



# Package HighCharts

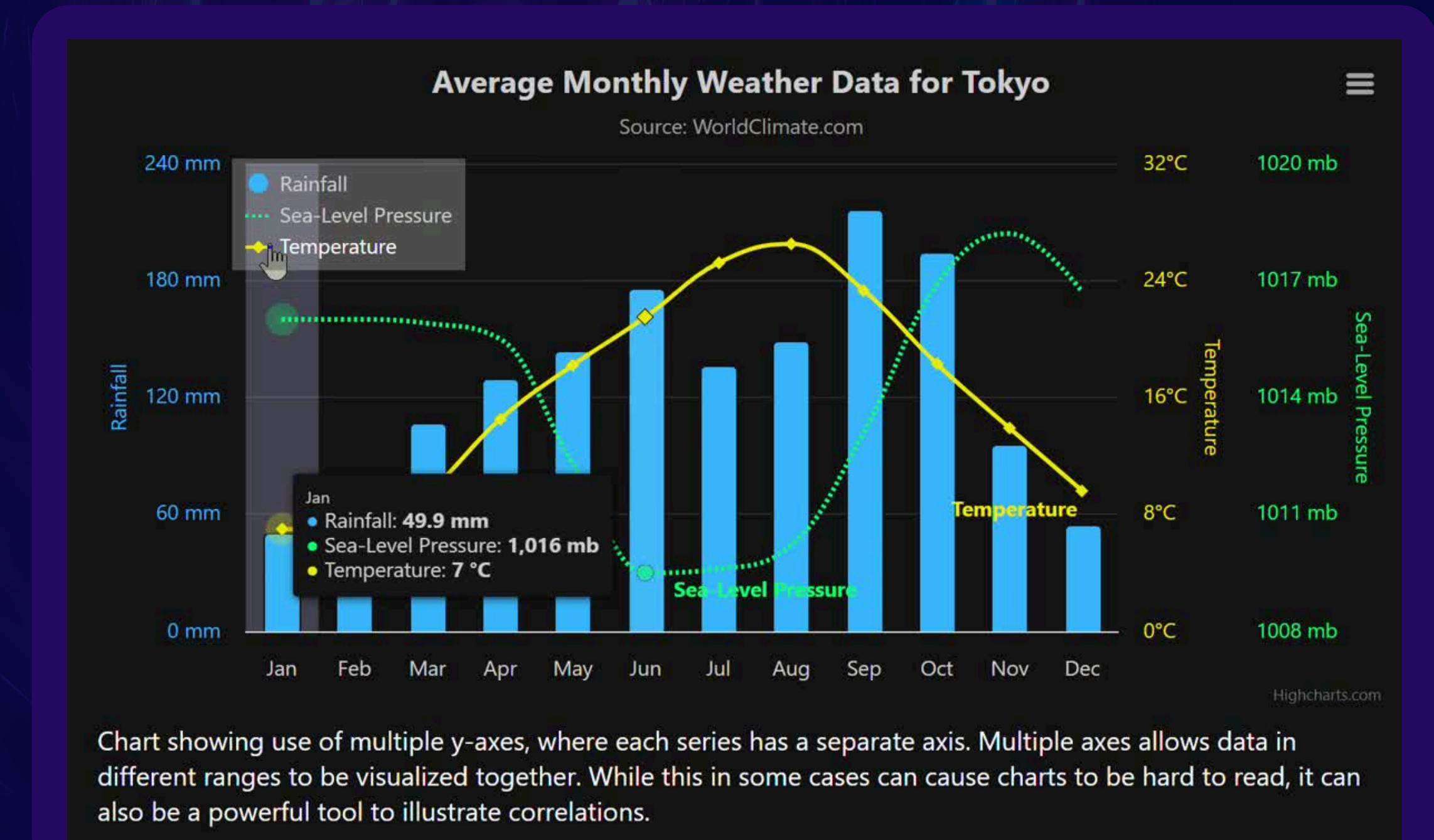
Vue.js

[Lien du Canva \(avec vidéos fonctionnelles\)](#)

Formation : Master 2 MIASHS, parcours IC

Module : Interaction Homme Machine 2

Étudiants : Arbaut Jean-Baptiste et Tropel Célia



# SOMMAIRE

---

## 1 - Introduction du sujet :

- 1.1 - Highcharts & Vue.js : Contexte d'utilisation
- 1.2 - Vue.js : Introduction rapide

## 2 - Présentation de Highcharts (utilisation avec Vue.js) :

- 2.1 - Présentation Sommaire
- 2.2 - Licences et produits
- 2.3 - Exemples de graphiques
- 2.4 - Tableau comparatif
- 2.5 - Avantages et Inconvénients
- 2.6 - Téléchargements avec NPM
- 2.7 - Architecture d'une app
- 2.8 - Configuration de Highcharts avec Vue.js

## 3 - Activité

## 4 - Sources



# 1 - INTRODUCTION DU SUJET



# 1.1 - HIGHCHARTS & VUE.JS : CONTEXTE D'UTILISATION



Pour cette présentation nous allons utiliser Highcharts avec Vue.js, pourquoi ?

- Parce que Vue.js est un framework JavaScript simple, moderne pour la création d'interfaces interactives.

Highcharts est une bibliothèque multiplateforme, elle peut être intégrée dans un grand nombre d'environnements :

- React
- Angular
- Vue.js
- Svelte
- TypeScript
- JavaScript “pur”
- Autres intégrations possibles :
  - .NET, Java, Python, PHP, Django, Node.js
  - iOS, Android
  - R via le package highcharter



# 1.2 - VUE.JS : INTRODUCTION RAPIDE

## Qu'est-ce que Vue.js ?

- Un **framework JavaScript** qui s'appuie aussi sur **HTML et CSS** permettant :
  - La création d'interfaces utilisateur
  - Les Single Page Applications
  - La génération de sites statiques
  - Des rendus multi-supports : desktop, mobile, WebGL
- Projet libre et open source distribué sous la licence MIT.
- Utilisé par de nombreuses entreprises notamment la Fondation Wikimedia, la NASA, Apple, Google, Microsoft, GitLab, Zoom...

