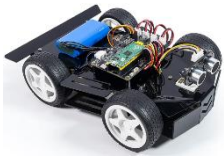


## 1. Mise en situation



Il est temps de mettre tout en œuvre pour faire fonctionner votre robot. Pour le faire avancer et le piloté nous allons devoir le programmer.

## 2. Définition

L'**algorithmique** est la science des algorithmes.

Un **algorithme** est une suite finie et non-ambiguë d'instructions qui indique la démarche à suivre pour résoudre un problème.

Un **algorithme** est une représentation graphique normalisée d'un algorithme. Il permet la réalisation d'un algorithme.

Le **pseudo-code** est une représentation textuelle normalisée d'un algorithme. C'est l'ensemble de caractères, de symboles et de règles constituant le **langage algorithmique**. Il permet la réalisation d'un algorithme.

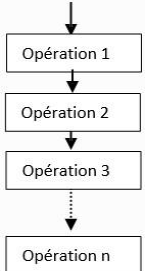
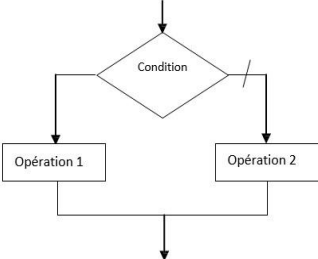
Les algorithmes sont des **méthodes** applicables par des ordinateurs pour résoudre différents problèmes. Leur **implémentation** dans tel ou tel **langage informatique** particulier est un **programme informatique**.

## 3. La structure d'un algorithme

Un algorithme est composé de :

- Structures de données : variables, tableaux, listes,...
- Structures algorithmiques.

Le tableau ci-après regroupe les principales structures algorithmiques.

Structure	Algorithme	Pseudo-code	Langage Python
<p><b>La structure linéaire</b></p> <p>La structure linéaire se caractérise par une suite d'actions à exécuter successivement dans l'ordre énoncé.</p>		<p><b>FAIRE</b></p> <p>Opération 1 ;</p> <p>Opération 2 ;</p> <p>Opération 3 ;</p> <p> </p> <p>Opération n ;</p> <p><b>FIN FAIRE</b></p>	<p>Opération 1</p> <p>Opération 2</p> <p>Opération 3</p> <p>Opération n</p>
<p><b>La structure alternative</b></p> <p>« SI...ALORS...SINON »</p> <p>La structure alternative n'offre que deux issues possibles à la poursuite de l'algorithme en s'excluant mutuellement.</p>		<p><b>SI</b> Condition <b>ALORS</b></p> <p>    <b>FAIRE</b> Opération 1 ;</p> <p><b>SINON</b></p> <p>    <b>FAIRE</b> Opération 2 ;</p> <p><b>FIN SI</b></p>	<p><b>if</b> (Condition) :</p> <p>    Opération1</p> <p><b>else :</b></p> <p>    Opération2</p>

