

# Atelier kids-lab.io

Créé par Yann Caron



**kids-lab.io**

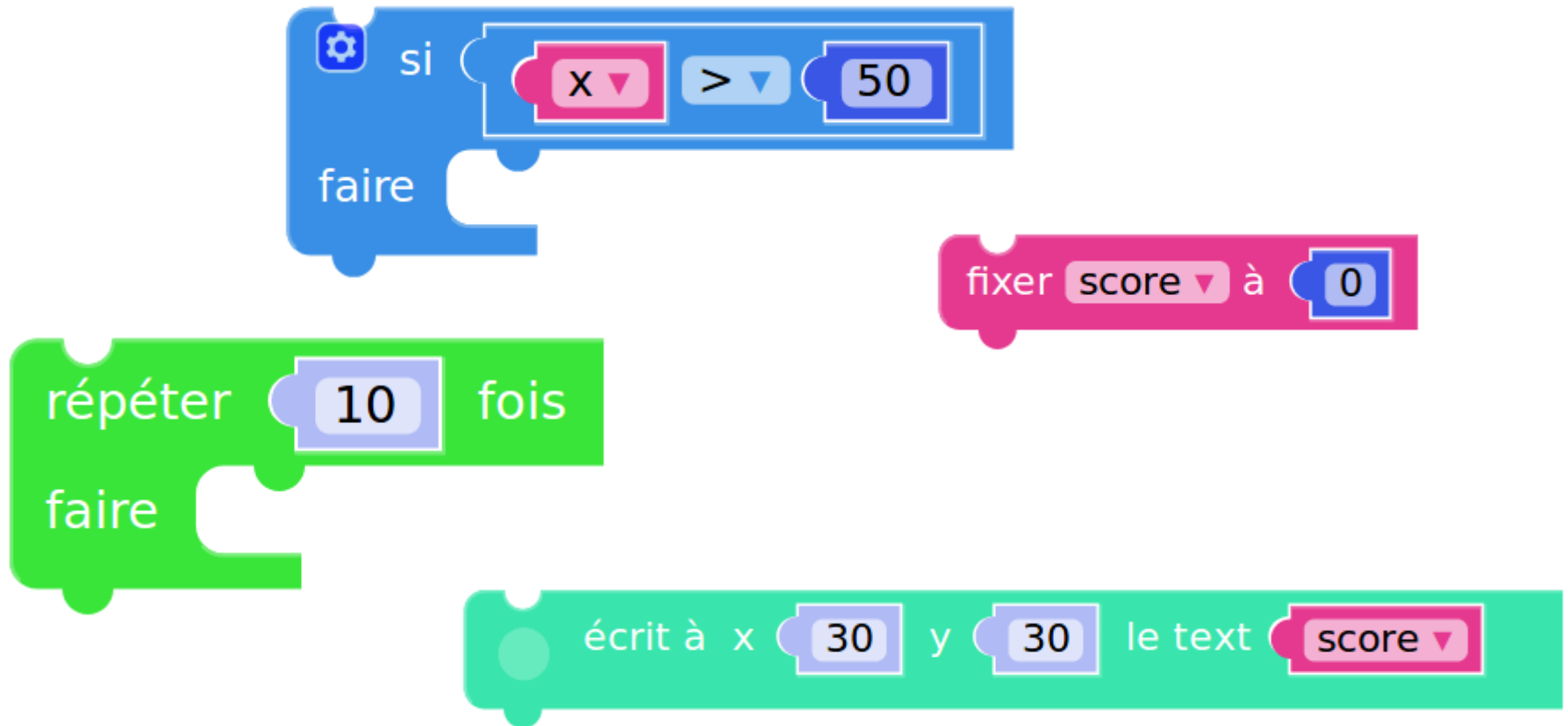
# Définition :: Un programme

- Suite d'instruction :: ce que le programme doit faire
- Structure :: comment le programme est organisé

