Les fonctions avec Fibonacci

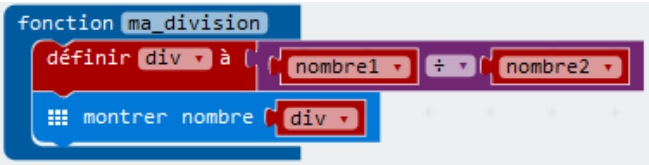
Il arrive souvent qu'un programme doive faire la même chose mais à des moments différents. Plus le programme est grand et compliqué, plus cela arrive. Pour éviter de devoir recopier exactement la même chose plusieurs fois, on va utiliser des **fonctions**. Une fonction est en fait un bout de programme auquel on va donner un nom et dès qu'on aura besoin de ce bout de programme, il suffira d'appeler la fonction grâce à son nom.

Il y a plusieurs utilités aux fonctions :

- **Factoriser** : pour éviter la redondance, c'est-à-dire éviter d'avoir plusieurs fois le même code à des endroits différents.
- **Faire de l'abstraction** : simplifier le code en mettant le nom de la fonction (par exemple « payer ») plutôt que de voir un long code incompréhensible à première vue.
- **Factoriser :** découper le programme en morceaux plus simples pour ne pas tout faire d'un coup.

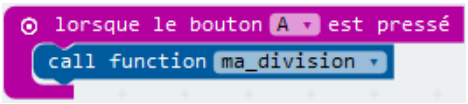
Déclaration, spécification, implémentation et appel de fonction

Déclaration : sert à définir la fonction pour que le programme sache qu'elle existe et ce qu'elle doit faire. La déclaration est composée de l'en-tête et du corps.



En-tête

Corps = implémentation

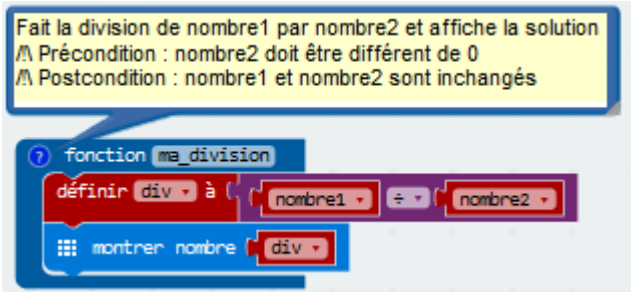


Appel : moment où on utilise la fonction dans le programme.

Implémentation : c'est la suite d'instructions qui sera effectuée par le programme quand on fera appel à la fonction.

Spécification : commentaire[1] dans le code qui explique ce que fait la fonction et, éventuellement, dans quelles conditions elle fonctionne. Elle est composée de :

- Une *description* de ce que fait la fonction
- Les *préconditions* : des choses qui doivent être vraies avant d'appeler la fonctionLes *postconditions* : des choses qui sont vraies après la fonction



[1] Un commentaire est quelque chose qu'on écrit dans le code mais qui n'est pas lu par le programme. C'est pour que les gens qui vont lire le programme plus tard comprennent mieux ce que le programme fait.

Faire un compteur de Fibonacci

La suite de Fibonacci : 0, 1, 1, 2, 3, 5, 8, 13, 21, …
Chaque nombre est la somme des deux précédents (en commençant par 0 et 1)

Fonctionnalité du compteur :

- Lorsqu'on appuie sur B, on augmente de 1 un nombre (nb) (=incrémenter)
- Lorsqu'on appuie sur A, on diminue de 1 ce nombre (nb) (=décrémenter)
- Lorsque le micro:bit est secoué, il affiche le nb^{ème} élément de Fibonacci
- Lorsqu'on appuie sur A+B, on remet tout à 0 (=réinitialiser)

Marche à suivre :

1. Quand on démarre, on initialise toutes les variables
    a. Quelles sont les variables utiles ?
    b. À combien doit-on les initialiser ?
2. Incrémenter et décrémenter le nombre nb avec B et A
    a. Incrémenter le nombre nb avec B
    b. Décrémenter le nombre nb avec A
3. Créer la fonction « suivant »
    a. Créer la fonction
    b. L'utiliser correctement pour calculer le nb^{ème} nombre de Fibonacci
4. Réinitialiser quand on appuie sur A+B
    a. Implémenter la réinitialisation
    b. Tenter de factoriser
5. Tester