

# Scratch, mais encore !

Marielle Léonard



Programmation  
par blocs

Deux approches

Puzzles

vs

Projet

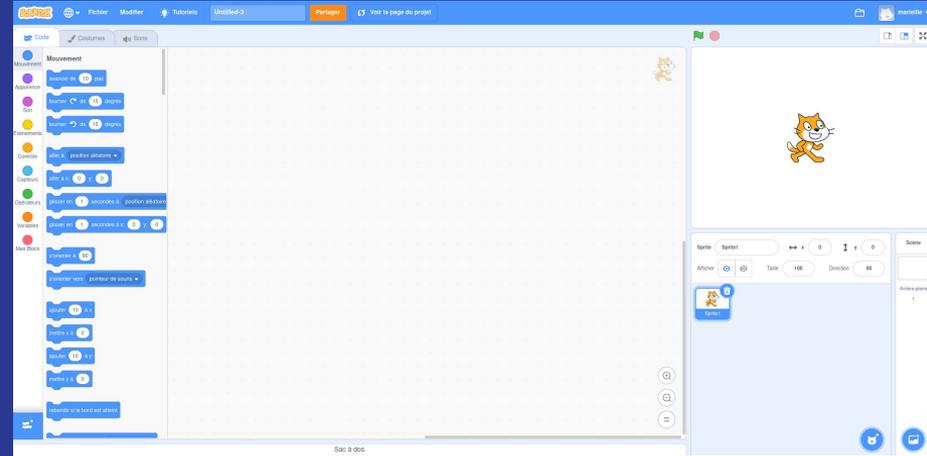
---

# Puzzles



- code.org
- Algorea

# Projet



- Scratch
- Snap!

# Puzzles

- Tâche prescrite, fermée
- Sous-ensemble de blocs disponible
- Évaluation par le système
- Organisés en parcours, avec une progression dans les notions algorithmiques
- Centrés sur la phase de conception d'algorithmes, dans une approche résolution de problème

# Projet

- Tâche ouverte, créative
  - Tous les blocs sont disponibles
  - Évaluation externe au système
  - Organisation et progression pédagogique externe au système
  - Toutes les phases de conception d'un projet
-

# Algorea Serious Game



Un exemple d'approche puzzles

The screenshot shows the Algorea Serious Game interface. The title bar indicates the current level: "Parcours > Parcours officiels > ALGOREA SERIOUS GAME > 2 - Donner plusieurs fois le même ordre au robot > Défis en Scratch > Planter des fleurs". The main title is "Planter des fleurs".

The interface includes a Scratch-style programming area with a "Programme du robot" block. The available blocks are: "tourner à gauche", "tourner à droite", "avancer", "semer une graine", and "répéter 10 fois".

The game area features a 10x5 grid. A green robot is positioned at the bottom right. There are several seeds and flowers on the grid. The instructions state: "Programmez le robot pour qu'il dépose une graine sur chaque tas de terre." and "Attention : le robot ne peut pas accéder à une case si une fleur a déjà poussé dessus. Il ne va quand même pas écraser les fleurs !".

At the bottom, there are playback controls and a progress bar. The footer includes "Réalisé par France-ioi" and "Infos légales | Contactez-nous".

<https://parcours.algorea.org/>

# Les concepts piliers de l'informatique

VERSION ☆

VERSION ☆☆

VERSION ☆☆☆

VERSION ☆☆☆☆

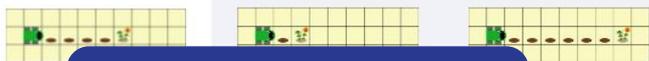
Programmez le robot pour qu'il dépose une graine sur chaque tas de terre.

Le robot ne peut pas accéder à une case si une fleur a déjà poussé dessus. Il ne va quand même pas écraser les fleurs !

Attention ! Le même programme doit fonctionner sur les trois tests.