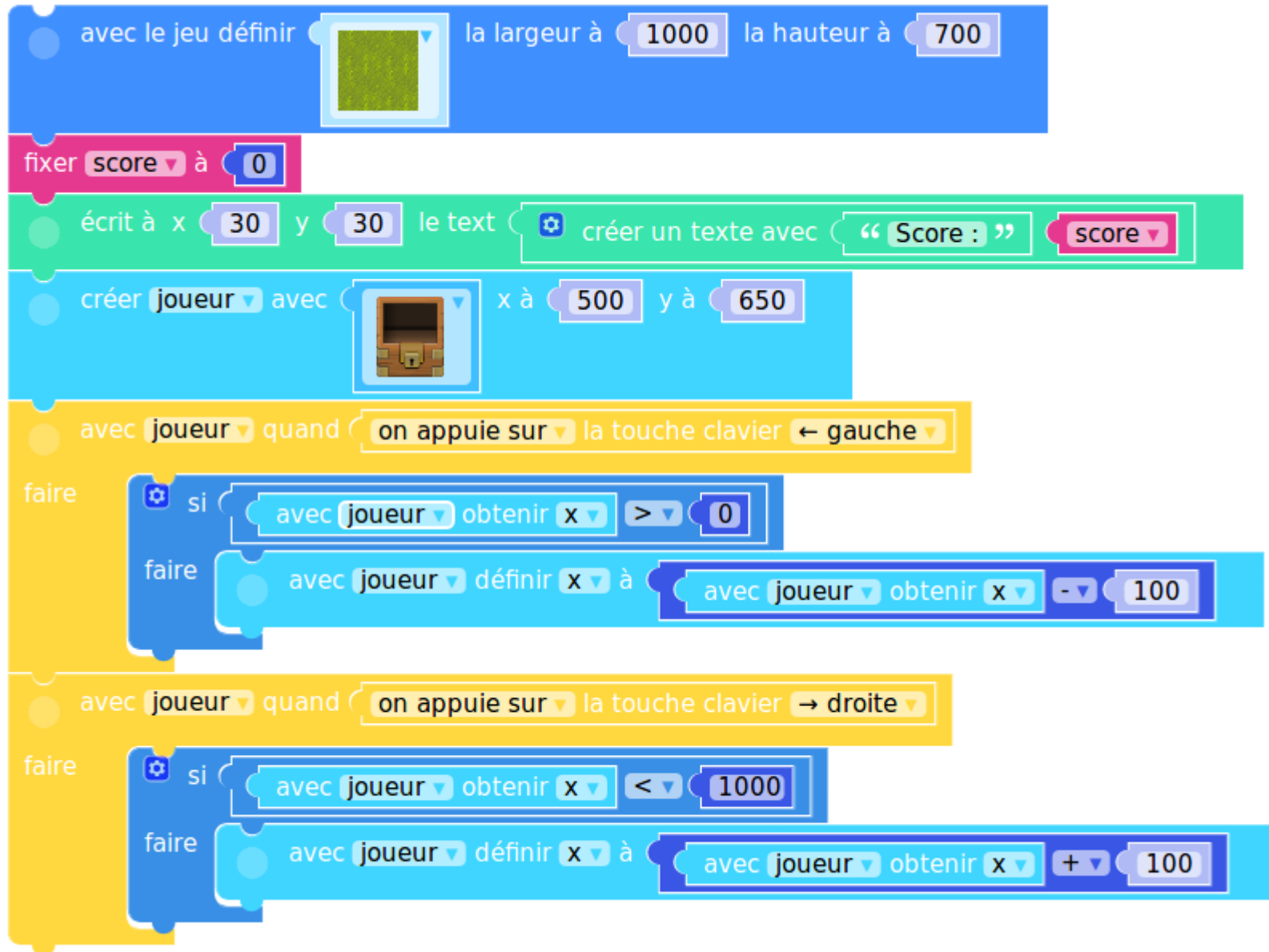


# Les blocs

- Un bloque :: une instruction :: un ordre

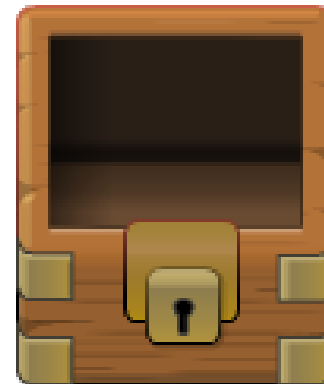


# Un programme avec des blocs



# Un moteur de jeu ?

