

# Vue.js : introduction

**Achref El Mouelhi**

Docteur de l'université d'Aix-Marseille  
Chercheur en programmation par contrainte (IA)  
Ingénieur en génie logiciel

`elmouelhi.achref@gmail.com`



# Plan

- 1 Introduction
- 2 IDE : Integrated Development Environment
- 3 Vue Devtools
- 4 Solution avec CDN
- 5 Solution avec Vite
  - Installation
  - Création et structure d'un projet créé avec Vite
- 6 Solution avec CLI
  - Installation
  - Création et structure d'un projet CLI
- 7 Commandes utiles

## Framework

- Plusieurs traductions
  - cadriciel
  - environnement de développement
  - cadre d'applications
  - ...
- Ensemble de composants logiciels
- Facilitant la réalisation d'une (partie de l') application
- Imposant une certaine structure, logique, syntaxe...

## Plusieurs types de Framework

- Frameworks applicatifs pour le développement d'applications web :
  - **Vue.js** pour **JavaScript**,
  - **Spring** pour **Java**,
  - **Symfony** pour **PHP**
  - ...
- Frameworks de présentation de contenu web :
  - **Bootstrap** pour **CSS**
  - ...
- Frameworks de persistance
- Frameworks de logging
- ...

## Framework : avantages

- Gain de temps
- Meilleure organisation de projet
- Respect de bonnes pratiques
- Meilleure efficacité
- Faciliter le travail d'équipe
- ...

## Framework : inconvénients

- Complexité
- Apprentissage
- ...

# Vue.js

## Vue.js

- Appelé aussi **Vue** (et prononcé **View**)
- Framework **JavaScript** (ou **TypeScript**)
- Écrit en **TypeScript**
- Open-source
- Créé par **Evan You** (ex salarié de **Google**) en 2014
- Permettant de créer des applications web et mobiles (**Vue Native**)
  - Front-End
  - Single page

## Vue.js respecte l'architecture **MVVM** : **Model-View-ViewModel**

- **Model** : représenté généralement par un objet/classe/interface référencé par la couche d'accès aux données
- **View**
  - contenant la disposition et l'apparence de ce qu'un utilisateur voit à l'écran,
  - recevant l'interaction avec l'utilisateur : clic, saisie, survol...
  - représenté dans **Vue.js** par la balise `<template>`.
- **ViewModel**
  - remplaçant du contrôleur dans l'architecture **MVC**,
  - connecté à la vue par le **data binding**,
  - représenté dans **Vue.js** par la balise `<script>`.



# Vue.js

## Vue.js utilise :

- les composants web
- le DOM Virtuel

## DOM Virtuel ?

- introduit, initialement, par **React**.
- une représentation en mémoire du **DOM** physique.
- permettant des mises à jour plus rapides et efficaces en ne modifiant que les parties nécessaires de l'interface utilisateur avant de synchroniser ces changements avec le **DOM** physique.

© Achref EL

## DOM Virtuel ?

- introduit, initialement, par **React**.
- une représentation en mémoire du **DOM** physique.
- permettant des mises à jour plus rapides et efficaces en ne modifiant que les parties nécessaires de l'interface utilisateur avant de synchroniser ces changements avec le **DOM** physique.

## Remarque

- Les modifications apportées au **DOM** physique entraînent généralement des opérations coûteuses en termes de temps de traitement.
- Chaque modification peut déclencher des réorganisations et des recalculs complexes de la mise en page.

## Quelques outils utilisés par **Vue.js**

- **npm** (**n**ode **p**ackage **m**anager) : le gestionnaire de paquets par défaut pour une application **JavaScript**.
- **vue-cli** command line interface : outil facilitant la création et la construction d'une application **Vue.js** en exécutant directement des commandes.
- **webpack** : bundler JavaScript
  - construit le graphe de dépendances.
  - regroupe des ressources de même nature (`.js` ou `.css...`) dans un ou plusieurs bundles.
  - fonctionne avec un système de module : un fichier **JS** est un module, un fichier **CSS** est un module...

## Les différentes versions d'**Vue.js**

- **Vue.js** 1.0 sorti en octobre 2015
- **Vue.js** 2.0 sorti en septembre 2016
- **Vue.js** 3.0 sorti en septembre 2020
- **Vue.js** 3.1 sorti en juin 2021
- **Vue.js** 3.2 sorti en août 2021
- **Vue.js** 3.3 sorti en mai 2023
- **Vue.js** 3.4 sorti en décembre 2023
- **Vue.js** 3.5 sorti en septembre 2024

## Quelques sites Web réalisés avec **Vue.js**

- **GitLab**
- **Alibaba**
- **Adobe**
- ...