Représentation  
de la **MACHINE**  
Robot virtuel



**INFORMATION**  
Grille donnée en entrée

pivoter à gauche

pivoter à droite

avancer

semer une graine

fleur devant

répéter jusqu'à

**LANGAGE**  
Scratch

Programme du robot

répéter jusqu'à fleur devant

avancer

semer

**ALGORITHME**  
exprimé en Scratch  
→ **PROGRAMME**

[Lien vers l'article  
de G. Dowek](#)

11/15

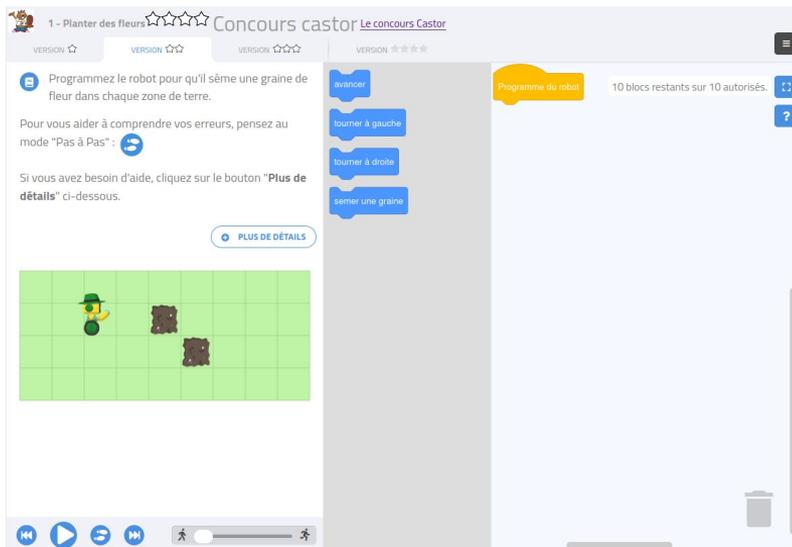
# Algorea Serious Game - Les points forts

- Objectif précis défini pour chaque puzzle
- Différenciation pédagogique autoportée : 4 versions de difficulté croissante pour chaque puzzle
- Outils pour amener les élèves à plus d'autonomie : mode pas à pas, tutoriels
- Interface pour l'enseignant, afin de visualiser l'activité de la classe en temps réel
- Fiche de synthèse pour chaque notion



# Algorea Serious Game - Adossé au concours Algorea

- Même interface
- Très adapté pour l'entraînement aux catégories blanche et jaune



## Catégorie blanche

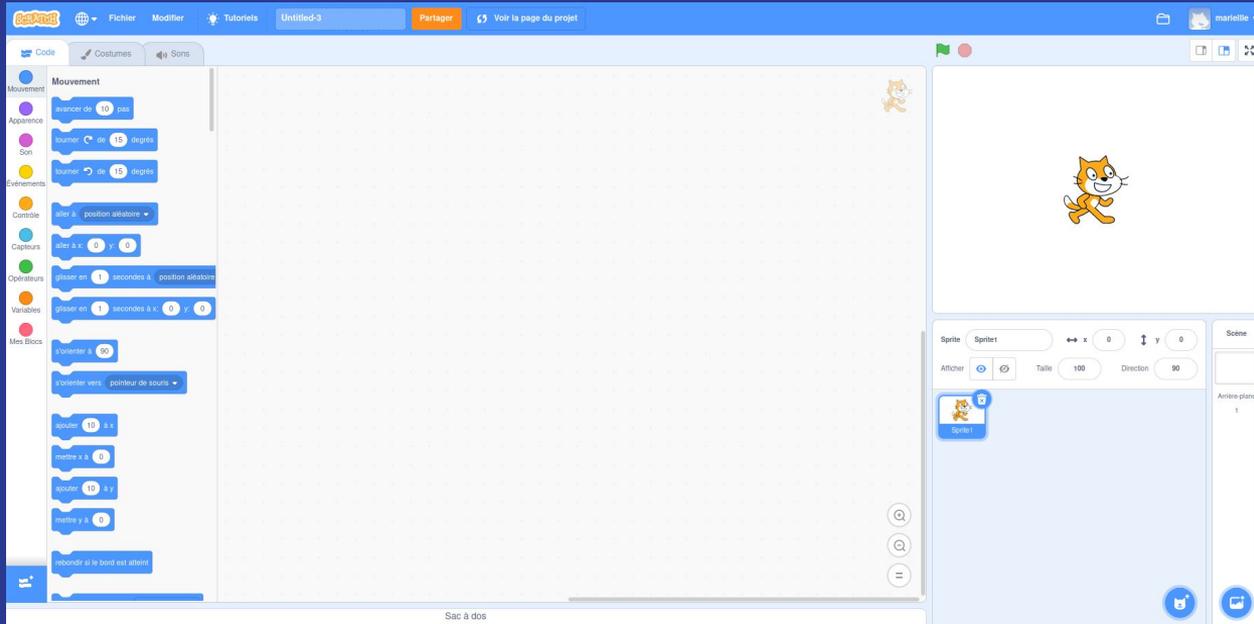
- séquences d'instructions
- appels de fonctions sans paramètres
- boucles répéter simples
- instructions conditions simples