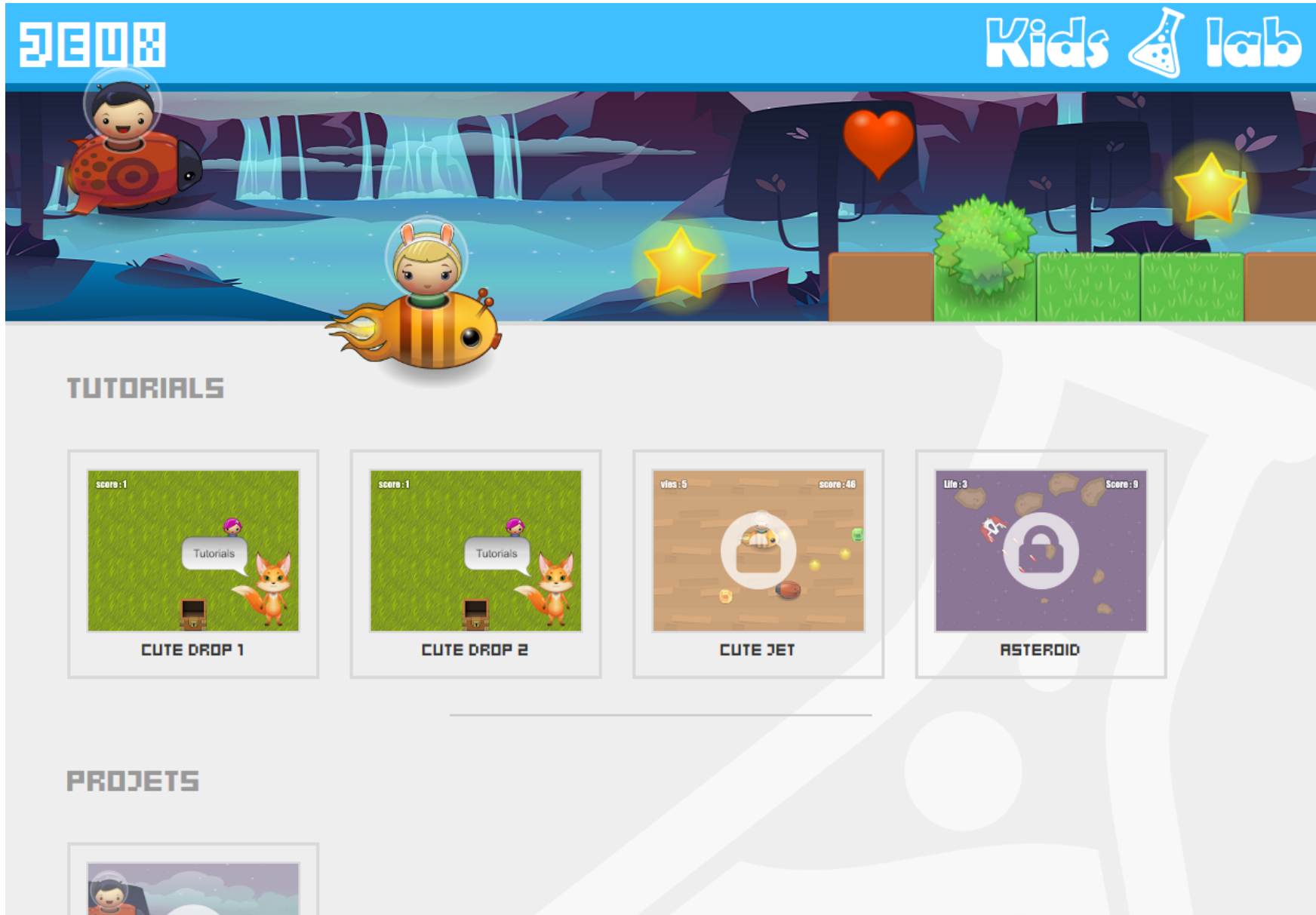
- Gère des acteurs ::
  - Propriétés (ex position)
  - Méthodes (ex déplacer)
  - Événements (ex quand touche)
- Gère la physique
- Gère les collisions



