

Fiche d'activité Botly : Cycle Initiation

Séance 4 : L'instruction conditionnelle 1

Durée : environ 45 minutes

Résumé de l'activité : Intégrer la notion d'instruction conditionnelle et ses contextes d'usages

Objectifs Pédagogiques :

- Appréhender les notions de base de la programmation (instruction, paramètre, succession d'instruction)
- Expérimenter la programmation par bloc
- Aborder la notion de "condition" et savoir l'appréhender en programmation
- Maîtriser la notion d'instruction conditionnelle

Participants : Entre 8 et 16 participants (2 élèves par ordinateur)

Matériel : 4 à 8 ordinateurs avec Botly-Studio, ou disposant d'une connexion internet, si possible un robot pour la présentation

Préparation : Disposer 8 ordinateurs disposant d'une connexion internet et ouvrir botly-studio.fr dans un navigateur (**MODE : DEBUTANT**)

Notions liées : Instruction, Bloc, Programme, Simulation, Botly, Boucle, instruction conditionnelle, condition

Introduction et rappel : 10'

Résumer la séance précédente par une série de question :

1. En programmation, comment appelle-t-on ce que l'on a vu et qui est comme une boîte dans laquelle on peut stocker une information ? **Une variable**
2. Comment s'appelle l'opération par laquelle on crée une variable et on lui attribue une valeur ? **C'est ce que l'on appelle déclarer une variable**
3. *Pouvez-vous me donner des exemples de à quoi sert une variable ?* **Une variable peut être utile pour contrôler la taille de plusieurs objets en une opération.**
4. Comment s'appelle la technique par laquelle on crée des spirales ? **La technique de l'incrémentation.**

Jusqu'à présent nous avons appris à programmer le déplacement du robot pour qu'il réalise les actions que nous lui indiquons dans un ordre bien défini et prévu à l'avance. Or parfois, on ne pas toujours tout prévoir. Un obstacle sur le chemin du robot ou une bordure de cadre, et le voilà bloqué, incapable de finir son programme.

Dans la vie, on est souvent confronté à des situations dans lesquelles il y a plusieurs possibilités. Et parfois, on doit faire face à des situations non prévues... bref, on doit pouvoir **CHOISIR entre plusieurs options.**

C'est à cela que sert l'instruction conditionnelle.

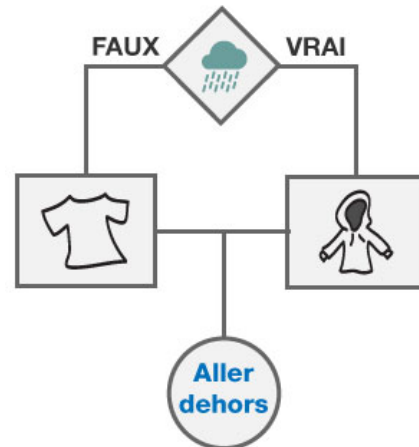
La notion d'instruction conditionnelle - 20'

Le terme "*instruction conditionnelle*" peut sembler un peu complexe, mais cela correspond à quelque chose que nous connaissons tous dans la vie quotidienne.

Par exemple lorsque l'on se demande comment s'habiller avant de sortir. Pour savoir s'il faut ou non prendre un manteau, on se demande s'il pleut. Si oui, on prend un manteau, si non on peut sortir en pull.

On peut le résumer avec le schéma ci-contre.

En langage de programmation, on dira que **SI** la condition "il pleut" est vraie **ALORS** on prendra un manteau, **SINON** on sortira en pull. C'est ce schéma que l'on va retrouver dans les blocs.



Source : Educocode : cours sur l'algorithmique

L'instruction SI ... ALORS

Pour accéder à ce type de bloc, on va ouvrir la bibliothèque "Logique", et choisir le premier bloc "Si... Faire...".

Avec ce bloc, on va pouvoir créer notre première instruction conditionnelle. Dans le crochet suivant le "SI", on va écrire la CONDITION à vérifier (dans l'exemple de tout à l'heure : il pleut). Dans le crochet suivant "FAIRE" on va écrire les instructions à réaliser.

Exemple :

Pour le premier exemple, on va choisir une condition simple : $1 = 1$. Si l'opération est vraie, le robot va avancer. Sinon il restera immobile. Voici les blocs à utiliser : voir ci-contre.



La solution :

Dans ce cas, on demande au robot ou au programme de vérifier que l'équation $1 = 1$ est exacte. Si oui, le robot avance, sinon, il reste immobile.