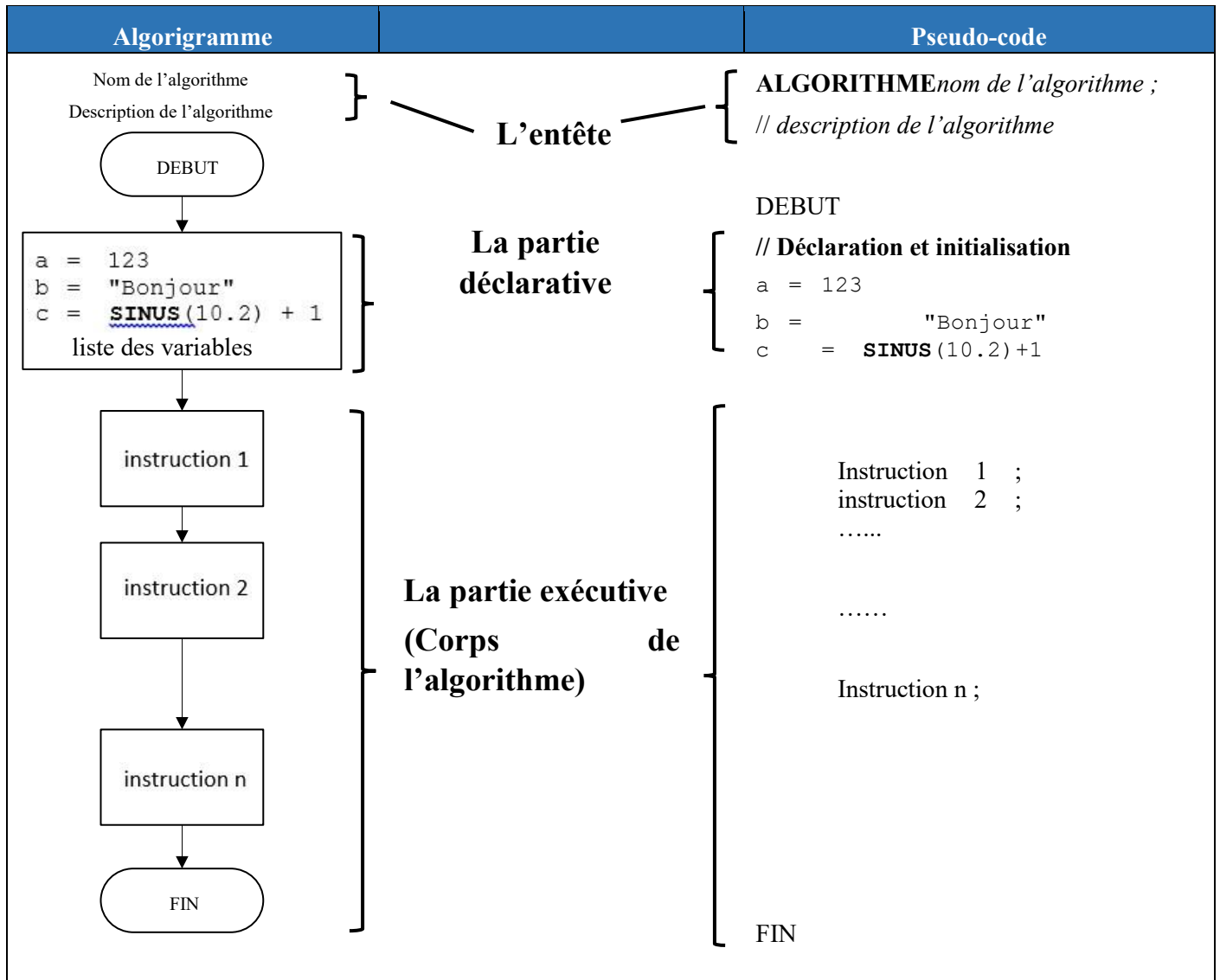


Structure	Algorithme	Pseudo-code	
<p><b>La structure itérative «TANT QUE...FAIRE »</b></p> <p>La structure itérative permet de répéter l'exécution d'une opération ou d'un traitement.</p> <p>Dans cette structure, on commence par tester la condition ; si elle est vraie, le traitement est exécuté.</p> <p><b>L'ACTION PEUT NE JAMAIS ETRE EXECUTEE.</b></p>		<p><b>TANT QUE</b> condition <b>FAIRE</b> Opération ; <b>FIN TANT QUE</b></p>	<p><b>while (condition) :</b> Opération1 Opération 2</p>
<p><b>La structure itérative « RÉPÉTER ... JUSQU'À »</b></p> <p>Dans cette structure, le traitement est exécuté une première fois puis sa répétition se poursuit jusqu'à ce que la condition soit vérifiée.</p> <p><b>L'ACTION EST EXECUTEE AU MOINS UNE FOIS</b></p>		<p><b>RÉPÉTER</b> Opération ; <b>JUSQU'À</b> Condition ; <b>FIN RÉPÉTER</b></p>	<p><b>While True :</b> Opération <b>If (Condition==True) :</b> <b>break</b></p>
<p><b>La structure itérative « POUR...DE...À... »</b></p> <p>Dans cette structure, le traitement est exécuté un nombre limité de fois, défini au préalable.</p>		<p><b>POUR i DE i1 À i2</b> <b>FAIRE</b> Opération ; <b>FIN POUR</b></p>	<p><b>for i in range 4 :</b> Opération</p>



## 4. L'organisation d'un algorithme

Un algorithme est organisé de la manière suivante :



On peut distinguer 3 parties bien distinctes :

- **L'entête** : le concepteur donne un nom à l'algorithme, définit le traitement effectué et les données auxquelles il se rapporte.
- **La partie déclarative** : le concepteur décrit les différents « objets » que l'algorithme utilise. On y retrouve les constantes et les variables.
- **La partie exécutive ou le corps de l'algorithme** (Dans cette partie, le concepteur écrit les actions décrivant l'algorithme. Elle est délimitée par les mots « DEBUT » et « FIN ».)

**Ne surtout pas oublier de commenter son programme afin de faciliter la relecture & la maintenance (cf. Lisibilité de l'algorithme).**