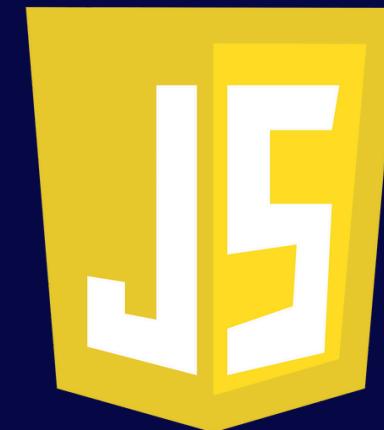
```
App.vue +  
1 <script setup>  
2 import { ref } from 'vue'  
3  
4 const msg = ref('Hello World!')  
5 </script>  
6  
7 <template>  
8   <h1>{{ msg }}</h1>  
9   <input v-model="msg" />  
10 </template>  
11
```

PREVIEW JS CSS SSR

Hello World!

Hello World!

JavaScript



HTML



&  
CSS



Les composants Vue peuvent être créés selon deux styles d'API différents :

- L'**API Options** : l'objectif est de définir la logique d'un composant en utilisant un objet (ex: data, methods).
- L'**API Composition** : l'objectif est de définir la logique d'un composant à l'aide de fonctions API importées.

# 2 - PRÉSENTATION DE HIGHCHARTS (UTILISATION AVEC VUE.JS)



# 2.1 - PRÉSENTATION SOMMAIRE



## Qu'est-ce que Highcharts ?

Une bibliothèque JavaScript de datavisualisation très complète :

- Développée par la société Highsoft (Norvège).
- Fonctionne sur : Chrome, Firefox, Safari, Edge...
- Idéale pour :
  - Dashboards
  - Rendus professionnels
  - Statistiques en temps réel
  - Graphiques de projet (Gantt, timelines)
- Plus de 90 types de graphiques disponibles.
- Versions :
  - Première release : 2009.
  - Actuelle : 12.4.0, sortie en septembre 2025.

## 2.2 - LICENCES ET PRODUITS

Highcharts est sous licence propriétaire :

- Gratuit pour les usages personnels / non commerciaux (version “Creative Commons”).
- Payant pour les usages commerciaux :
  - Internal License : pour des applications internes d’entreprises.
  - SaaS License / SaaS+ : en cas de vente d’un service en ligne.
  - OEM License : en cas de revente un produit qui embarque Highcharts (logiciel, app, etc.).



L’écosystème :



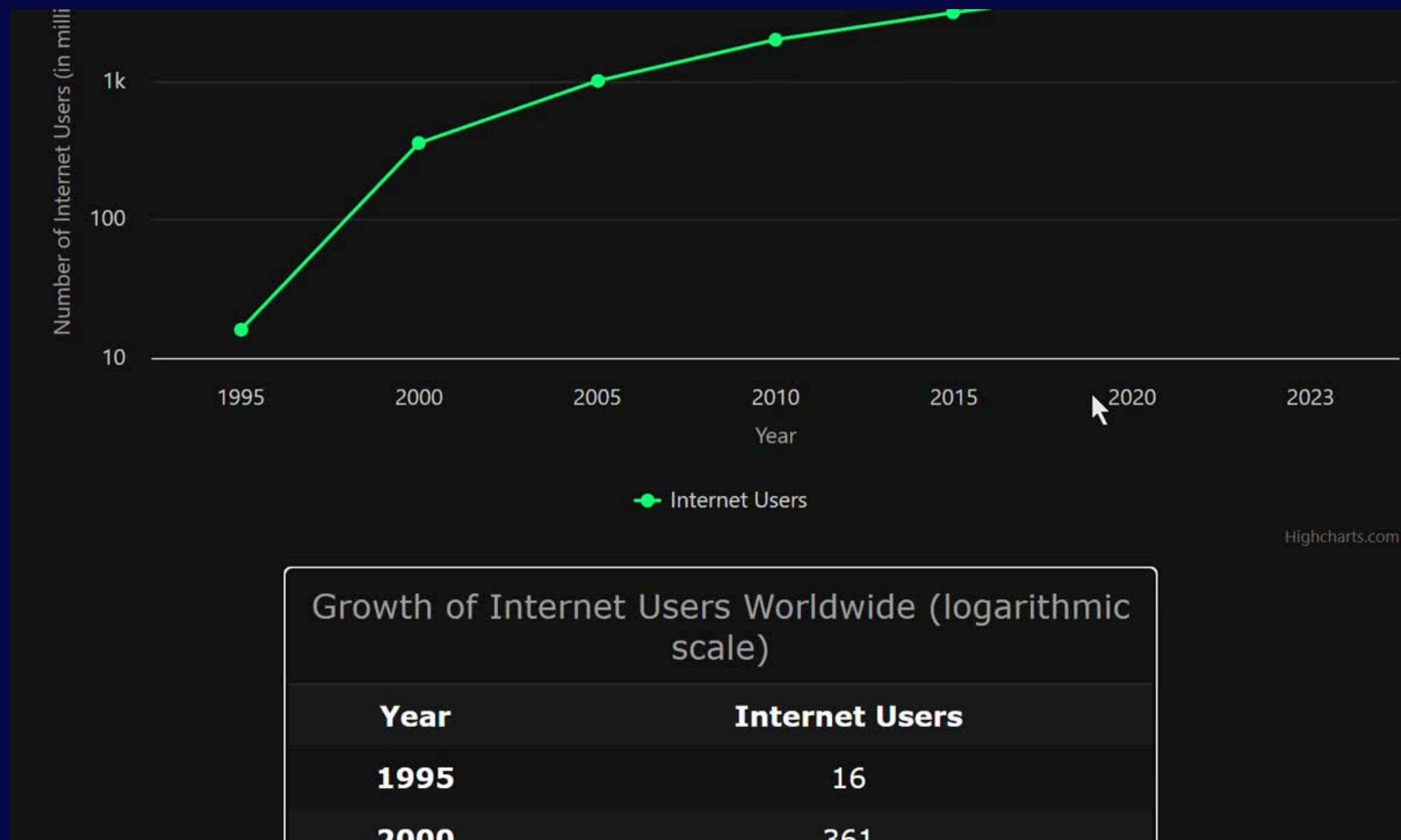
Produit	Utilisation
<b>Highcharts Core (et More)</b>	Graphiques classiques (line, bar, area, pie...)
<b>Highstock</b>	Séries temporelles et graphiques financiers
<b>Highmaps</b>	Cartes géographiques
<b>Highcharts Gantt</b>	Diagrammes de Gantt et gestion de projets