# Vue.js

## Comment utiliser **Vue.js** ?

- en utilisant une **CDN** : **C**ontent **D**elivery **N**etwork
- en faisant une installation avec **NPM** : **N**ode **P**ackage **M**anager
- en installant **CLI** : **C**ommand **L**ine **I**nterface

# Vue.js

## Quel Environnement de Développement Intégré (IDE) pour Vue.js ?

- **Visual Studio Code** (À ne pas confondre avec Visual Studio)
- Eclipse
- ...

© Achref EL M...



# Vue.js

## Quel Environnement de Développement Intégré (IDE) pour Vue.js ?

- **Visual Studio Code** (À ne pas confondre avec Visual Studio)
- Eclipse
- ...

## Visual Studio Code (ou VSC), pourquoi ?

- Gratuit.
- Extensible selon le langage de programmation.
- Recommandé par les développeurs Front-end.

**VSC** : téléchargement

`code.visualstudio.com/download`

# Vue.js

**VSC** : extension pour **Vue.js 2**

**Vetur**

© Achref EL MOUËL

# Vue.js

**VSC** : extension pour **Vue.js 2**

**Vetur**

**VSC** : extension pour **Vue.js 3**

- **Volar** remplacé par **Vue - Official**
- **Vue 3 Snippets**

## Quelques raccourcis VSC

- Pour activer la sauvegarde automatique : aller dans `File > AutoSave`
- Pour indenter son code : `Alt` `Shift` `f`
- Pour commenter/décommenter : `Ctrl` `:`
- Pour sélectionner toutes les occurrences : `Ctrl` `f2`
- Pour sélectionner l'occurrence suivante : `Ctrl` `d`
- Pour placer le curseur dans plusieurs endroits différents : `Alt`

## Vue Devtools

- Extension multi-navigateur pour le débogage des applications **Vue.js**.
- Installation : <https://devtools.vuejs.org/guide/browser-extension>

Solution avec **CDN** : démarche

- Depuis **VSC**, allez dans `File > Open Folder...`
- Cliquez sur `Nouveau dossier` et saisissez `vue-cdn`
- Cliquez sur `vue-cdn` puis sur le bouton `Sélectionner un dossier`
- Dans `vue-cdn`, créez deux fichiers `index.html` et `script.js`
- Dans `index.html`, saisissez `html:5` ou `!` puis cliquez sur `Entree`

## Code généré

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>

<body>

</body>

</html>
```



# Vue.js

Référençons `script.js` dans `index.html`

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cours Vue.js</title>
</head>

<body>
  <script src="script.js">
  </script>
</body>

</html>
```

# Vue.js

Ajoutons la CDN (Vue.js)

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cours Vue.js</title>
</head>

<body>
  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
  <script src="script.js">
  </script>
</body>

</html>
```

**On peut aussi choisir un autre hébergeur (cdnjs par exemple)**

```
https://cdnjs.cloudflare.com/ajax/libs/vue/3.4.21/vue.cjs.js
```

© Achref EL MOUL

**On peut aussi choisir un autre hébergeur (cdnjs par exemple)**

```
https://cdnjs.cloudflare.com/ajax/libs/vue/3.4.21/vue.cjs.js
```

**Ou (jsdelivr)**

```
https://cdn.jsdelivr.net/npm/vue@3.4.21/dist/vue.global.min.js
```

Spécifions la zone de la page où l'application Vue.js sera chargée

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cours Vue.js</title>
</head>

<body>
  <div id='app'>
    {{ message }}
  </div>

  <script src="https://unpkg.com/vue@3/dist/vue.global.js"></script>
  <script src="script.js">
  </script>
</body>

</html>
```

Ajoutons le contenu suivant dans `script.js`

```
const App = Vue.createApp({
  data() {
    return {
      message: "Hello world"
    };
  }
});
App.mount('#app');
```

© Achref EL MOUEZ

Ajoutons le contenu suivant dans `script.js`

```
const App = Vue.createApp({
  data() {
    return {
      message: "Hello world"
    };
  }
});
App.mount('#app');
```

## Explication

- `Vue` : une classe **Vue.js** prédéfinie chargée depuis la **CDN**
- `createApp()` : méthode permettant de créer une application **Vue.js**
- `data()` : fonction permettant de retourner des données affichées souvent dans la zone d'application
- `message` : une variable contenant la valeur `Hello world`
- `App.mount('#app')` : méthode permettant de charger l'application **Vue.js** dans l'élément **HTML** ayant l'identifiant **app**