# Notre jeu :: cute-drop






# Notre logiciel :: kids-lab.io





# Notre logiciel :: tutoriels

**GAME**    


Kids  lab


Jeu

Acteur

Évènement

Math

avec le jeu définir  taille plein écran (largeur : 8, hauteur : 6) physique ☒


crée hero avec  x à 4 y à 6





quand on appuie sur la touche flèche droite → du clavier


faire avec hero définir x à avec hero obtenir x + 1

quand on appuie sur la touche flèche gauche ← du clavier

faire avec hero définir x à avec hero obtenir x - 1

crée drop avec  x à 1 y à 1

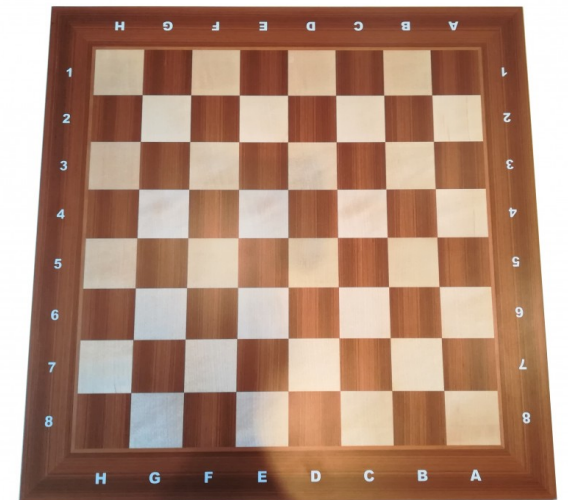






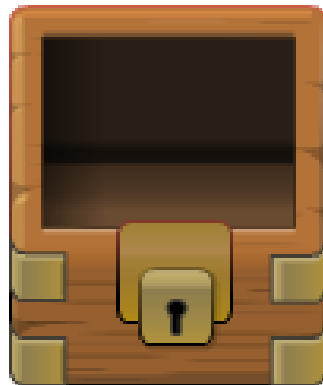
**A nous de jouer !  
Lançons le tutoriel**

# Coordonnées



# Question

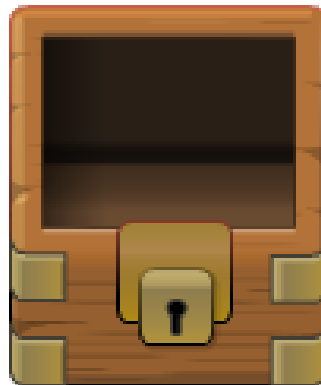
- Comment placer notre coffre en bas ?
- Comment placer notre coffre au centre (horizontal) ?



**A nous de jouer !**  
**Continuons le tutoriel**

# Question

- Comment déplacer notre coffre ?
- On veut le déplacer de 1 vers la droite
- A chaque fois qu'on appuie sur la touche droite

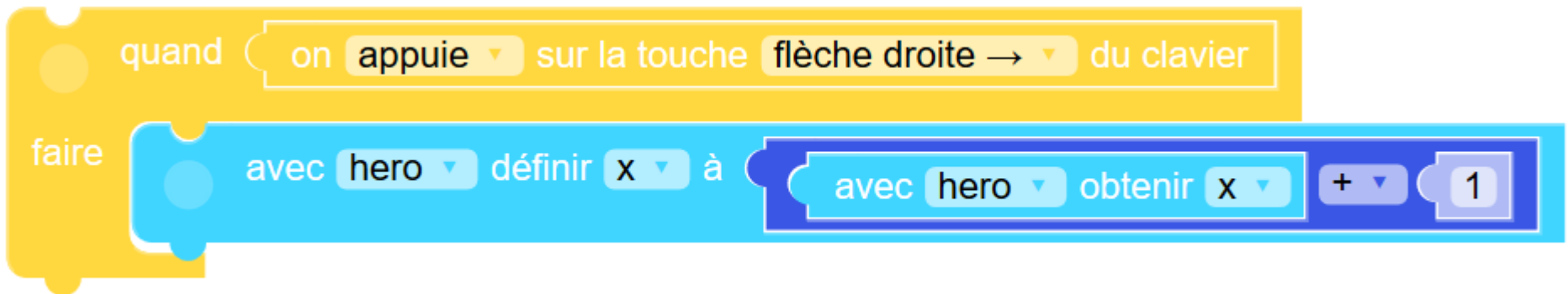


# Indices



# Solution

- La position du joueur  $\Leftarrow$  la position du joueur + 1
- $x \Leftarrow x + 1$
- $x = x + 1$





**A nous de jouer !**  
**Poursuivons le tutoriel**

# Du nouveau :: Le hasard

- Demander à l'ordinateur de trouver un nombre au hasard pour nous
- On utilise la fonction

entier aléatoire entre

1

et

100



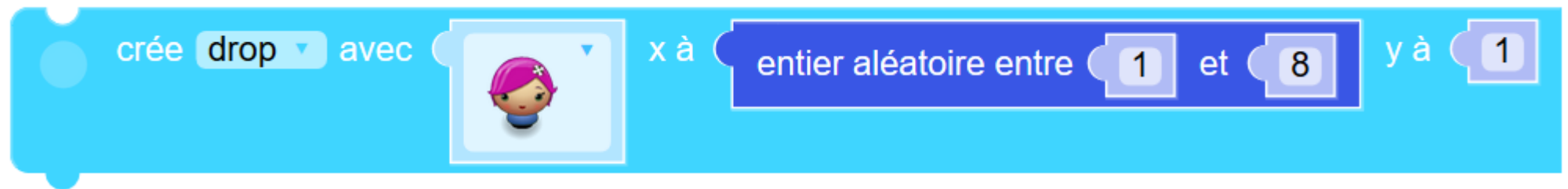
# Question

- On veut que « **drop** » ne soit pas toujours à la même position.
- Autrement dit, on veut qu'il ait une position différente à chaque fois qu'on lance le jeu