*Chaque module peut nécessiter une licence commerciale spécifique.*

## 2.3 - EXEMPLES DE GRAPHIQUES



### 1 - Graphiques classiques (line, bar, area, pie...):



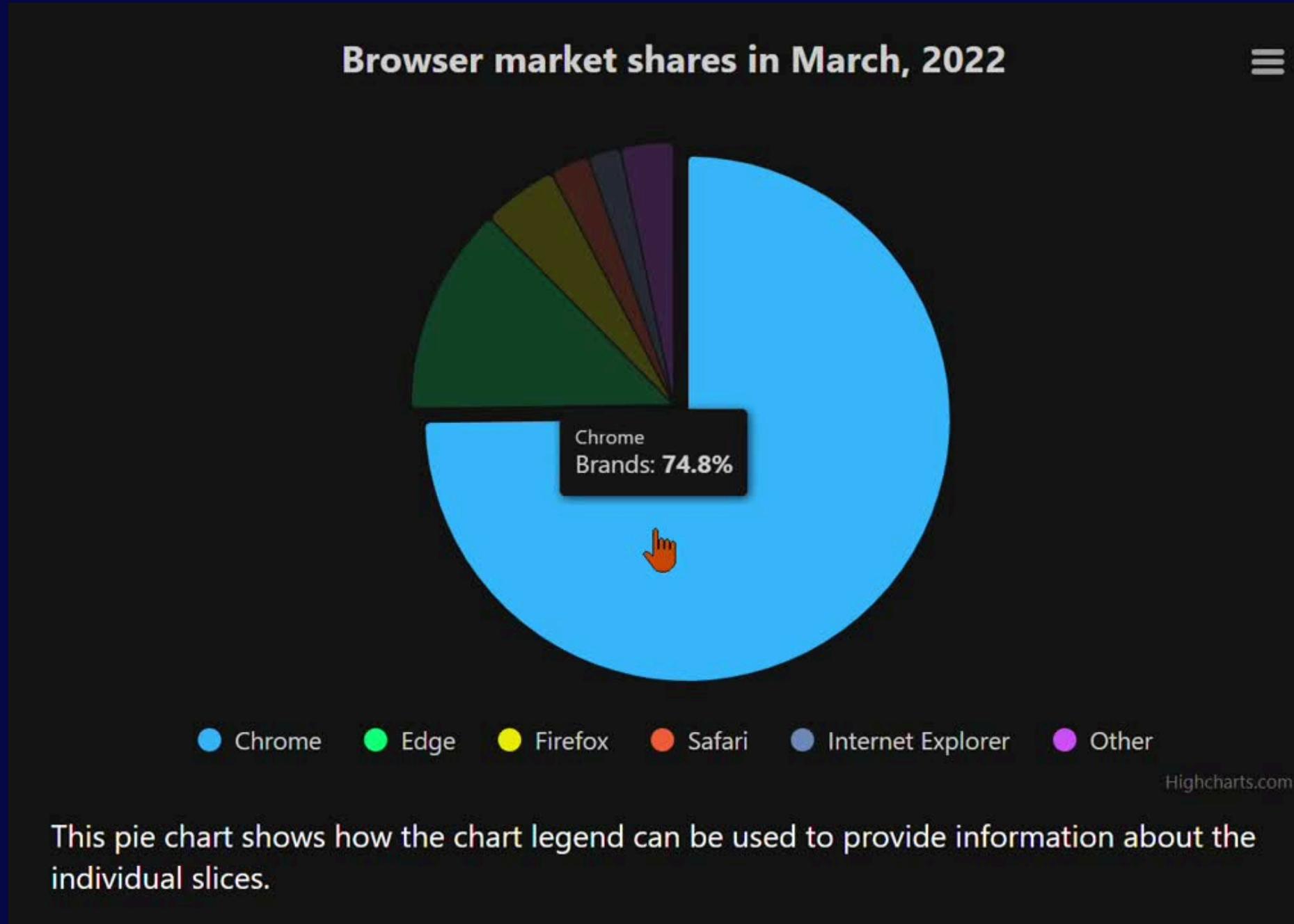
JS

```
1 Highcharts.chart('container', {
2   title: {
3     text: 'Growth of Internet Users Worldwide (logarithmic scale)'
4   },
5 
6   accessibility: {
7     point: {
8       valueDescriptionFormat:
9         '{xDescription}{separator}{value} million(s)'
10    }
11  },
12 
13   xAxis: {
14     title: {
15       text: 'Year'
16     },
17     categories: [1995, 2000, 2005, 2010, 2015, 2020, 2023]
18   },
19 
20   yAxis: {
21     type: 'logarithmic',
22     title: {
23       text: 'Number of Internet Users (in millions)'
24     }
25   },
26 
27   tooltip: {
28     headerFormat: '{series.name}<br />',
29     pointFormat: '{point.y} million(s)'
30   },
31 
32   series: [
33     {
34       name: 'Internet Users',
35       data: [16, 361, 1018, 2025, 3192, 4673, 5200],
36       color: 'var(--highcharts-color-1, #2ca02c)'
37     }
38   ];
39});
```

## 2.3 - EXEMPLES DE GRAPHIQUES



### 1 - Graphiques classiques (line, bar, area, pie...):



JS

```
1 // Data retrieved from https://netmarketshare.com/
2 // Build the chart
3 Highcharts.chart('container', {
4   chart: {
5     plotBackgroundColor: null,
6     plotBorderWidth: null,
7     plotShadow: false,
8     type: 'pie'
9   },
10   title: {text: 'Browser market shares in March, 2022'},
11   tooltip: {pointFormat: '{series.name}: <b>{point.percentage:.1f}%</b>'},
12   accessibility: {
13     point: {valueSuffix: '%'}
14   },
15   plotOptions: {
16     pie: {
17       allowPointSelect: true,
18       cursor: 'pointer',
19       dataLabels: {enabled: false},
20       showInLegend: true
21     }
22   },
23   series: [
24     {
25       name: 'Brands',
26       colorByPoint: true,
27       data: [
28         {name: 'Chrome', y: 74.77, sliced: true, selected: true},
29         {name: 'Edge', y: 12.82}, {name: 'Firefox', y: 4.63},
30         {name: 'Safari', y: 2.44}, {name: 'Internet Explorer', y: 2.02},
31         {name: 'Other', y: 3.28}
32       ]
33     });
34 }
```

