

Variables et conditionnelles

Qu'est-ce qu'une variable ?

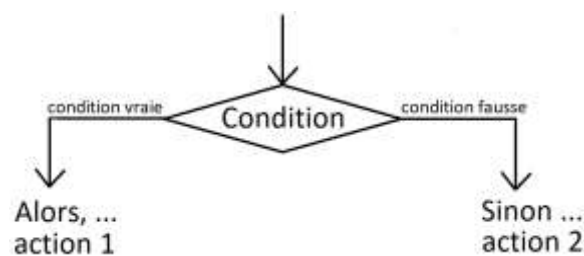
Un programme traite en général énormément d'informations. De ce fait, il est nécessaire de stocker ces informations quelque part et plus précisément, dans la mémoire. Les informations à stocker peuvent être de plusieurs types : des nombres (entiers ou réels), du texte, des valeurs booléennes (vrai ou faux) ou éventuellement des structures de données plus complexes.

Ces valeurs vont être stockées dans des variables. Une variable associe un identifiant (= nom unique) à une valeur en mémoire. On peut mettre une valeur dans une variable, changer cette valeur ou la récupérer pour l'utiliser ailleurs.

Étape	Instruction	Mémoire
1	a = 5;	a: 5
2	b = 2;	a: 5 b: 2
3	a = a + 3;	a: 8 b: 2
4	c = a * b;	a: 8 b: 2 c: 16
5	a = c;	a: 16 b: 2 c: 16
6	d = "Val";	a: 16 b: 2 c: 16 d: "Val"

Qu'est-ce qu'une conditionnelle ?

Une importante partie de la programmation est de pouvoir dire *quand* un programme doit effectuer une certaine action. Les instructions conditionnelles permettent de le faire en indiquant une condition dans laquelle une action sera faite et éventuellement ce que le programme doit faire si la condition est fautive. La condition aura pour valeur **vrai** ou **faux**, il peut s'agir par exemple d'une comparaison de variables ou d'une variable booléenne.



Si la condition est **vraie**, alors l'instruction va exécuter **l'action 1** mais si la condition est **fautive**, alors l'instruction va exécuter **l'action 2**. On exécutera soit l'action 1, soit l'action 2 mais jamais les deux.

Un exemple de conditionnelle est « SI il pleut ALORS reste à la maison SINON va à la plage » où on a deux actions différentes et on va effectuer l'une des deux actions en fonction de la condition.

Jeu du plus ou moins

Le micro:bit choisi un nombre entre 0 et 10 (inclus) et le joueur doit essayer de trouver ce nombre grâce aux indications du micro:bit. Le joueur indiquera un nombre grâce aux boutons A et B et lorsqu'il secouera le micro:bit, celui-ci indiquera si le nombre est plus grand, plus petit ou si c'est gagné.

Marche à suivre :

1^{ère} partie : Les variables

Nous allons avoir besoin de 2 variables : le nombre choisi par le micro:bit et le nombre que l'on va proposer. Commençons par créer ces 2 variables : je vais les appeler « choix_microbit » et « mon_nombre ». *(Vous pouvez leur donner le nom que vous voulez mais l'idéal est un nom significatif, c'est-à-dire qui montre bien à quoi elle sert)*

1. Au démarrage, il faut :
 - a. Définir *mon_nombre* à 0
 - b. Afficher *mon_nombre* *(pour que le joueur sache combien elle vaut)*
 - c. Définir *choix_microbit* au hasard de 0 à 10 *(c'est le nombre qu'il faudra deviner)*
2. Lorsque on appuie sur A, il faut :
 - a. Diminuer *mon_nombre* de 1
 - b. Afficher *mon_nombre* *(pour que le joueur sache combien elle vaut)*
3. Lorsque on appuie sur B, il faut :
 - a. Augmenter *mon_nombre* de 1
 - b. Afficher *mon_nombre* *(pour que le joueur sache combien elle vaut)*

2^{ème} partie : Les conditions

Il faut modifier les blocs « lorsqu'on appuie sur A » et « lorsqu'on appuie sur B » car le nombre à deviner étant entre 0 et 10, ça ne sert à rien de proposer un nombre plus petit que 0 ou plus grand que 10. Cela deviendra :

Lorsque on appuie sur A :

- a. **Si** *mon_nombre* est > 0, **alors** diminuer sa valeur de 1
- b. Afficher *mon_nombre* *(pour que le joueur sache combien elle vaut)*

Lorsque on appuie sur B, il faut :

- c. **Si** *mon_nombre* est < 10, **alors** augmenter sa valeur de 1
- d. Afficher *mon_nombre* *(pour que le joueur sache combien elle vaut)*

Dernière étape :

Lorsqu'on secoue :

- a. **Si** *mon_nombre* est < que *choix_microbit*
 - **alors** afficher « + »
 - **sinon Si** *mon_nombre* est > que *choix_microbit*
 - **alors** afficher « - »
 - **sinon** afficher « bravo »
- (si le nombre n'est ni < ni > c'est que c'est le bon)*