# Solution

- On va utiliser la fonction de « hasard » ou « random » en anglais
- Et on va dire que drop est créé avec pour coordonnée X un nombre choisi au hasard entre 1 et 8

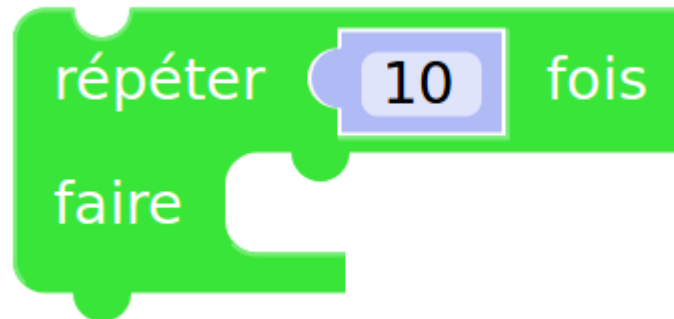


**A nous de jouer !**  
**On retourne sur le tutoriel**

# Du nouveau :: Les boucles

- Répéter un morceau de programme plusieurs fois

- On utilise



# Question

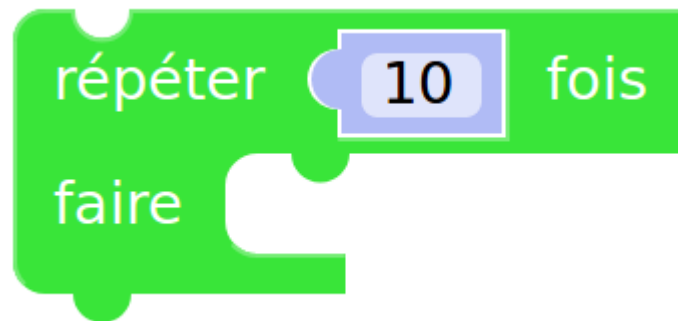
- On veut créer plusieurs drop



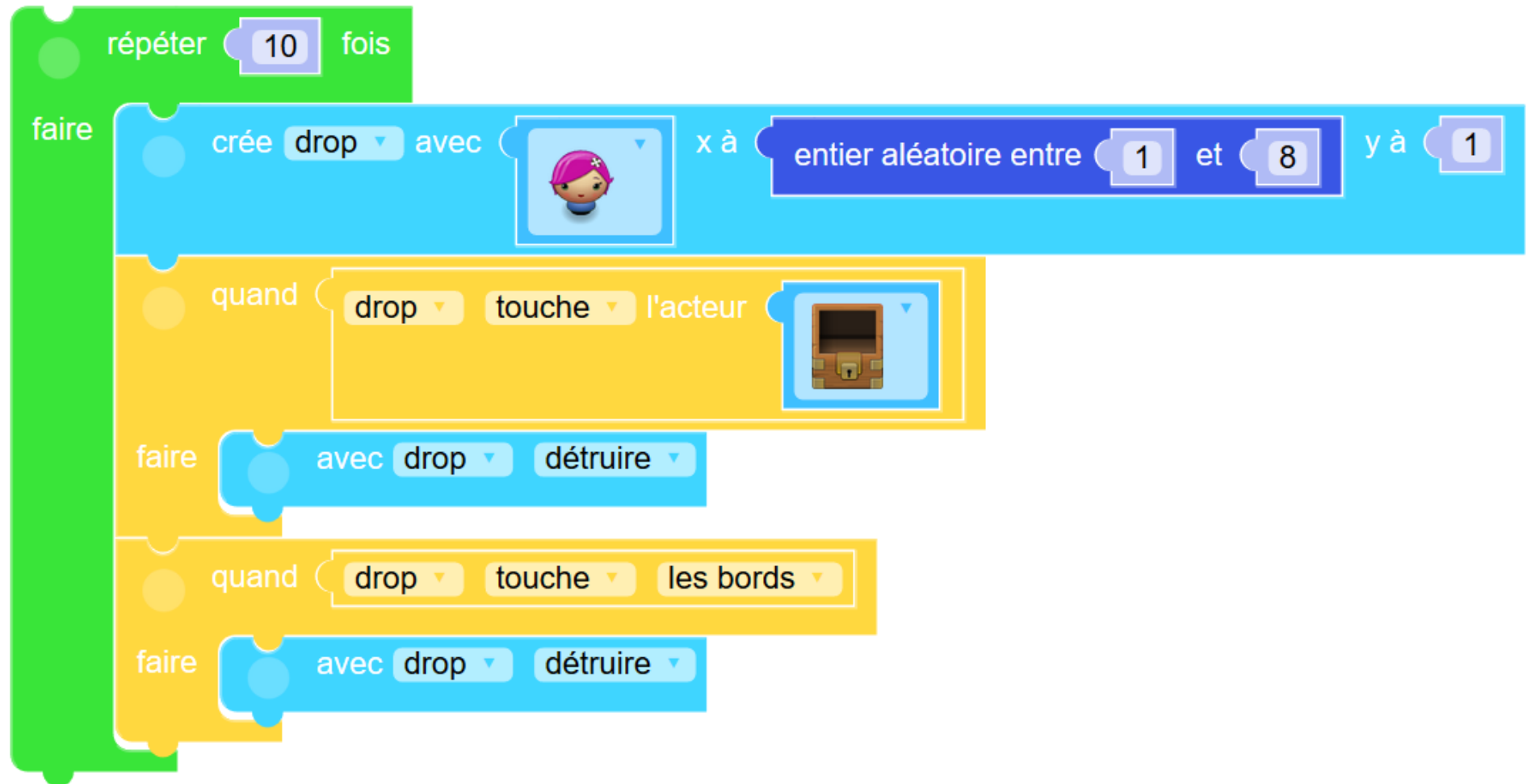


# Indice

- On va utiliser une boucle pour répéter la partie du programme qui crée les drops