## 2.3 - EXEMPLES DE GRAPHIQUES



### 2 - Séries temporelles et graphiques financiers :

Utilisé pour :

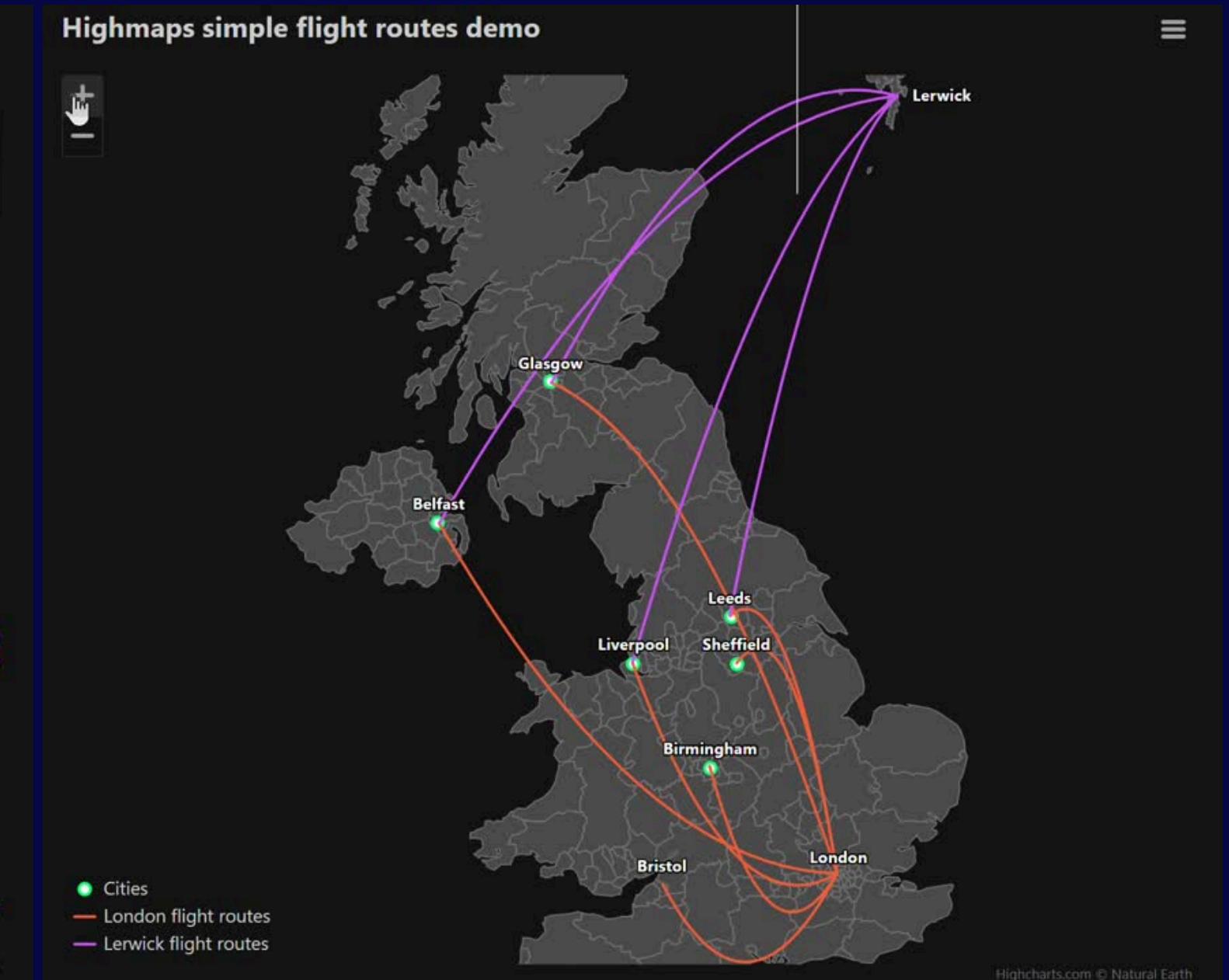
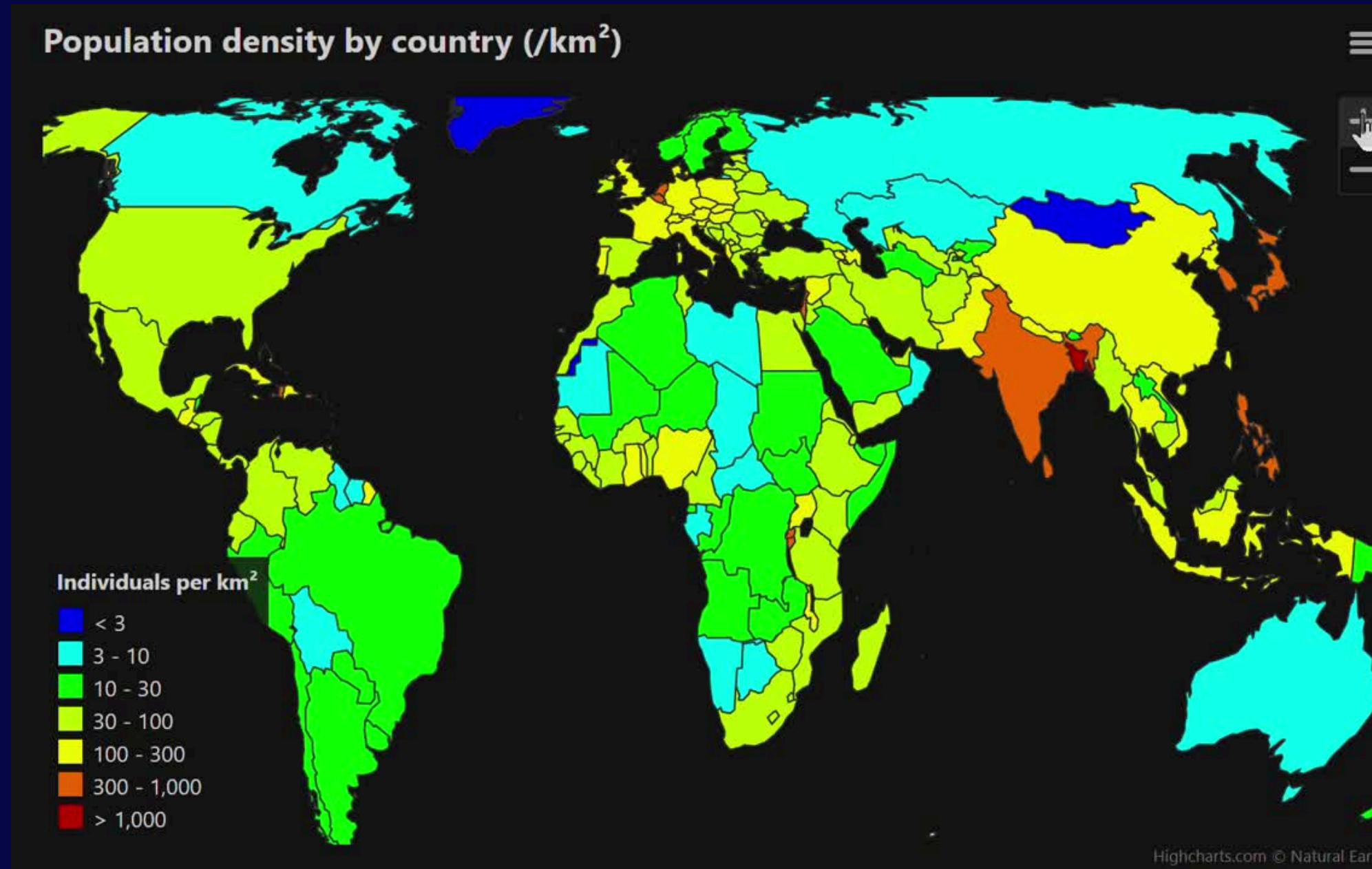
- Des données en direct.
- Des données MAJ en continu.
- Des volumes de données importants.
- Des graphiques spécialisés (finances, scientifiques...).



