

# Mission 3 : Solution d'une équation du second degré

## Python et continuité pédagogique



### Rappel de la mission

- **Objectif :** Écrire un programme qui permet de trouver les solutions d'une équation du second degré.

- **Principe du programme**

Nous allons écrire un programme auquel on va entrer :

- Le coefficient "a"
- Le coefficient "b"
- Le coefficient "c"

Le programme affichera le résultat sous cette forme : ↵

$(r1, r2) \Rightarrow$  si l'équation admet 2 réponses ↵

$(r1,) \Rightarrow$  si l'équation admet une unique réponse ↵

$() \Rightarrow$  si l'équation n'admet pas de solution réelle

- **Fonction à utiliser**

- \* `input()` : fonction qui va demander à l'utilisateur d'entrer un nombre. De base il considère que c'est une chaîne de caractère.
- \* `float()`: fonction qui va dire que ce qui est entré est un nombre réel
- \* `print()`: permet d'afficher (envoyer) le résultat
- \* `if .... elif .... else:` test conditionnel si .... alors , sinon si .... alors, sinon ....  
{(attention à bien mettre ":" après la condition)}

- **Astuce ↵**

pour vous aider, vous pouvez importer la bibliothèque "math" en début de programme

« `import math` » ce qui vous permettra d'utiliser la fonction `sqrt()` qui permet de calculer la racine carré d'un nombre. ↵ exemple :

« `math.sqrt(9)` ↵

`3` »

# Mission 3 : Solution d'une équation du second degré

## Python et continuité pédagogique



### Réponse au programme

Script dans EduPython:

```
•   import math
•
•
•     def rac_eq_2nd_deg(x, y, z):
•         a = x
•         b = y
•         c = z
•         delta = b*b - 4*a*c
•         if delta > 0 :
•             r1 = (-b - math.sqrt(delta))/(2*a)
•             r2 = (-b + math.sqrt(delta))/(2*a)
•             return (min(r1, r2), max(r1,r2))
•         elif delta == 0 :
•             r1 = (-b)/(2*a)
•             return (r1,)
•         else :
•             return ()
•
• 18  print(rac_eq_2nd_deg(float(input()),float(input()),float(input())))
```