

# 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 Environment Development
- 3 Vue Devtools
- 4 Solution avec CDN
- 5 Solution avec NPM
  - Installation
  - Création et structure d'un projet créé avec NPM
- 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
- ...

## Vue.js : inconvénients

- Complexité
- Apprentissage
- ...

# Vue.js

## Vue.js

- Appelé aussi **Vue** (et prononcé **View**)
- Framework **JavaScript** (ou **TypeScript**)
- Open-source
- Créé par **Evan You** (ex salarié de **Google**) en 2014
- Écrit en **JavaScript** et **TypeScript**
- 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.

## 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** **c**ommand **l**ine **i**nterface : 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

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

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

## 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

## 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`

© Achref EL MOUETRI

# Vue.js

## VSC : téléchargement

`code.visualstudio.com/download`

## VSC : extension pour Vue.js

- **Vue.js 2** : Vetur
- **Vue.js 3** : Volar

## 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.js

## Vue Devtools

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

## Solution avec **CDN** : démarche

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

# Vue.js

## Code généré

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

<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">
  <title>Document</title>
</head>

<body>

</body>

</html>
```

# Vue.js

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

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

<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">
  <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="en">

<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">
  <title>Cours Vue.js</title>

</head>

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

</html>
```



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

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

<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">
  <title>Cours Vue.js</title>

</head>

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

  <script src="https://unpkg.com/vue@next"></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`

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

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

# 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 installer Vue.js, exécuter la commande

```
npm install vue -g
```

# Vue.js

## Remarque

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

Pour installer Vue.js, exécuter la commande

```
npm install vue -g
```

Pour installer une version bien précise (par exemple 2.1.0)

```
npm install -g vue@2.1.0
```



# Vue.js

Pour créer un nouveau projet Vue.js avec npm

```
>npm init vue@latest

Vue.js - The Progressive JavaScript Framework

✓ Project name: ... cours-vue-npm
✓ 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 Cypress for both Unit and End-to-End testing? ... No / Yes
✓ Add ESLint for code quality? ... No / Yes

Scaffolding project in <chemin-projet>

Done. Now run:

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

# Vue.js

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

```
npm install
```

© Achref EL ME

# Vue.js

Ouvrez le dossier `cours-vue-npm` 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 nécessaires de la librairie **Node.js** 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** (outil de construction, build, de projet **Vue.js**)
- `index.html` : point d'entrée de l'application

# 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` : 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 App from './App.vue'

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

# 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 cours-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 cours-vue-cli
```

```
$ npm run serve
```

# Vue.js

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

```
cd cours-vue-cli
```

© Achref EL MOUELHI ©

# Vue.js

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

```
cd cours-vue-cli
```

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

```
code .
```

# Vue.js

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

```
cd cours-vue-cli
```

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

```
code .
```

Pour lancer l'application, exécutez

```
npm run serve
```

# Vue.js

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

- `node_modules` : contenant les fichiers nécessaires de la librairie **Node.js** 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**

# 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` : 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

## Question

Où placer les fichiers statiques, dans `public` ou `assets` ?

© Achref EL MOUELHI ©

# Vue.js

## Question

Où placer les fichiers statiques, dans `public` ou `assets` ?

## Explication

- `assets` : contenu géré par **Webpack** : tout fichier placé dans `assets` sera ajouté dans le graphe de dépendance de **Webpack**. On utilise le chemin relatif pour référencer (utiliser le `.`).
- `public` : contenu non géré par **Webpack** : tout fichier placé dans `public` sera copié et ne passera pas par Webpack. On utilise le chemin absolu pour le référencer.

# Vue.js

## Réponse

- Placez dans `assets` tout fichier que vous souhaitez qu'il bénéficie des fonctionnalités d'optimisation de **Webpack** lors de la construction (`build`).
- Placez dans `public` tout fichier que vous souhaitez qu'il préserve son nom après le `build`.

# 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')
```

# Angular

## Vue avec CLI

- Moins rapide
- Utilisant **WebPack**
- Code généré compatible avec tous les navigateurs anciens/modernes
- Recommandé pour les grands projets

## Vue avec Vite (npm)

- Plus rapide
- Utilisant **Rollup**
- Code généré compatible avec uniquement les navigateurs récents
- Recommandé pour les petits projets

# 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
```