## 2.3 - EXEMPLES DE GRAPHIQUES



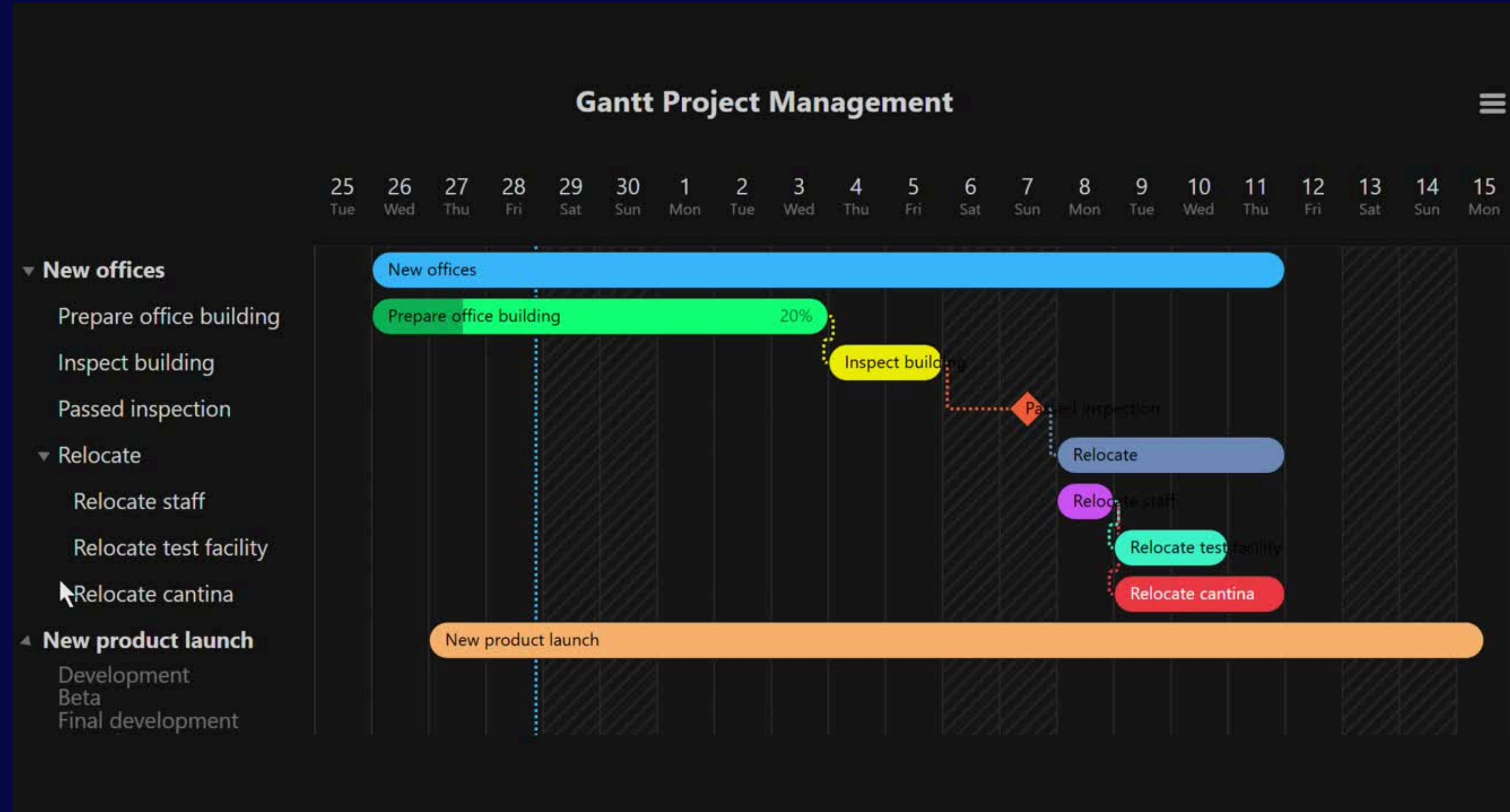
### 3 - Cartes géographiques :



