

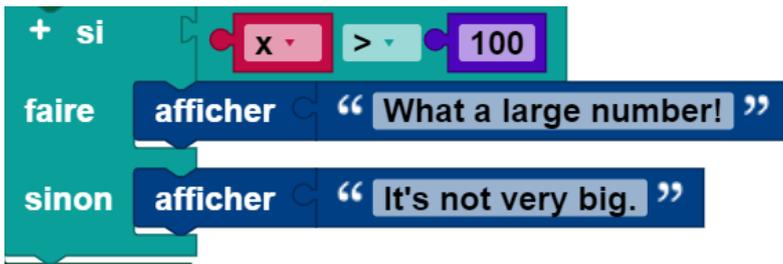
Logique

La logique booléenne est un système mathématique simple qui a deux valeurs :

- vrai
- faux

Les blocs logiques dans ROBO Pro Coding sont généralement destinés à contrôler les conditions et les boucles.

Voici un exemple :



Si la valeur de la variable x est supérieure à 100, la condition est vraie et le texte « Quel grand nombre ! » est édité. Si la valeur de x n'est pas supérieure à 100, la condition est fausse et « Ce n'est pas très grand » est édité. Les valeurs booléennes peuvent également être stockées dans des variables et transmises à des fonctions, de même que les nombres, le texte et les valeurs de liste.

Blocs

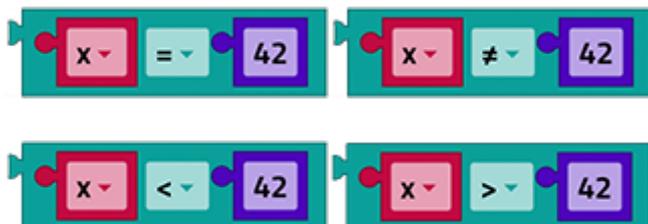
Si un bloc attend une valeur booléenne comme entrée, une entrée manquante est interprétée comme **incorrecte**. Les valeurs non booléennes ne peuvent pas être insérées directement là où des valeurs booléennes sont attendues, bien qu'il soit possible (mais non conseillé) d'enregistrer une valeur non booléenne dans une variable et de l'insérer ensuite dans l'entrée de condition. Cette méthode n'est pas recommandée et son comportement peut changer dans les versions futures de ROBO Pro Coding.

Valeurs

Un seul bloc avec une liste déroulante indiquant soit **vrai**, soit **faux** peut être utilisé pour extraire une valeur booléenne :

Opérateurs de comparaison

Il y a six opérateurs de comparaison. Deux entrées (normalement deux nombres) sont transmises à chacune et l'opérateur de comparaison renvoie **vrai** ou **faux**, selon la manière dont les entrées sont comparées.





Les six opérateurs sont : égal, non égal, inférieur, supérieur, inférieur ou égal, supérieur ou égal.

Opérateurs logiques

Le bloc **et** transmet alors et seulement alors le signal **vrai** si ses deux valeurs d'entrée sont vraies.



Le bloc **ou** transmet le signal **vrai** si au moins une de ses deux valeurs d'entrée est vraie.



pas

Le bloc **pas** transforme une saisie booléenne en sa contrepartie. Par exemple, le résultat de :



faux.

En l'absence de saisie, la valeur **vraie** est enregistrée, de sorte que le bloc suivant génère la valeur **faux** :



Toutefois, il n'est pas recommandé de laisser une entrée vide.

Triple opérateur

Le triple opérateur se comporte comme un bloc **si-sinon** miniature. Il prend trois valeurs d'entrée, la première valeur d'entrée étant la condition booléenne à tester, la deuxième valeur d'entrée étant la valeur à restituer si le test est **vrai**, la troisième valeur d'entrée étant la valeur à restituer, si le test est faux. Dans l'exemple suivant, la variable **Couleur** est définie sur le rouge si la variable **x** est inférieure à 10, sinon la variable **Couleur** est définie sur le vert.



Un bloc triple peut toujours être remplacé par un bloc **si-sinon**. Les deux exemples suivants sont exactement identiques.

A Scratch 'affichez' block with a conditional test. The test is 'x = 0'. The 'si vrai' branch contains the text 'Game over!'. The 'si faux' branch contains the text 'It's going on!'.

A Scratch 'si' block with a conditional test. The test is 'x = 0'. The 'faire' branch contains an 'affichez' block with the text 'Game over!'. The 'sinon' branch contains an 'affichez' block with the text 'It's going on!'.

Revision #6
Created 17 November 2021 20:44:07 by Admin
Updated 18 February 2022 13:51:58 by Admin