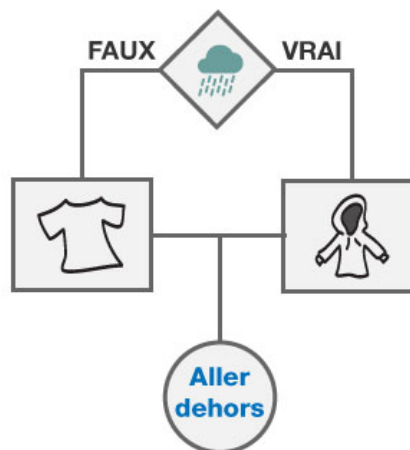


Mais pourquoi reste-t-il immobile ? Tout simplement parce qu'il passe à la suite des instructions. Il sort de l'instruction conditionnelle. On ne peut donc pas lui demander de réagir différemment si la condition n'est pas exacte. Pour cela, on va avoir besoin d'une instruction supplémentaire.

L'instruction SI ... ALORS ... SINON

Dans le cas précédent, on a écrit une instruction qui vérifie si une condition est VRAIE ou FAUSSE et qui exécute une instruction si elle est vraie. Mais lorsqu'elle est fausse, rien ne se passe, le programme passe à la suite des instructions. Comment faire alors lorsque l'on veut que le robot exécute une autre action si la condition n'est pas remplie ?

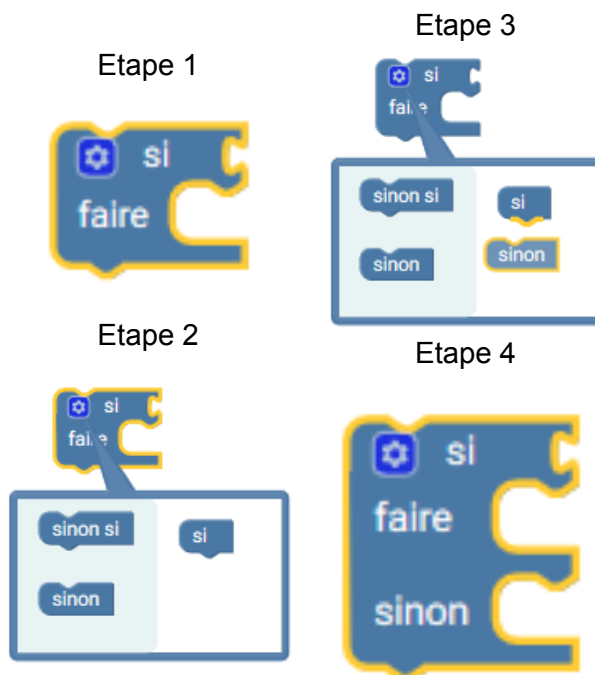
Pour reprendre l'exemple de départ, on a la condition "IL PLEUT". SI elle est vraie, ALORS je prends mon manteau. Mais SI elle est fausse ALORS, il ne se passe rien... Comment faire pour que je prenne mon pull lorsqu'il ne pleut pas ?



Pour pouvoir indiquer une autre instruction si la condition n'est pas remplie, il faut utiliser l'instruction SI... ALORS... SINON !

Pour cela, on va simplement modifier un peu l'instruction conditionnelle de base, en cliquant sur le petit engrenage.

- Etape 1 : on clique sur le petit engrenage bleu marine.
- Etape 2 : une fenêtre contenant plusieurs blocs s'ouvre.
- Etape 3 : on vient positionner le bloc « Sinon » sous le bloc « SI » figurant dans le cadre.
- Etape 4 : A la fin de la manipulation, le bloc de l'instruction conditionnel est modifié et contient un étage supplémentaire.



Une fois cette instruction modifiée, on va pouvoir écrire 2 instructions différentes et différencier le cas où l'instruction est VRAIE de celui où l'instruction est FAUSSE.

Voir l'exemple ci-contre.



Exercice d'application :

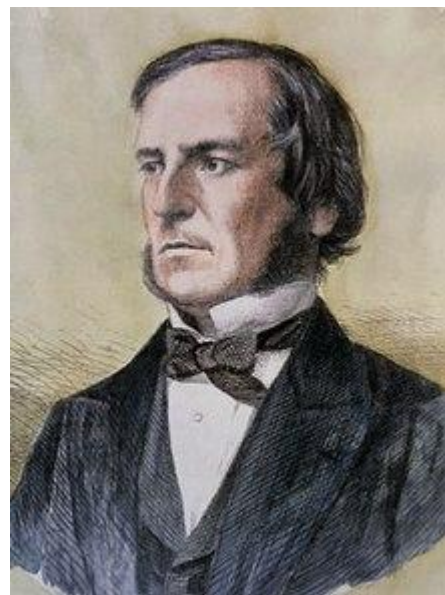
Écrivez une instruction conditionnelle qui indique :

“Si un chiffre (par exemple : 4) est pair, ALORS avancer de 100 mm, SINON, reculer de 100 mm”.