## 2.3 - EXEMPLES DE GRAPHIQUES



### 4 - Diagrammes de Gantt et gestion de projets :

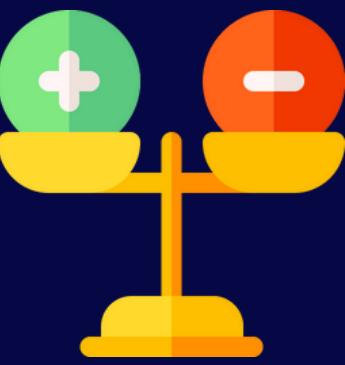




## 2.4 - TABLEAU COMPARATIF

Critères	Highcharts	Chart.js	ECharts	D3.js
<b>Licence gratuite</b>	Oui (non commercial)	Oui (MIT)	Oui (Apache 2.0)	Oui (BSD)
<b>Difficulté de prise en main d'après les utilisateurs</b>	Moyenne car très documenté	Facile	Facile à Moyenne	Difficile
<b>Qualité visuelle</b>	Professionnelle	Standard - Moyenne	Haute et Moderne	Moyenne mais customisable
<b>Types de graphiques disponibles</b>	Plus de 90	Environ 10	Très large + 3D	Illimité (à coder)
<b>Graphiques classiques (line, bar, pie, area...)</b>	Oui	Oui	Oui	Oui mais à coder
<b>Graphiques financiers</b>	Oui (Highstock)	Non	Moyen	Oui mais à coder
<b>Graphiques de cartes</b>	Oui (Highmaps)	Non	Oui	Oui mais complexe
<b>Graphiques Gantt</b>	Oui (Highcharts Gantt)	Non	Non	Oui mais complexe
<b>Intégration Vue.js</b>	Oui (highcharts-vue)	Oui	Oui	Possible mais difficile
<b>Intégration React</b>	Oui	Oui	Oui	Possible mais difficile
<b>Intégration Angular</b>	Oui	Oui	Oui	Possible mais difficile

## 2.5 - AVANTAGES ET INCONVÉNIENTS



### Avantages :

- Solution de **qualité professionnelle** adaptée aux dashboards et applications métier.
- **Écosystème complet** (graphiques, cartes, financiers, Gantt).
- Très bonne **interactivité** (zoom, exploration, animations fluides).
- **Documentation claire** et nombreux exemples.
- Serveur discord de discussion, HighCharts GPT.



### Inconvénients :

- Licence **commerciale payante**, ce qui peut limiter son usage.
- Moins flexible pour les **visualisations entièrement sur-mesure** (par rapport à D3.js).
- Configuration parfois **dense**, surtout pour les utilisateurs débutants.





## 2.6 - TÉLÉCHARGEMENTS AVEC NPM



*Il faut créer au préalable un projet, par ex un projet Vue.js :*

- *npm create vue@latest*
- *npm install dans le projet.*

Télécharger Highcharts :

**npm install highcharts**

Télécharger le plugin de Vue :

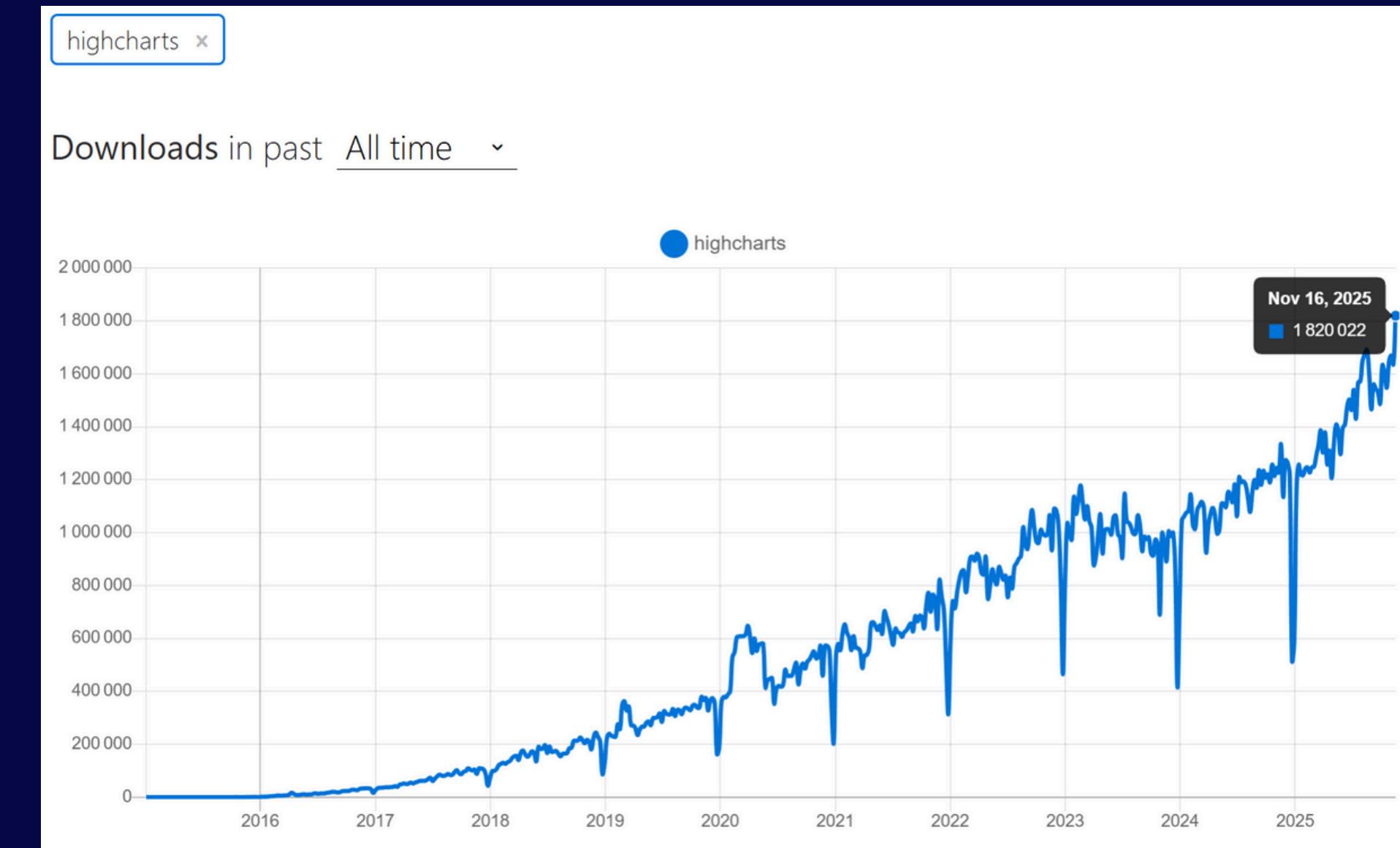
**npm install highcharts-vue**

Télécharger d'autres modules :