# Solution

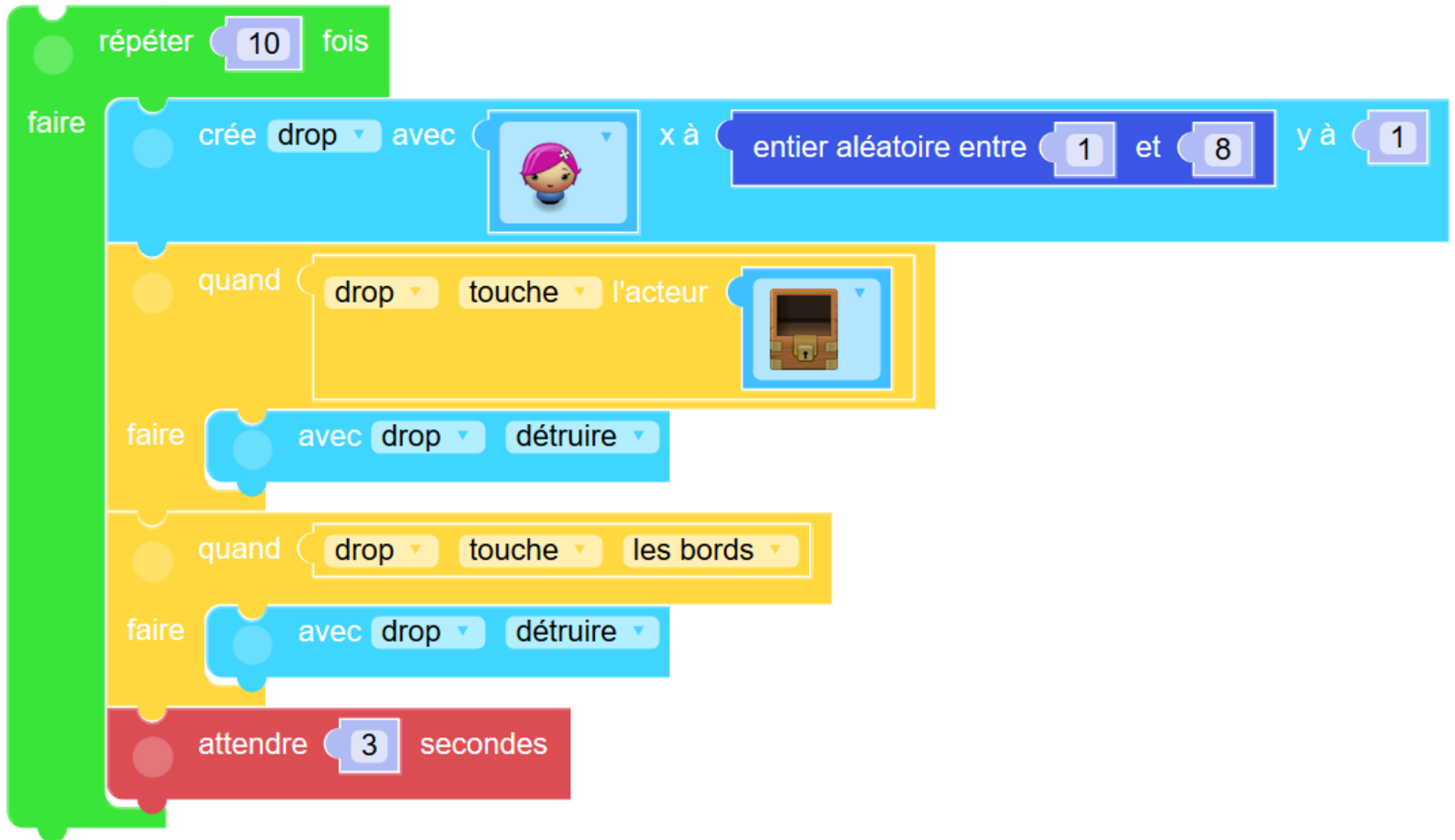


# Pause !

- Et pour qu'ils ne tombent pas tous en même temps, on va mettre le programme en pause



# Tout ensemble



**A nous de jouer !