Les Boléens

Une dernière chose avant de passer aux exercices. Dans les exemples que nous avons évoqués, nous avons utilisé des conditions qui étaient soit VRAIES, soit FAUSSES. Dans la vie de tous les jours, les choses sont souvent plus compliquées, mais en programmation, on a besoin de simplifier. On a donc souvent recours à des outils mathématiques et logique. Dans ce cas, on utilise la logique développée par le mathématicien Georges Boule au XIXème siècle (voir illustration ci-contre).



George Boole vers 1860 ([Wikipédia](#))

Il crée une algèbre binaire, dite [booléenne](#), n'acceptant que deux valeurs numériques : 0 et 1. Cette algèbre aura de nombreuses applications en téléphonie et en informatique.

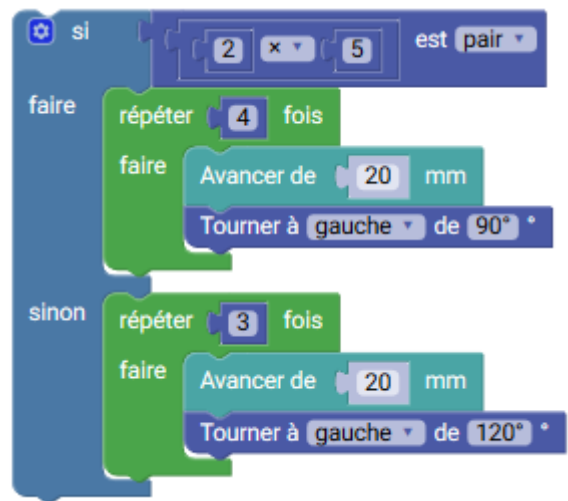
Pour les variables qui ne peuvent être que VRAIE ou FAUSSE, on parle de variable de type booléenne ou de booléen.

a	b	c	b . c	a + b . c	a + b	a + c	(a + b) . (a + c)
0	0	0	0	0	0	0	0
0	0	1	0	0	0	1	0
0	1	0	0	0	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1
1	1	0	0	1	1	1	1
1	1	1	1	1	1	1	1

Défi 1 : Un programme qui trace des figures différentes en fonction d'un calcul

Le premier défi est de construire un programme qui fasse dessiner une figure différente en fonction du résultat d'un calcul. Le programme devra respecter les conditions suivantes :

- Si le résultat est pair, il faudra tracer un CARRE
- Si le résultat est impair, il faudra tracer un TRIANGLE
- En ce qui concerne la condition : l'opération devra être la MULTIPLICATION DE 2 CHIFFRES et devra utiliser le bloc "... est pair"
- Il faudra utiliser une instruction SI... ALORS...SINON



Défi 2 : Un programme qui trace des figures différentes si un calcul est exact

Le deuxième défi est de construire un programme qui fasse dessiner une figure différente en fonction du résultat d'un calcul. Le programme devra respecter les conditions suivantes :

- Si le résultat est vrai, il faudra tracer un CARRÉ
- Si le résultat est faux, il faudra tracer un TRIANGLE
- En ce qui concerne la condition : l'opération devra être la MULTIPLICATION DE 2 CHIFFRES ET SON RÉSULTAT et devra utiliser le bloc "... est vrai"
- Il faudra utiliser une instruction SI... ALORS...SINON



$$6 \times 9 = 53$$

FAUX

$$400 \times 45 = 7500$$

FAUX

Défi 3 : Vérifier facilement si les opérations suivantes sont correctes ou non

Grâce à votre programme, vous pouvez rapidement savoir si une opération est correcte ou non. Utilisez le pour vérifier le résultat des opérations suivantes :

$$62 \times 65 = 4030$$

VRAI

$$666 \div 9 = 74$$

VRAI

$$14 \times 99 = 1368$$

FAUX

$$210 \div 140 = 30 \div 20$$

VRAI

Ainsi, on peut voir que l'on peut programmer des actions même dans le cas où l'on ne connaît pas le résultat de l'opération. Il suffit de demander au programme d'effectuer le calcul.

Conclusion

Bilan

Qu'est-ce que j'ai appris ? Qu'est-ce qui était difficile ?

A retenir

En programmation, il est parfois nécessaire de **choisir entre plusieurs options**, comme dans la vie où l'on est confronté à des choix (mettre un manteau ou un pull).

Pour cela, on utilise une **instruction conditionnelle** qui vérifie si une **condition** est remplie ou non.

La structure d'une instruction conditionnelle peut être SI...ALORS, mais aussi SI...ALORS....SINON.

On peut également faire appel à la logique booléenne qui repose sur une algèbre binaire reposant sur des 0 et des 1 (ou VRAI et FAUX) pour construire ses programmes.

Trace écrite