**npm install @highcharts/map-collection**

**npm install highcharts-gantt**

À titre indicatif, voici l'évolution des téléchargements du package Highcharts avec NPM :



*Régulièrement dans le top 3 des librairies de datavisualisation professionnelles.*

## 2.7 - ARCHITECTURE D'UNE APP



Voici un exemple d'architecture simple d'une app Vue. JS utilisant Highcharts :

```
src/
  └── components/
    └── Chart.vue          # Composant qui encapsule Highcharts
  └── composables/
    └── useSalesData.js    # Récupération + transformation des données
  └── views/
    └── SalesDashboard.vue # UI + filtres + chart options
  └── main.js              # import highcharts + highcharts-vue
public/
  └── index.html           # Conteneur <div id="app">
api/
  └── /api/sales           # Backend (optionnel)
```

- **Composant graphique** : un composant Vue dédié encapsule Highcharts et reçoit les options de la vue.
- **Gestion des données** : centralise la récupération et la transformation des données.
- **Vue Tableau de bord** : importe le graphique et paramétrage des filtres / options.
- **Initialisation** : Highcharts et son plugin Vue sont déclarés globalement dans main.js.

# 2.8 - CONFIGURATION DE HIGHCHARTS AVEC VUE.JS

## 1 - Cration d'un graphique dans src/components/ChartColumn.vue :

```
<template>
<div id="chart-box">
  <highcharts :options="chartOptions"></highcharts>
</div>
</template>

<script>
export default {
  name: "ChartColumn",
  data() {
    return {
      chartOptions: {
        chart: { type: 'column', height: 400 },
        title: { text: 'Ventes mensuelles' },
        xAxis: { categories: ['Jan', 'Fv', 'Mar', 'Avr', 'Mai', 'Juin'] },
        yAxis: { title: { text: 'Ventes (k)' } },
        series: [
          { name: 'Produit A', data: [5, 3, 4, 7, 2, 4] }
        ]
      }
    }
  }
}
</script>
```

## 2 - Configuration globale dans src/main.js :

```
import { createApp } from 'vue'
import App from './App.vue'
import HighchartsVue from 'highcharts-vue'
import './assets/style.css'

const app = createApp(App)
app.use(HighchartsVue)
app.mount('#app')
```

## 3 - Optionnel : ajout de fichiers HTML et CSS pour la mise en forme.



css



HTML

&

# 2.8 - CONFIGURATION DE HIGHCHARTS AVEC VUE.JS

4 - Appel du composant dans une page, ici  
src/App.vue :

```
<template>
<div id="app">
<h1 style="text-align:center;">Ventes mensuelles</h1>
<ChartColumn />
</div>
</template>

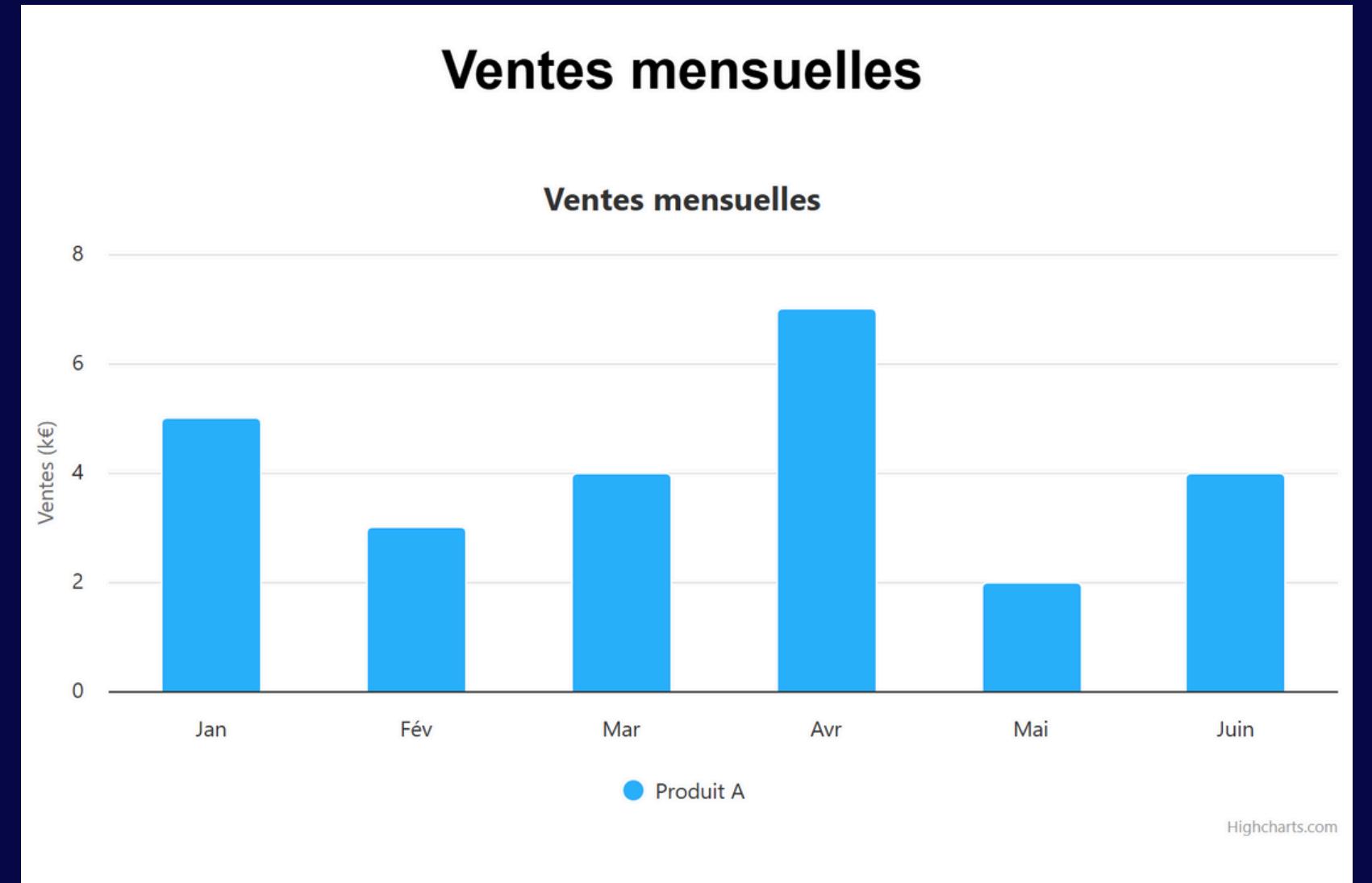
<script>
import ChartColumn from './components/ChartColumn.vue'

export default {
  components: {
    ChartColumn
  }
}
</script>
```

5 - Lancer la commande :

```
npm run dev
```

6 - Résultat :



# 3 - ACTIVITÉ



## 3.1 - ACTIVITÉ



Pour l'activité, nous allons nous concentrer sur l'apprentissage des graphiques Highcharts (et non pas sur l'architecture avec Vue.js). Pour cela, nous avons créé un fichier type “Notebook”, vous permettant de visualiser des graphiques et leurs codes, de les modifier et d'en créer.

