

Les fonctions

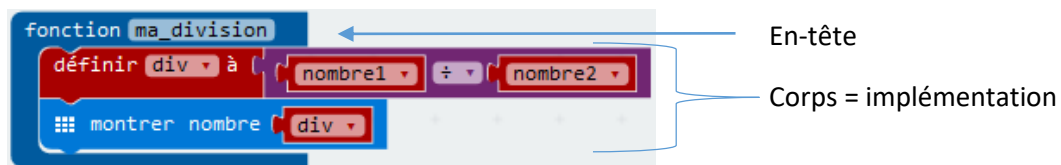
Il arrive souvent qu'un programme doive faire la même chose mais à des moments différents. Plus le programme est grand et compliqué, plus cela arrive. Pour éviter de devoir recopier exactement la même chose plusieurs fois, on va utiliser des **fonctions**. Une fonction est en fait un bout de programme auquel on va donner un nom. Dès qu'on aura besoin de ce bout de programme, il suffira d'appeler cette 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.
- **Pas tout faire d'un coup** : découper le programme en morceaux plus simples (diviser pour régner) 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 que la fonction existe et ce qu'elle doit faire. La déclaration est composée de l'en-tête et du corps.



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

```
call function ma_division
```

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

Spécification : c'est un commentaire¹ dans le code qui explique ce que fait la fonction et, éventuellement, dans quelles conditions elle fonctionne. Elle est en général composée de :

- Une *description* de ce que fait la fonction
- Les *préconditions* : des choses qui doivent être vraies avant de lancer la fonction
- Les *effets de bord* : des choses qui sont impactées par la fonction (typiquement, des variables qui sont modifiées)
- Les *postconditions* : des choses qui sont vraies après la fonction

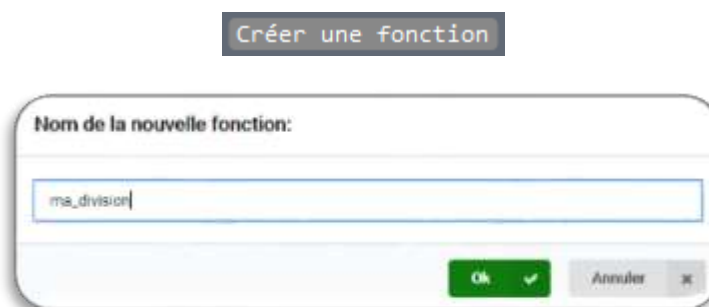
¹ Un commentaire est quelque chose qu'on écrit dans le code mais qui n'est pas lu par le programme

En bloc

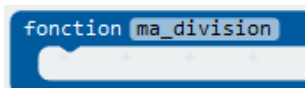


Lorsqu'on code en bloc avec micro:bit, les fonctions se trouvent dans le menu avancé. Au début, le menu des fonctions ne contient aucun bloc d'appel de fonction car aucune fonction n'a été déclarée, il n'en existe donc tout simplement pas.

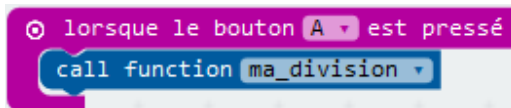
Pour **déclarer** une fonction, il suffit de cliquer sur le bouton « créer une fonction » en haut du menu fonction et ensuite nommer celle-ci :



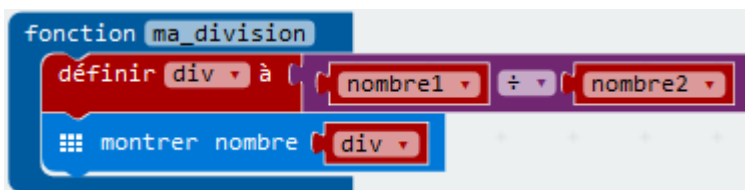
Le bloc de déclaration de la fonction apparaîtra alors :



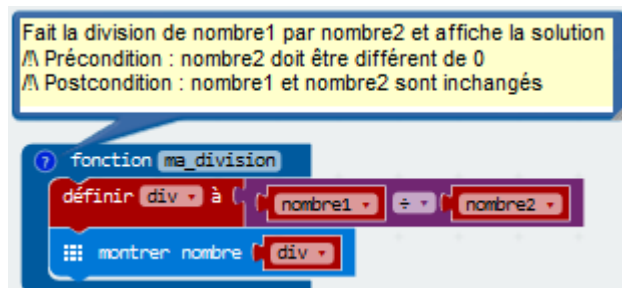
En retournant dans le menu fonction, un nouveau bloc avec le nom de la fonction s'y trouvera, il s'agit du bloc d'**appel** de la fonction. Il pourra être utilisé comme n'importe quel bloc d'instruction et lorsque le programme rencontrera ce bloc, il fera ce qui se trouve dans le bloc de déclaration (pour le moment rien). Par exemple :



Pour **implémenter** une fonction, il suffit de remplir le bloc de déclaration. Dans le langage de programmation par bloc, toutes les variables sont considérées comme « globales », elles peuvent donc être librement utilisées et modifiées² par la fonction.



Pour faire la **spécification**, il faut faire un clic droit sur le bloc de déclaration et sélectionner « ajouter un commentaire ». Un petit point d'interrogation apparaîtra alors en haut à gauche du bloc. Lorsque vous cliquez dessus, un cadre vous permettra d'y inscrire vos commentaires.



² Les variables modifiées dans la fonction le seront définitivement c'est-à-dire qu'elles garderont leur nouvelle valeur quand la fonction sera terminée.