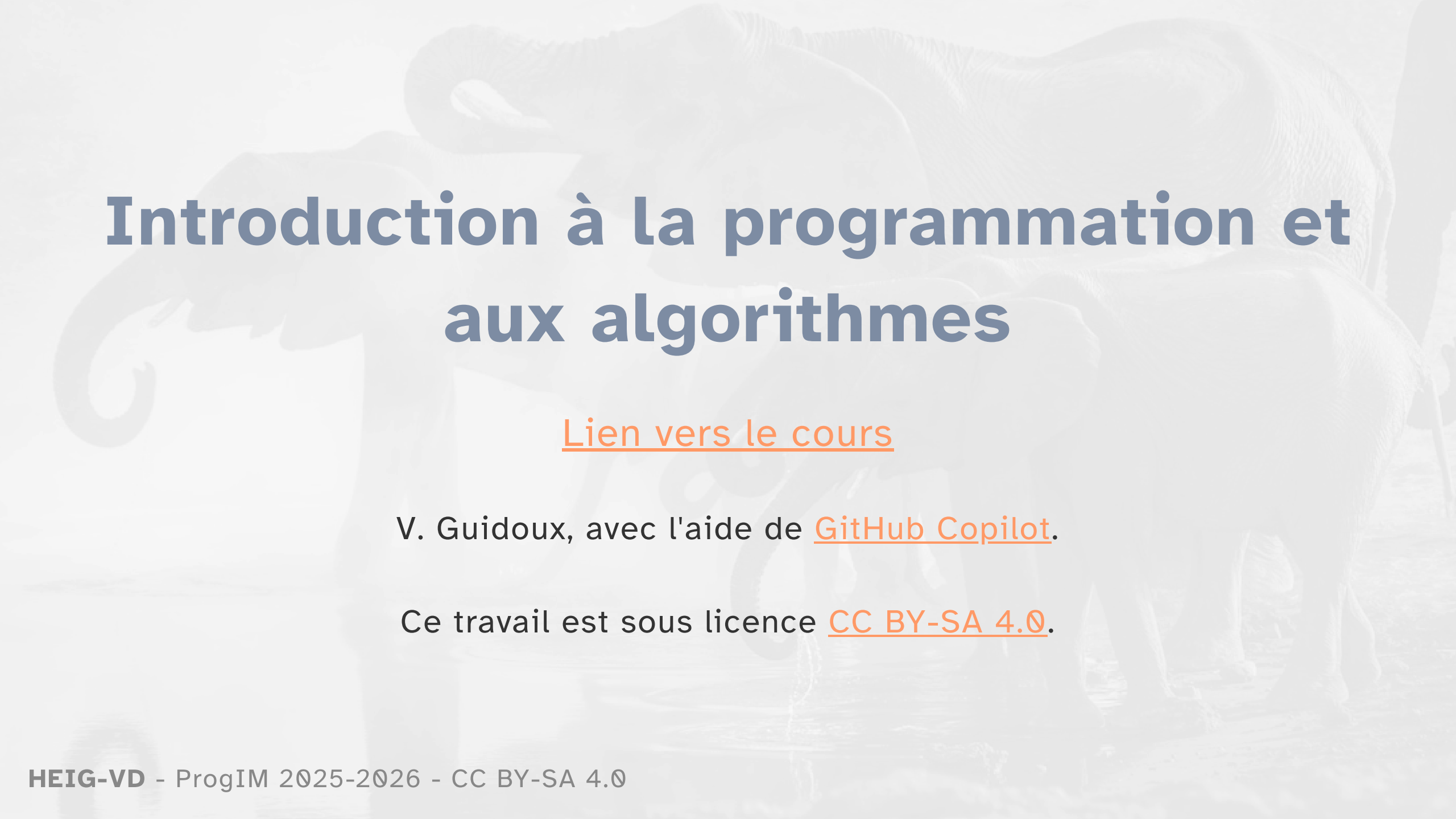


A background image showing a herd of elephants in a savanna landscape. One elephant in the foreground is partially visible, with its trunk and head. In the background, another elephant is standing, and a third one is partially visible on the right. The scene is set in a dry, open landscape with some trees in the distance.

Introduction à la programmation et aux algorithmes

[Lien vers le cours](#)

V. Guidoux, avec l'aide de [GitHub Copilot](#).

Ce travail est sous licence [CC BY-SA 4.0](#).

Retrouvez plus de détails dans le support de cours

Cette présentation est un résumé du support de cours. Pour plus de détails, consultez le [support de cours](#).

Objectifs (1/2)

- Expliquer ce qu'est la programmation à l'aide d'exemples du quotidien
- Comprendre la notion d'algorithme grâce à la métaphore de la recette



Objectifs (2/2)

- Suivre et imaginer des instructions simples pour résoudre un problème
- Prendre conscience de l'importance de la précision dans les instructions



Qu'est-ce que la programmation ?

La programmation, c'est donner des instructions précises à un ordinateur pour qu'il réalise une tâche.

C'est comme expliquer à quelqu'un comment faire quelque chose, étape par étape.



Qu'est-ce qu'un algorithme ?

Un algorithme, c'est une suite d'instructions ordonnées qui permettent de :

- Résoudre un problème
- Atteindre un objectif

On retrouve des algorithmes partout dans la vie quotidienne !