## Pour tester, utilisons l'extension **Live Server**

- Installez l'extension **Live Server**
- Faites un clic droit sur `index.html`
- Cliquez sur `Open with Live Server`
- Vérifiez l'affichage de `Hello world`



# Vue.js

## La version utilisant ES Modules

```
import { createApp } from 'https://unpkg.com/vue@3/dist/vue.esm-browser.js'

createApp({
  data() {
    return {
      message: "Hello world"
    };
  }
}).mount('#app')
```

# Vue.js

Et l'attribut `type="module"` dans la balise `script`

```
<!DOCTYPE html>
<html lang="fr">

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Cours Vue.js</title>
</head>

<body>
  <div id='app'>
    {{ message }}
  </div>

  <script src="script.js" type="module">
  </script>
</body>

</html>
```

### Solution avec **CDN** : limites

- Impossible de compiler le code (pas de fichier avec l'extension `.vue`)
- Impossible de créer et importer des composants

## Vite

- Outil de développement web léger et rapide
- Générateur de projet front-end
- Créé par **Evan you** (Créateur de **Vue.js**)
- Conçu, initialement, pour accélérer le processus de développement des applications **Vue.js**
- Open-source
- Prenant en charge les langages **JavaScript** et **TypeScript**
- Intégrant plusieurs frameworks et librairies : **React.js**, **Vue.js**, **Vanilla**, **Svelte...**

## Relation avec esbuild Rollup

- **Vite** est un outil de build moderne pour le développement frontend.
- Il utilise deux moteurs :
  - **esbuild** pour le développement (serveur rapide, transpilation JS/TS).
  - **Rollup** pour le build de production.
- **Vite** agit comme une surcouche de haut niveau qui cache la complexité de **Rollup**.
- **Rollup** gère la génération des bundles finaux, avec un système de plugins très modulable.

## Résumé

**Vite** = esbuild (dev) + Rollup (build prod)

## HMR : Hot Module Reloading

- Fonctionnalité de **Vite**
- Permettant de voir instantanément les modifications apportées à leur code sans avoir à rafraîchir manuellement la page du navigateur
- Utilisant les **WebSocket** pour assurer une navigation bidirectionnelle entre le serveur de développement (fourni par des outils comme **Vite**, **Webpack**...) et le navigateur

## Vite : un moteur adopté par les frameworks modernes

- **Nuxt 3** (depuis 2022) supporte **Vite** nativement.
- **React** : utilisation courante avec `vite + react` (`vite-plugin-react`).
- **SvelteKit** : utilise **Vite par défaut** depuis sa création (2022).
- **SolidJS** : avec **SolidStart** qui repose entièrement sur **Vite**.
- ...

## Même **Angular** s'y met

- **Angular 17** (novembre 2023)
  - Introduit `@angular/cli vite` comme moteur de build alternatif.
  - Objectif : un **serveur de dev plus rapide**, et un **HMR natif performant**.
- **@angular/build** (2024)
  - Nouveau pipeline de build basé sur **Vite** et **esbuild**.
  - Devient la future base par défaut pour **Angular CLI (Angular 18+)**.



## Pourquoi ce choix ?

- **Webpack** devenu trop lourd et lent à maintenir.
- La communauté **JavaScript** s'oriente vers des outils plus simples et performants.

© Achref EL M

# Vue.js

## Pourquoi ce choix ?

- **Webpack** devenu trop lourd et lent à maintenir.
- La communauté **JavaScript** s'oriente vers des outils plus simples et performants.

## Par conséquent

**Vite** s'impose comme un standard moderne, même dans les frameworks historiquement liés à **Webpack**.

# Vue.js

## Remarque

Pour installer **Vue.js**, il faut télécharger et installer **Node.js** (Dernière version stable LTS)

© Achref EL MOUELHI ©

# Vue.js

## Remarque

Pour installer **Vue.js**, il faut télécharger et installer **Node.js** (Dernière version stable LTS)

## Pour **Node.js**, il faut

- aller sur <https://nodejs.org/en/>
- choisir une version récente, télécharger et installer

# Vue.js

## Remarque

Pour installer **Vue.js**, il faut télécharger et installer **Node.js** (Dernière version stable LTS)

## Pour **Node.js**, il faut

- aller sur <https://nodejs.org/en/>
- choisir une version récente, télécharger et installer