## Catégorie jaune

- appels de fonctions avec paramètres
- boucles répéter imbriquées
- imbrication de boucles et d'instructions conditionnelles
- création de fonctions sans paramètres
- variables simples

# Scratch

Quelques fonctionnalités pour un usage pédagogique



<https://scratch.mit.edu/>

# Puzzles

## Algorea

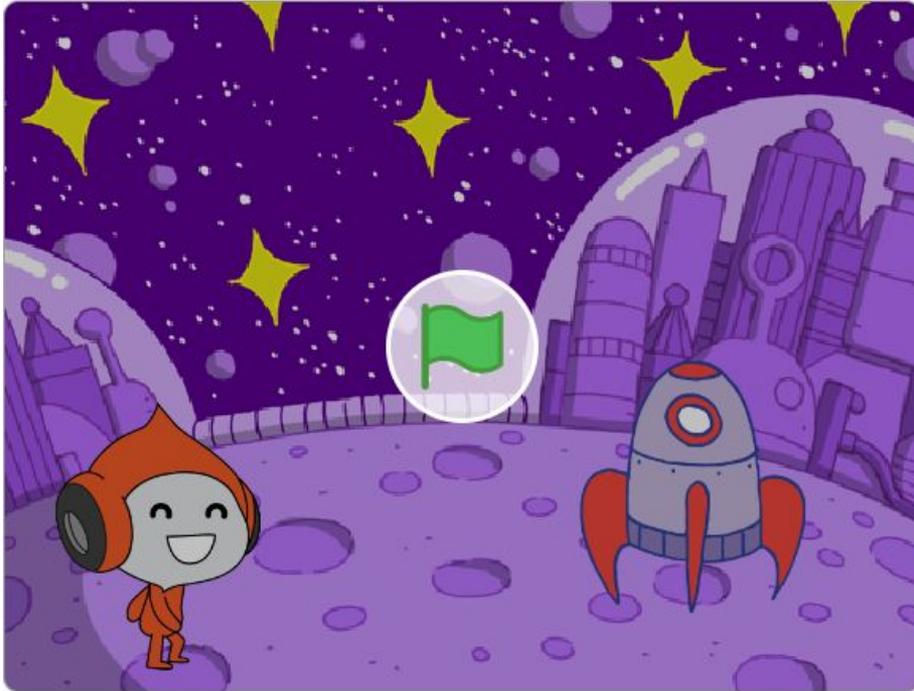
- Langages Scratch, mais aussi Blockly et Python
- Paradigme de programmation impérative (ou séquentielle)

# Projet

## Scratch

- Des éléments issus de plusieurs paradigmes de programmation :
    - événementielle
    - parallèle
    - objet
    - ... et aussi impérative
-