Organisation du TP :

- Lire le README.md avant de débuter le TP.
- Le TP se décompose en plusieurs exercices, comportant une consigne, une zone de code, un bouton exécuter et le graphique associé.

Une correction est à disposition.

Lien :

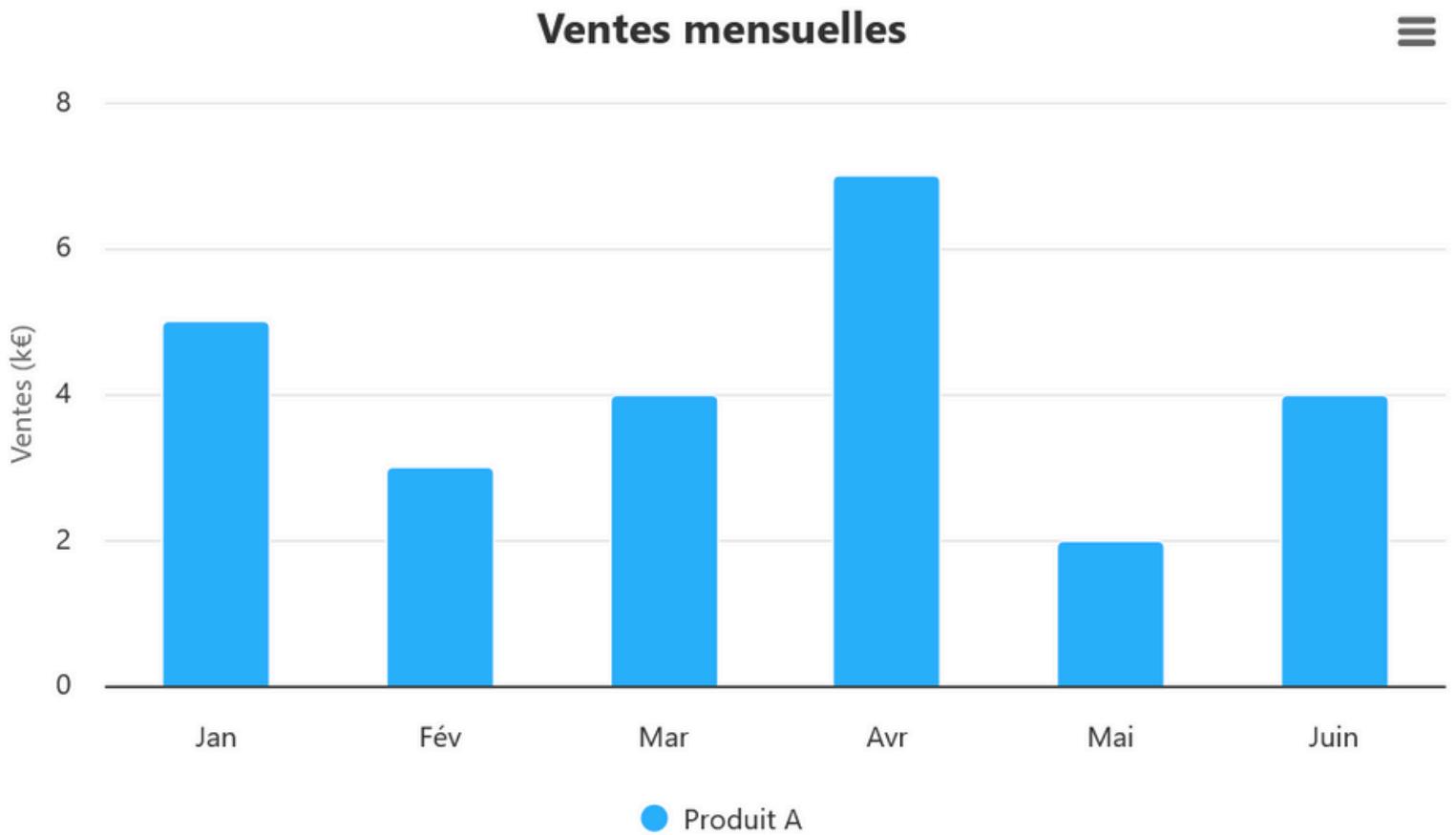
<https://github.com/CTRetu/IHM-Activite-HighCharts>

### Exercice 1 : Graphique en colonnes

Modifie les valeurs pour afficher d'autres ventes mensuelles. + Ajoute un mois supplémentaire.

```
Highcharts.chart('chart1', {  
    chart: { type: 'column' },  
    title: { text: 'Ventes mensuelles' },  
    xAxis: { categories: ['Jan', 'Fév', 'Mar', 'Avr', 'Mai', 'Juin'] },  
    yAxis: { title: { text: 'Ventes (k€)' } },  
    series: [{ name: 'Produit A', data: [5, 3, 4, 7, 2, 4] }]  
});
```

▶ Exécuter



# 4 - SOURCES



## Contenu :

- <https://vuejs.org/>
- <https://fr.wikipedia.org/wiki/Vue.js>
- <https://www.highcharts.com/>
- <https://nptrends.com/highcharts>
- <https://www.chartjs.org/>
- <https://echarts.apache.org/en/index.html>
- <https://d3js.org/>
- <https://apexcharts.com/>

## Illustrations :

- <https://vuejs.org/>
- <https://www.highcharts.com/>
- <https://www.highcharts.com/demo>
- <https://www.flaticon.com/fr/>, auteurs : freepik, Pixel Buddha, Gregor, Creatype, juicy\_fish, kosonicon, Vector Squad, Smashicons, pictogramer.

MERCI DE VOTRE ATTENTION ET  
PARTICIPATION