**Pour vérifier l'installation depuis une console (invite de commandes), exécutez**

```
node -v
```

# Vue.js

## Remarque

- Pour créer un projet avec **Vite**, il existe trois commandes différentes.
- Nous utiliserons la dernière commande qui génère moins de composants

# Vue.js

**Première commande, exécuter la commande**

```
npm init vue@latest
```

© Achref EL MOU

# Vue.js

## Première commande, exécuter la commande

```
npm init vue@latest
```

## Deuxième commande, exécuter la commande

```
npm create vue@latest
```



# Vue.js

Et ensuite, répondre aux différentes questions ainsi

```
Vue.js - The Progressive JavaScript Framework
```

```
✓ Project name: ... vue-vite
✓ Add TypeScript? ... No / Yes
✓ Add JSX Support? ... No / Yes
✓ Add Vue Router for Single Page Application development? ... No / Yes
✓ Add Pinia for state management? ... No / Yes
✓ Add Vitest for Unit Testing? ... No / Yes
✓ Add an End-to-End Testing Solution? ... No / Yes / Cypress / Playwright
✓ Add ESLint for code quality? ... No / Yes
✓ Add Prettier for code formatting? ... No / Yes
```

```
Scaffolding project in <chemin-projet>
```

```
Done. Now run:
```

```
cd vue-vite
npm install
npm run dev
```

### Troisième commande, exécuter la commande

```
npm init vite@latest
```

© Achref EL MOUELHI ©

## Troisième commande, exécuter la commande

```
npm init vite@latest
```

## Ensuite

```
✓ Project name: ... cours-vue-vite
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
>  Vue
  React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```

## Troisième commande, exécuter la commande

```
npm init vite@latest
```

## Ensuite

```
✓ Project name: ... cours-vue-vite
? Select a framework: » - Use arrow-keys. Return to submit.
  Vanilla
>  Vue
  React
  Preact
  Lit
  Svelte
  Solid
  Qwik
  Others
```

## Et enfin

```
✓ Project name: ... cours-vue-vite
✓ Select a framework: » Vue
? Select a variant: » - Use arrow-keys. Return to submit.
  TypeScript
>  JavaScript
  Customize with create-vue
  Nuxt
```

# Vue.js

Ouvrez le dossier `vue-vite` avec VSC et installez les dépendances avec

```
npm install
```

© Achref EL ME

# Vue.js

Ouvrez le dossier `vue-vite` avec VSC et installez les dépendances avec

```
npm install
```

Pour lancer l'application, exécutez

```
npm run dev
```

# Vue.js

## Arborescence d'un projet **Vue.js**

- `node_modules` : contenant les fichiers **Node.js** nécessaires pour un projet **Vue.js**
- `src` : contenant les fichiers sources de l'application
- `package.json` : contenant l'ensemble de dépendance de l'application
- `public` : contenant les fichiers statiques à intégrer systématiquement dans `dist`
- `vite.config.js` : fichier de configuration de **Vite** pour un projet **Vue.js**)
- `index.html` : point d'entrée de l'application
- `README.md` : permet de fournir une vue d'ensemble de l'application, décrire son objectif et expliquer les fonctionnalités principales.

Si vous aviez choisi **TypeScript** comme langage, vous auriez

- `tsconfig.json` : fichier de configuration principal pour **TypeScript** pouvant contenir les paramètres de compilation, les chemins d'inclusion et d'exclusion, les options de module...
- `tsconfig.app.json` : fichier spécifique à l'application **Vue** pouvant contenir des propriétés comme chemins d'importation, les options de type...
- `package.node.json` : fichier de configuration permettant d'activer des fonctionnalités **TypeScript** spécifiques à **Node.js**, comme les modules CommonJS, les cibles ECMAScript...



# Vue.js

## Que contient `src` ?

- `assets` : contenant les fichiers statiques à intégrer dans `dist` s'ils sont explicitement référencés dans le projet
- `components` : contient trois composants prédéfinis
  - `HelloWorld.vue`
  - `TheWelcome.vue`
  - `WelcomeItem.vue` : composant référencé par `TheWelcome.vue`
- `main.js` (ou `main.ts`) : fichier référencé par `index.html` et permettant de charger l'application **Vue.js** dans l'élément ayant l'identifiant `app`
- `App.vue` : fichier référencé par `main.js` et utilisant les deux composants `HelloWorld.vue` et `TheWelcome.vue`

# Vue.js

## Contenu d'`index.html`

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="UTF-8" />
    <link rel="icon" href="/favicon.ico" />
    <meta name="viewport" content="width=device-width, initial-scale
      =1.0" />
    <title>Vite App</title>
  </head>
  <body>
    <div id="app"></div>
    <script type="module" src="/src/main.js"></script>
  </body>
</html>
```