# Exemple de projet - En route vers l'espace



<https://scratch.mit.edu/projects/685483124>

# Puzzles

- Progression et guidage portés par le système, de manière assez automatisée

## Besoin

- de pouvoir reprendre la main (en cours de développement pour Algorea)

# Projet

## Scratch

- Organisation et progression pédagogique externe au système

## Donc besoin :

- d'instrumenter l'accompagnement
  - de guider la progression
-

# Les commentaires

Code Costumes Sons

Mouvement

- avancer de 10 pas
- Apparence
- lourner de 15 degrés
- lourner de 15 degrés
- Événements
- Contrôle
- aller à position aléatoire
- aller à x: -145 y: -84
- Opérateurs
- glisser en 1 secondes à position aléatoire
- Variables
- glisser en 1 secondes à x: -145 y: -84
- Mes Blocs
- s'orienter à 90
- s'orienter vers pointeur de souris
- ajouter 10 à x
- mettre x à -145
- ajouter 10 à y
- mettre y à -84

Quand le drapeau vert est cliqué

aller à x: -140 y: -75

attendre 1 secondes

avancer de 200 pas

Etape 3 : introduire la notion de boucle et l'utiliser pour animer un sprite.  
Defi : donner l'illusion que Pico s'éloigne en marchant (changer de costume) pour rejoindre la fusée et disparaître en entrant dans celle-ci.

Indice : vous aurez besoin de ce bloc pour faire répéter une séquence d'actions à Pico.

répéter 10 fois

ajouter 10 à la taille

Pour changer la taille de Pico, utiliser ce bloc

Sac à dos

Sprite Pico Walking x: -145 y: -84

Atchier Afficher Taille 100 Direction 90

Scène

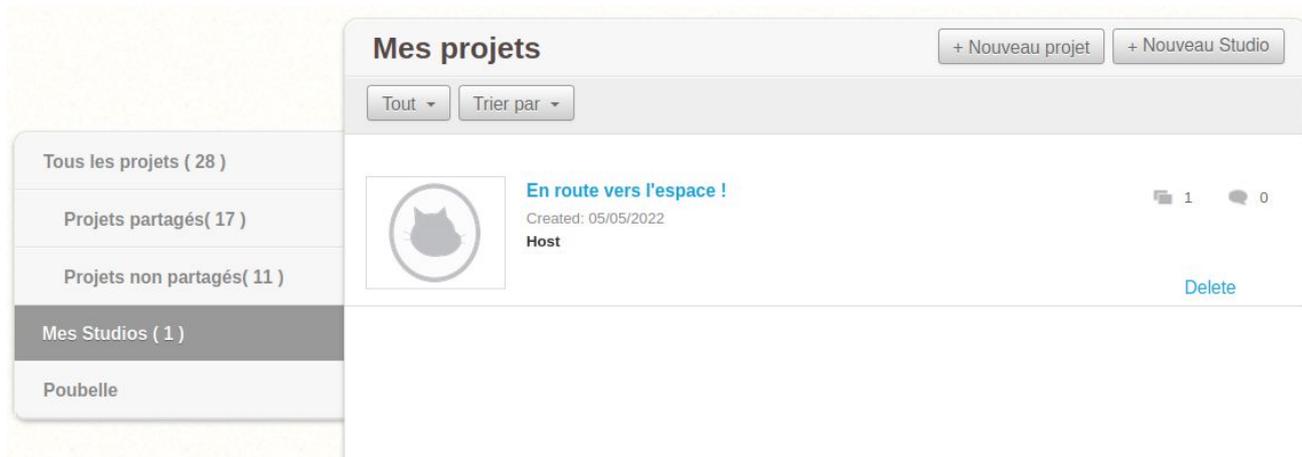
Arrière-plans 2

Pico Walking Rocketship

Quelques usages

- Donner une consigne
- Donner un indice
- Faire un retour contextuel sur un projet

# Les studios



Quelques usages :

- Décomposer un projet et établir une progression
- Regrouper les projets de la classe et ne transmettre qu'un seul lien
- Versioning



