

# 1 Programmer la distance entre deux nombres réels

La distance entre deux nombres réels a été définie dans le cours par :



## Définition :

- La distance entre deux réels  $a$  et  $b$ , notée  $d(a; b)$ , est la différence entre le plus grand de ces deux réels et le plus petit. La valeur de cette distance dépend donc de l'ordre de  $a$  et de  $b$  :
- si  $a > b$  alors  $d(a; b) = a - b$  ;
  - si  $a \leq b$  alors  $d(a; b) = b - a$  ;

À partir de cette définition, on peut définir un algorithme qui demande à l'utilisateur de saisir deux valeurs  $x$  et  $y$  et qui renvoie la distance entre  $x$  et  $y$ . Cet algorithme utilisera une structure conditionnelle : Si *condition* alors : *instruction(s)* . Ci-dessous, on présente l'algorithme écrit en pseudo-langage et sa programmation en Scratch puis en Python.


Algorithme en pseudo-langage	Programme en Scratch	Programme en Python
<hr/> <p align="center">Algorithme du calcul de la distance</p> <hr/> <p><b>Variables</b>  <math>a</math> (réel), <math>b</math> (réel)</p> <p><b>Début</b></p> <pre> 1  a ← Saisir("Valeur de a ? ") 2  b ← Saisir("Valeur de b ? ") 3  Si a &gt; b alors : 4        d ← a - b 5  Sinon : 6        d ← b - a 7  Fin Si 8  Afficher("La distance vaut : ", 9    d) 9  Fin </pre> <hr/>		<p><b>Code Python :</b></p> <pre> 1  a = float(input("Valeur de a ? ")) 2  b = float(input("Valeur de b ? ")) 3  if a &gt; b : 4      d = a - b 5  else : 6      d = b - a 7  print("La distance vaut : ",d) </pre>

Cet algorithme peut être "encapsulé" dans un bloc de fonction :

Algorithme en pseudo-langage	Programme en Scratch	Programme en Python
<hr/> <p align="center">Fonction distance</p> <hr/> <p><b>Variables</b>  <math>a</math> (réel), <math>b</math> (réel)</p> <p><b>Fonction</b> distance(<math>a</math>, <math>b</math>) :</p> <pre> 1  Si a &gt; b alors : 2        d ← a - b 3  Sinon : 4        d ← b - a 5  Fin Si 6  renvoie d 7  Fin Fonction </pre> <hr/>		<p><b>Code Python :</b></p> <pre> 1  def distance(a, b) : 2      """ cette fonction calcule la 3      distance entre deux réels """ 4      if a &gt; b : 5          d = a - b 6      else : 7          d = b - a 8      return d </pre>

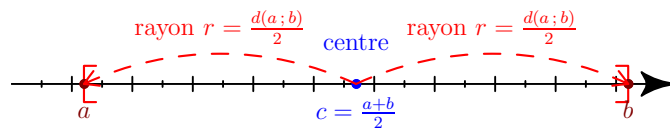
En Python, on déclare une fonction avec le mot-clé **def** et on renvoie un résultat avec le mot-clé **return**

Pour utiliser un tel bloc, il faut **faire un appel de fonction** en attribuant des valeurs aux **paramètres** de la fonction :

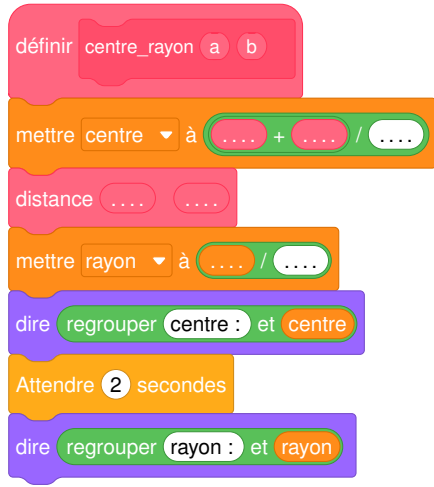
Algorithme en pseudo-langage	Programme en Scratch	Programme en Python
<div>Appel de la fonction</div> <div>distance(-5,13)</div>	 <p>Avec Scratch, on accole le bloc distance avec un bloc chapeau dans la zone de script puis on remplit les champs avec deux valeurs et on clique sur le drapeau. (On peut aussi cliquer sur le bloc seul dans la zone de script).</p>	<div> <b>Console Python :</b>  <pre>&gt;&gt;&gt; distance(-5,13) 18</pre> </div> <p>En Python, on se place dans la console puis on appelle la fonction par son nom en donnant des valeurs aux variables.</p>

## 2 Centre et rayon d'un intervalle

On peut définir le centre et le rayon d'un intervalle borné  $[a; b]$  :



Compléter les algorithmes et programmes suivants pour qu'ils renvoient le centre d'un intervalle  $[a; b]$  dont on aura saisi les extrémités comme paramètres d'une fonction `centre_rayon`, qui réutilisera la fonction `distance` précédente :

Algorithme en pseudo-langage	Programme en Scratch	Programme en Python
<div>Fonction <code>centre_rayon</code></div> <div>Variables</div> <div><math>a</math> (réel), <math>b</math> (réel)</div> <div>Fonction <code>centre_rayon(a, b)</code> :</div> <div>1 centre ← ...</div> <div>2 rayon ← ...</div> <div>3 renvoie centre, rayon</div> <div>4 Fin Fonction</div>		<div>Code Python :</div> <pre>1 def centre_rayon(a, b) : 2     """cette fonction 3     détermine le centre et 4     le rayon d'un intervalle 5     [a ; b]""" 6     centre = ... 7     rayon = ... 8     return centre, rayon</pre>

**Remarque :** en Scratch comme en Python, on peut réutiliser une fonction à l'intérieur d'une autre fonction à condition que ces deux fonctions soient définies dans le même script.

## 3 Résolution d'équations et d'inéquations utilisant la distance

- Définir une fonction `equation(a, r)` qui renvoie les solutions de l'équation  $|x - a| = r$ , avec  $r > 0$ .
  - Traduire cette fonction en une fonction Python.
- Définir une fonction `inequation(a, r)` qui renvoie l'intervalle solution de l'inéquation  $|x - a| \leq r$ , avec  $r > 0$ .
  - Traduire cette fonction en une fonction Python.
- Réfléchir à une fonction `inequation_2(a, r, signe)` qui résoudra les inéquations de type  $|x - a| \dots r$  :

**Console Python :**  

```
>>> inequation_2(4,7,">=")
Solution : ]-∞ ; -3] ∪ [11 ; +∞[
```