# Vue.js

## Contenu de `src/main.js`

```
import { createApp } from 'vue'
import './style.css'
import App from './App.vue'

createApp(App).mount('#app')
```

# Vue.js

## Pourquoi importer `style.css` dans `main.js` ?

- `index.html` sert uniquement de **point d'entrée** analysé par Vite.
- `main.js` devient le **vrai point de départ du bundle**.
- Importer le **CSS** dans `main.js` permet :
  - d'intégrer le style au système de **ES modules**
  - de bénéficier des optimisations de Vite (minification, cache...).
- Ajouter le **CSS** dans `index.html` (`<link>`) contourne le build et est déconseillé.

# Vue.js

## Résumé

Approche	Fichier	Recommandée ?
<code>&lt;link rel="stylesheet" ...&gt;</code>	<code>index.html</code>	Non
<code>import './style.css'</code>	<code>main.js</code>	Oui

**Vite gère le CSS comme une dépendance du projet.**

# Vue.js

**Pour installer Vue/CLI, exécuter la commande**

```
npm install -g @vue/cli
```

© Achref EL MOUELHI ©

# Vue.js

**Pour installer Vue/CLI, exécuter la commande**

```
npm install -g @vue/cli
```

**Pour vérifier la version d'Vue/CLI installée, exécuter la commande**

```
vue -version
```

# Vue.js

**Pour installer Vue/CLI, exécuter la commande**

```
npm install -g @vue/cli
```

**Pour vérifier la version d'Vue/CLI installée, exécuter la commande**

```
vue -version
```

**Ou**

```
vue -V
```



# Vue.js

## Pour créer un nouveau projet Vue.js avec CLI

```
>vue create vue-cli
```

```
Vue CLI v5.0.8 ? Please pick a preset: (Use arrow keys)
```

```
> Default ([Vue 3] babel, eslint)
```

```
Default ([Vue 2] babel, eslint)
```

```
Manually select features
```

```
Running completion hooks...
```

```
Generating README.md...
```

```
Successfully created project cours-vue-cli.
```

```
Get started with the following commands:
```

```
$ cd vue-cli
```

```
$ npm run serve
```

Ensuite, placez vous dans le dossier `cours-vue-cli`

```
cd vue-cli
```

© Achref EL MOUELHI ©

Ensuite, placez vous dans le dossier `cours-vue-cli`

```
cd vue-cli
```

Ouvrez le dossier `cours-vue-cli` avec VSC

```
code .
```

© Achref EL MOU

Ensuite, placez vous dans le dossier `cours-vue-cli`

```
cd vue-cli
```

Ouvrez le dossier `cours-vue-cli` avec VSC

```
code .
```

Pour lancer l'application, exécutez

```
npm run serve
```

Ensuite, placez vous dans le dossier `cours-vue-cli`

```
cd vue-cli
```

Ouvrez le dossier `cours-vue-cli` avec VSC

```
code .
```

Pour lancer l'application, exécutez

```
npm run serve
```

Ou

```
vue serve
```

# Vue.js

## Arborescence d'un projet **Vue.js**

- `node_modules` : contenant les fichiers **Node.js** nécessaires pour un projet **Vue.js**
- `src` : contenant les fichiers sources de l'application
- `package.json` : contenant l'ensemble de dépendance de l'application
- `public` : contenant le point d'entrée de l'application `index.html`
- `vue.config.js` : fichier chargeant les services de **Vue-CLI**
- `tsconfig.json` : fichier de configuration principal pour **TypeScript** (Si vous avez choisi **TypeScript** comme langage)
- `README.md` : permet de fournir une vue d'ensemble de l'application, décrire son objectif et expliquer les fonctionnalités principales.

# Vue.js

## Que contient `src` ?

- `assets` : unique dossier accessible aux visiteurs et contenant logo, CSS, images, sons...
- `components` : contient un composant `HelloWorld.vue`
- `main.js` (ou `main.ts`) : point d'entrée de **Webpack** permettant de charger l'application **Vue.js** dans l'élément ayant l'identifiant `app`
- `App.vue` : fichier référencé par `main.js` et utilisant le composant `HelloWorld.vue`

# Vue.js

## Question

Où faut-il placer les fichiers statiques d'une application **Vue.js**, dans `public` ou dans `assets` ?