Studio pour les élèves :

<https://scratch.mit.edu/studios/31695361/>

Studio avec étapes résolues :

<https://scratch.mit.edu/studios/31559202/>

# Scratch

Quelques ressources complémentaires



# Wiki : documentation en français



## Navigation

Accueil  
Portail de la  
Communauté  
Modifications récentes  
Page au hasard  
Aide  
À propos

## Outils

Pages liées  
Suivi des pages liées  
Pages spéciales  
Version imprimable  
Lien permanent  
Informations sur la page

## Pour les auteurs

Scratch Wiki Fr  
Pages à finir  
To Do List  
Liste des utilisateurs  
Liste des fichiers  
Liste des pages  
Statistiques du Wiki

## Enquête de Satisfaction

Nous effectuons pour le moment des enquêtes de satisfaction. Nous aimerions entendre ce que vous pensez à propos du Wiki.

Répondre

## Montrer et cacher une variable

### Montrer et cacher une variable



Sur [Scratch](#), il est possible d'afficher ou de masquer l'étiquette d'une **variable** directement sur [la scène](#). Cette étiquette, contenant le nom de la **variable** ainsi que sa valeur, peut [s'afficher de différentes façons](#).

Il peut s'avérer pratique de montrer la valeur de la **variable** dans un [projet](#), par exemple pour montrer un score ou le temps restant dans un jeu, mais il peut être aussi pratique de la cacher à un moment donné, par exemple pour cacher une **variable** contenant une réponse à une question que l'utilisateur doit trouver.

Cela fonctionne sur tous les types de **variables** : [locales](#), [globales](#), [cloud](#)...

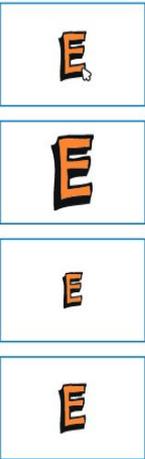
**Info** : L'étiquette de la variable est montrée par défaut lorsque vous créez une variable.

<https://fr.scratch-wiki.info/wiki/>

# Cartes ressources pour les élèves

## Change la Taille

Grossis puis réduis la taille d'une lettre.



Anime un Nom

5



## Change la Taille

scratch.mit.edu

### PREPARE-TOI

Choisis un sprite dans la liste.

Choisis la catégorie Lettres.

Choisis une lettre.

### AJOUTE CE CODE



Entre un chiffre négatif pour rétrécir.

### TESTE TON CODE

Clicke sur ta lettre.

### ASTUCE

Clicke sur ce bloc pour réinitialiser la taille.



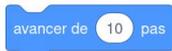
<https://resources.scratch.mit.edu/www/cards/fr/scratch-cards-a-ll.pdf>