La recette de cuisine : une métaphore

Une recette, c'est un algorithme du quotidien :

1. Prendre un saladier
2. Casser deux œufs dans le saladier
3. Ajouter 100g de sucre
4. Mélanger
5. Verser dans un moule
6. Cuire 30 minutes au four

Comment présenter nos exemples

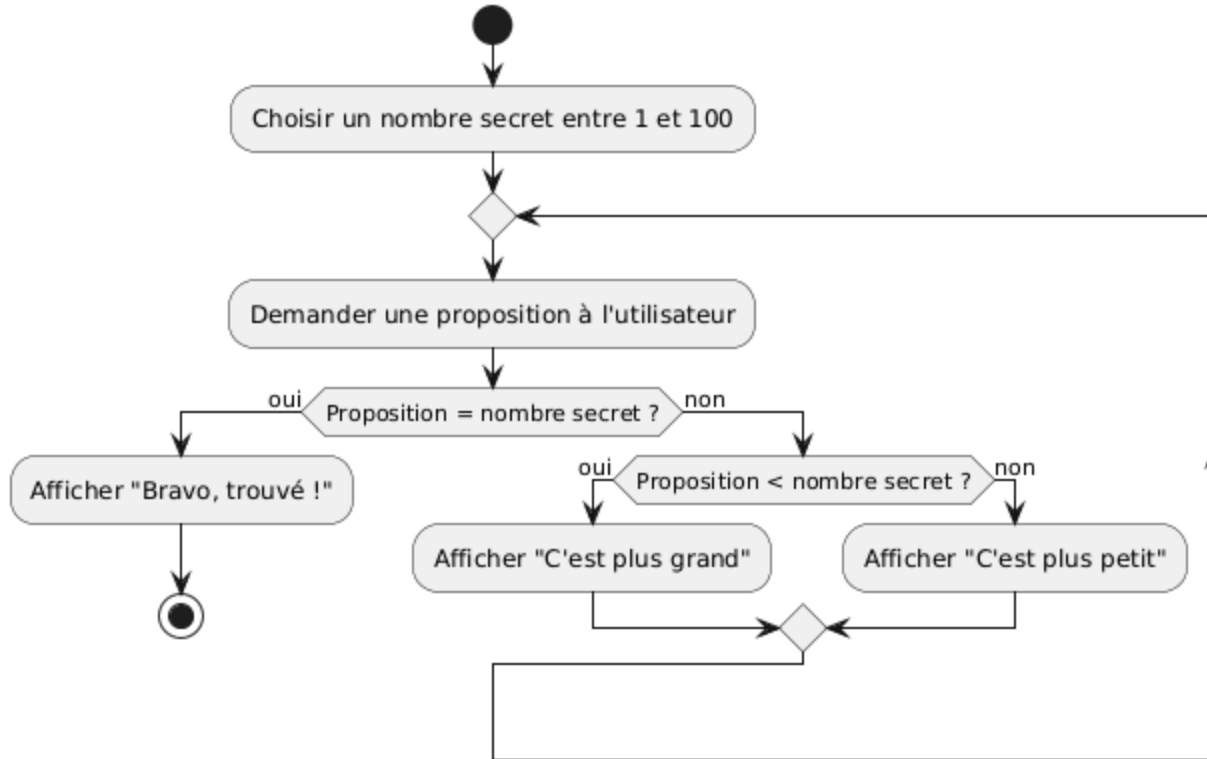
Dans ce cours, chaque concept sera illustré de 3 façons :

1. En français (explication ou pseudo-code)
2. En PlantUML (diagramme d'algorithme)
3. En Java (code source)

En français

L'ordinateur choisit un nombre au hasard entre 1 et 100. L'utilisateur doit deviner ce nombre. Après chaque proposition, l'ordinateur indique si le nombre à deviner est plus grand, plus petit ou égal à la proposition. Le jeu continue jusqu'à ce que l'utilisateur trouve le bon nombre.

avec PlantUML



```
public class GuessNumber {  
    public static void main(String[] args) {  
        Random rand = new Random();  
        int secret = rand.nextInt(100) + 1;  
        Scanner scanner = new Scanner(System.in);  
        int guess = 0;  
  
        System.out.println("Devine le nombre entre 1 et 100 :");  
        while (guess != secret) {  
            System.out.print("Ta proposition : ");  
            guess = scanner.nextInt();  
            if (guess < secret) {  
                System.out.println("C'est plus grand.");  
            } else if (guess > secret) {  
                System.out.println("C'est plus petit.");  
            } else {  
                System.out.println("Bravo, tu as trouvé !");  
            }  
        }  
    }  
}
```

Questions

Est-ce que vous avez des questions ?

À vous de jouer !

- (Re)lire le support de cours.
- Poser des questions si nécessaire.

➡ [Lien vers le cours](#)

N'hésitez pas à vous entraidez si vous avez des difficultés !



Sources (1/3)

- [Illustration principale](#) par [Richard Jacobs](#) sur [Unsplash](#)
- [Illustration](#) par [Aline de Nadai](#) sur [Unsplash](#)
- [Illustration](#) par [Nguyen Dang Hoang Nhu](#) sur [Unsplash](#)
- [Illustration](#) par [Samantha Fortney](#) sur [Unsplash](#)
- [Illustration](#) par [Tim van Cleef](#) sur [Unsplash](#)
- [Illustration](#) par [Alec Favale](#) sur [Unsplash](#)
- [Illustration](#) par [Samuel Girven](#) sur [Unsplash](#)
- [Illustration](#) par [Nikita Kachanovsky](#) sur [Unsplash](#)