**  
**On continue le tutoriel**

# Du nouveau :: Les variables

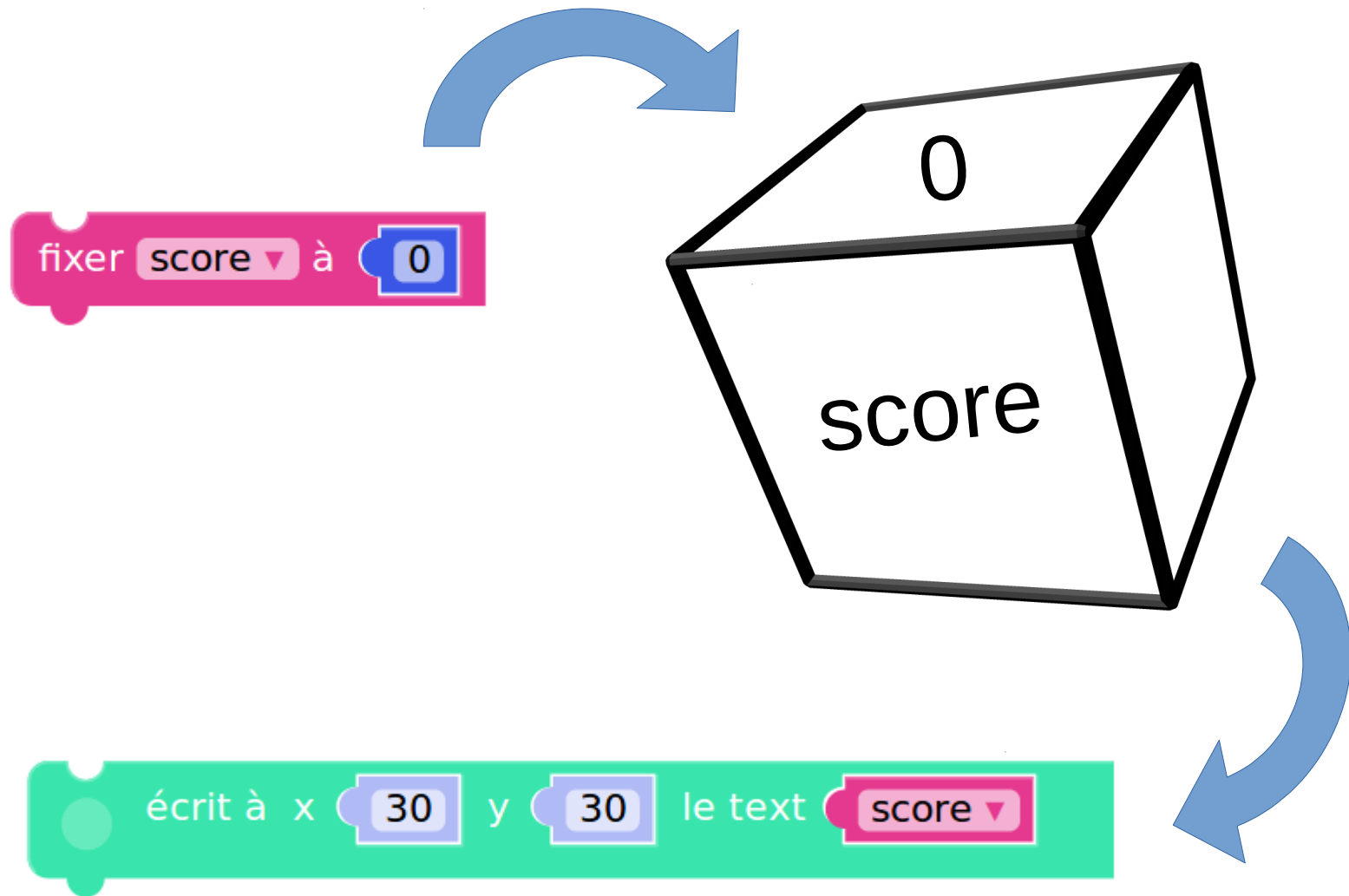
- Une boîte dans la mémoire de l'ordinateur qui a un nom
- On peut mettre un nombre dedans