# Scratchblock

## Générer des images de blocs

**scratchblocks** v3.6.1 • by blob8108

Export SVG Export PNG



help github **translate** generate previous

Scratch 3.0 ▾ Français ▾ 100%

avancer de (10) pas

répéter (10) fois

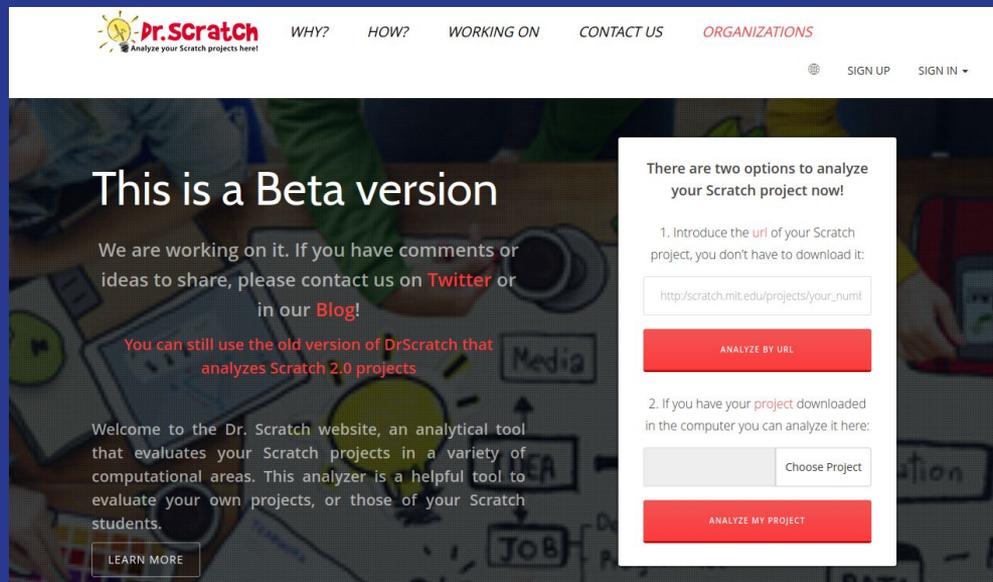
- Une alternative aux copies d'écran, par exemple pour concevoir des supports pédagogiques
- Possibilité de concevoir des images de blocs personnalisés

<http://scratchblocks.github.io/#?style=scratch3&lang=fr&script=>

Documentation associée :  
<https://github.com/iremlp/c2it-article-scratchblocks/blob/master/main.pdf>

# Revenons sur l'évaluation

Dr Scratch, un outil pour évaluer un projet réalisé dans l'environnement Scratch



The screenshot shows the Dr. Scratch website interface. At the top, there is a navigation bar with the logo and menu items: WHY?, HOW?, WORKING ON, CONTACT US, and ORGANIZATIONS. Below the navigation bar, a large white box contains the text: "This is a Beta version", "We are working on it. If you have comments or ideas to share, please contact us on [Twitter](#) or in our [Blog!](#)", and "You can still use the old version of DrScratch that analyzes Scratch 2.0 projects". Below this, there is a "LEARN MORE" button. To the right, a white box contains the text: "There are two options to analyze your Scratch project now!". It lists two options: "1. Introduce the [url](#) of your Scratch project, you don't have to download it:" followed by a text input field containing "http://scratch.mit.edu/projects/your\_num1" and a red "ANALYZE BY URL" button; and "2. If you have your [project](#) downloaded in the computer you can analyze it here:" followed by a "Choose Project" button and a red "ANALYZE MY PROJECT" button.

<http://www.drscratch.org/>

# Puzzles

- Évaluation automatique par le système
- Contraintes du système qui amènent à mobiliser certaines notions algorithmiques
- Attribution d'un score et/ou de badges
- Barre de progression

# Projet

- Évaluation externe au système
  - Peut devenir chronophage pour l'enseignant
  - Pas de contrainte du système à utiliser des notions algorithmiques avancées
  - Maîtrise des notions algorithmiques pas forcément facile à évaluer
-

# Dr Scratch - Un feedback global sur un projet



HELP

DR. SCRATCH(BETA VERSION)



Score: **8/21** [Tweet](#)

The level of your project is... **DEVELOPING!**

You're doing a great job. Keep it up!!:

[Come back to your Scratch project.](#)

## Bad habits

- 0 duplicated scripts.
- 0 sprite naming.
- 0 backdrop naming.
- 0 dead code.

## Project certificate

<https://scratch.mit.edu/projects/685483124>

[Download](#)

## Level up

## Level

Flow control	
Data representation	
Abstraction	
User interactivity	
Synchronization	
Parallelism	
Logic	

