheNom() {  
    console.log(nom);  
  }  
  return afficheNom;  
}
```

```
var maFonction = creerFonction();  
maFonction();
```

Démonstration

capitalize

1ère lettre de chaque mot en MAJ
hello world => Hello World

camelCase

Capitalize + coller les mots
hello world => HelloWorld

Exercices

snake_case

Joindre les mots par des underscores en MIN
Hello World, i am john => hello-word-i-am-john

leet - Cryptage (uniquement les voyelles)

anaconda => 4n4c0nd4
A=>4, E=>3, I=>1, O=>0, U=>(_), Y=>7

yoda

Inverser la position des mots d'une phrase
Hello world => world Hello

**Renvoyer une chaîne vide en cas de valeur
non String**