On peut utiliser le nombre



# Du nouveau :: Les variables





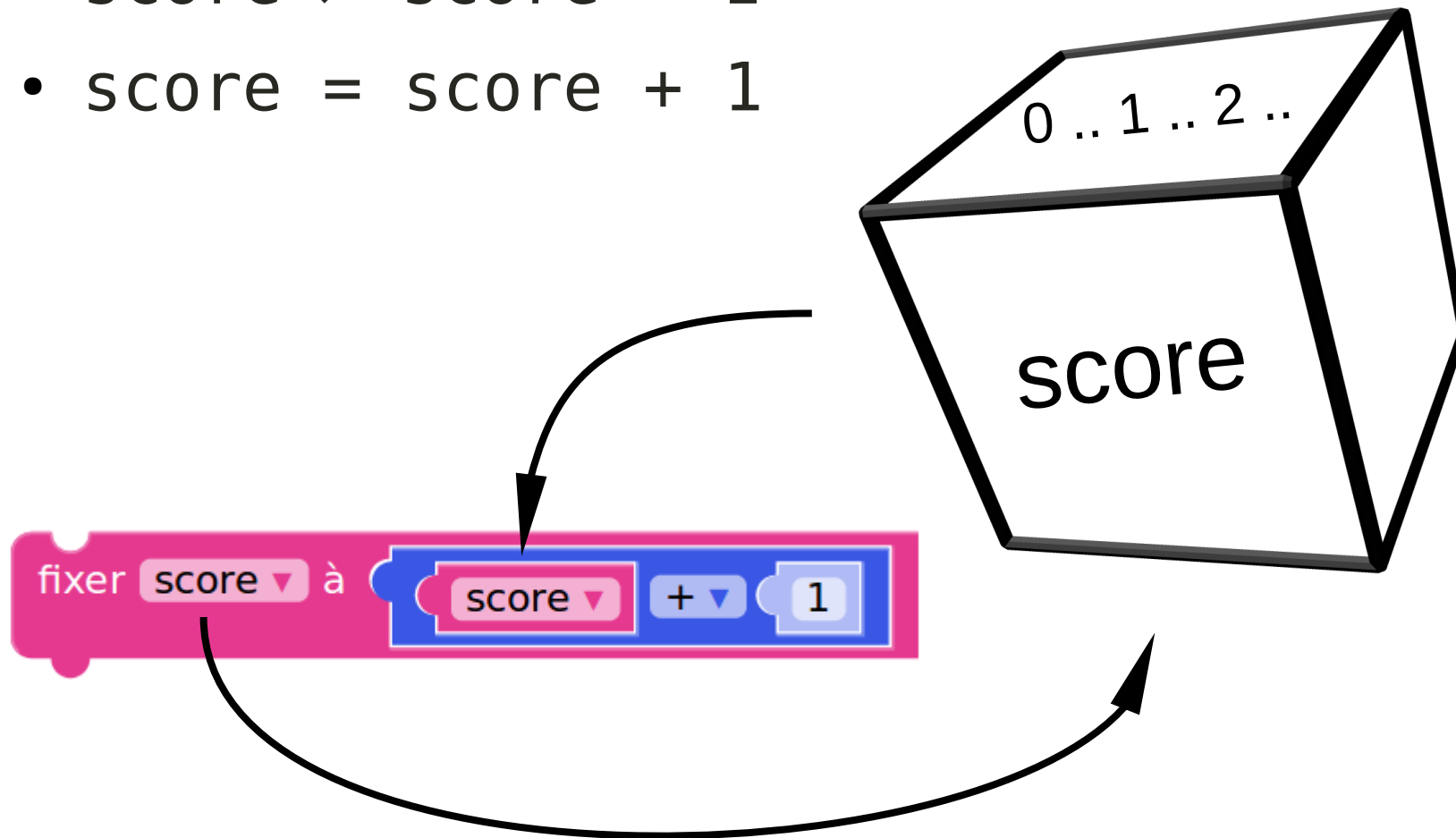
# Question

- Comment ajouter 1 points au score ?
- Chaque fois que drop entre en collision avec la boîte !



# Solution

- $\text{score} \leftarrow \text{score} + 1$
- $\text{score} = \text{score} + 1$



**A nous de jouer !  
Créons notre jeu  
maintenant :-)**

**Merci de votre participation**