© Achref EL MOUELHI



# Vue.js

## Question

Où faut-il placer les fichiers statiques d'une application **Vue.js**, dans `public` ou dans `assets` ?

## Explication

- `src/assets` pour les fichiers à intégrer dans le graphe de dépendances : images, feuilles de style... Ils sont traités (compression, renommage, hash) par le bundler.
- `public` pour les fichiers accessibles directement tels quels, sans traitement par le bundler (favicon, robots.txt, images fixes...).

## Réponse

- Placez dans `assets` tout fichier que vous souhaitez qu'il bénéficie des fonctionnalités d'optimisation de **Webpack** (compression, hashage...) lors de la construction (`build`).
- Utilisez `public` pour les fichiers statiques accessibles tels quels après le `build`.

© Achref EL ME

# Vue.js

## Réponse

- Placez dans `assets` tout fichier que vous souhaitez qu'il bénéficie des fonctionnalités d'optimisation de **Webpack** (compression, hashage...) lors de la construction (`build`).
- Utilisez `public` pour les fichiers statiques accessibles tels quels après le `build`.

## Remarques

- Un fichier `logo.png` placé dans `assets` peut être référencé `./assets/logo.png`.
- Un fichier `logo.png` placé dans `public` peut être référencé directement `/logo.png`.

# Vue.js

## Contenu d'`index.html`

```
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="<%= BASE_URL %>favicon.ico">
    <title><%= htmlWebpackPlugin.options.title %></title>
  </head>
  <body>
    <noscript>
      <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work
        properly without JavaScript enabled. Please enable it to continue.</strong>
    </noscript>
    <div id="app"></div>
    <!-- built files will be auto injected -->
  </body>
</html>
```

# Vue.js

## Contenu de src/main.js

```
import { createApp } from 'vue'
import App from './App.vue'

createApp(App).mount('#app')
```

# Vue.js

## Vue avec CLI

- Temps de démarrage plus long (dev et build)
- Basé sur **WebPack** (config plus complexe mais flexible)
- Code généré compatible avec tous les navigateurs anciens/modernes
- Recommandé pour les grands projets

## Vue avec Vite

- Démarrage ultra rapide
- Utilisant **ESBuild** (dev) et **Rollup** (build)
- Code ciblant **uniquement les navigateurs modernes**
- Initialement pensé pour **petits projets**, mais désormais utilisé à grande échelle

# Vue.js

## Remarque

Depuis quelques mois, on recommande de ne plus utiliser **vue-cli** et d'utiliser **Vite**

© Achref EL MOUELHI ©

## Remarque

Depuis quelques mois, on recommande de ne plus utiliser **vue-cli** et d'utiliser **Vite**

**Documentation officielle :** <https://cli.vuejs.org/#getting-started>

### Vue CLI is in Maintenance Mode!

For new projects, please use create-vue to scaffold Vite-based projects. Also refer to the Vue 3 Tooling Guide for the latest recommendations.



# Vue.js

## Remarque

Depuis quelques mois, on recommande de ne plus utiliser **vue-cli** et d'utiliser **Vite**

**Documentation officielle :** <https://cli.vuejs.org/#getting-started>

### Vue CLI is in Maintenance Mode!

For new projects, please use create-vue to scaffold Vite-based projects. Also refer to the Vue 3 Tooling Guide for the latest recommendations.

## Pour migrer de **vue-cli** vers **Vite**

[https://vueschool.io/articles/vuejs-tutorials/  
how-to-migrate-from-vue-cli-to-vite/](https://vueschool.io/articles/vuejs-tutorials/how-to-migrate-from-vue-cli-to-vite/)

# Vue.js

## Pour désinstaller Vue.js

```
npm uninstall vue-cli -g
```

© Achref EL MOUELHI ©

# Vue.js

## Pour désinstaller Vue.js

```
npm uninstall vue-cli -g
```

## Pour vider le cache (Avant la version 5 de npm)

```
npm cache clean
```

# Vue.js

## Pour désinstaller Vue.js

```
npm uninstall vue-cli -g
```

## Pour vider le cache (Avant la version 5 de npm)

```
npm cache clean
```

## Pour vider le cache (depuis la version 5 de npm)

```
npm cache verify
```

# Vue.js

**Pour mettre à jour la version de Vue.js**

```
npm update -g @vue/cli
```