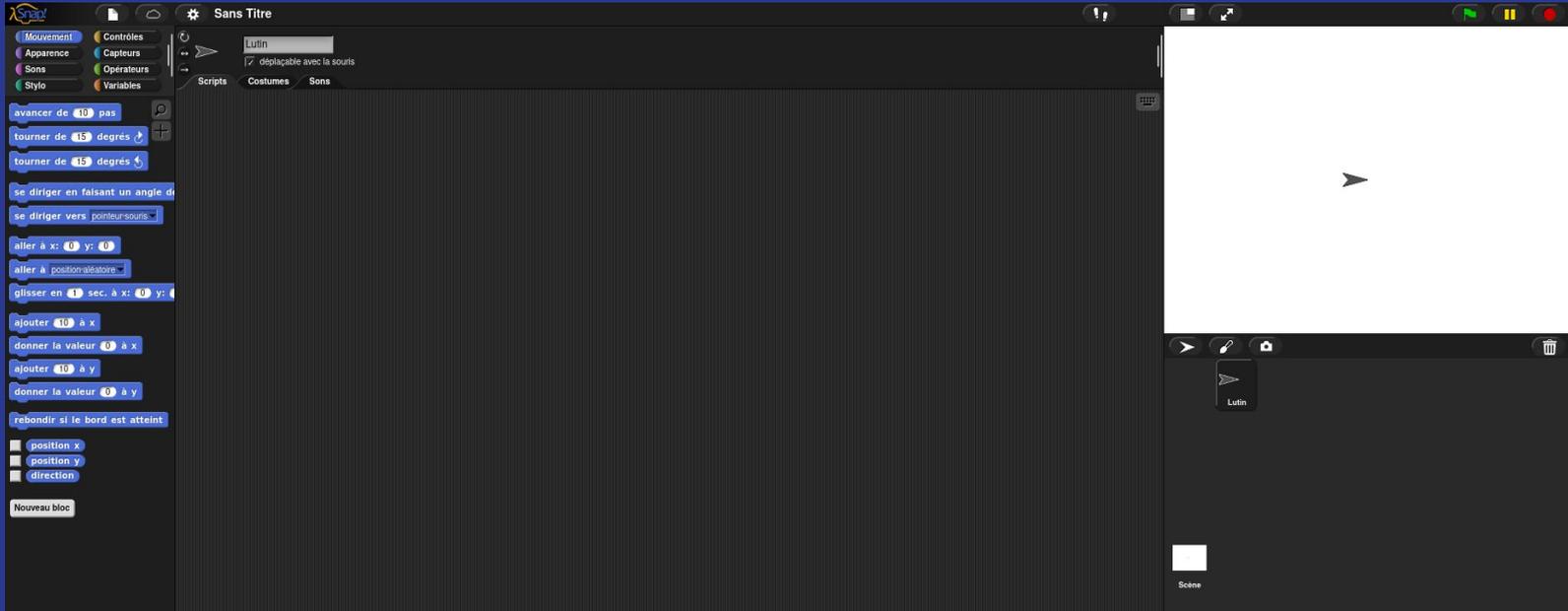
# Dr Scratch - Documenté par un article scientifique

	Niveau de maîtrise de la compétence			
Concept de pensée informatique	Aucune (0)	Basique (1)	Avancée (2)	Experte (3)
Abstraction et décomposition	-	Plus d'un sprite et plus d'un script		
Programmation parallèle	-		Deux scripts déclenchés par l'appui sur une touche, ou par un clic sur un sprite	Deux scripts déclenchés par la réception du même message, par un changement d'arrière-plan, utilisation de clones
Pensée logique	-			
Synchronisation	-			
Contrôle du flux	-	Séquence de blocs		
Interactivité (Entrées / Sorties)	-			
Représentation des données	-	Modification des propriétés du sprite		

[Lien vers l'article scientifique de Moreno-Leon](#)

# Snap!

Une alternative à Scratch



<https://snap.berkeley.edu/>

# Snap!

Aller plus loin dans les concepts de programmation



Fonctions avec  
valeur de retour



Plus de  
fonctionnalités  
sur les listes



Possibilité de rendre  
explicite la variable  
du compteur de  
boucle

# Passer de Scratch à Snap!



Scratch's high ceiling not high enough for you? Snapinator lets you open your [Scratch](#) projects in [Snap!](#), a block-based programming language with a low floor, wide walls, and *no ceiling*.

## Input

Paste a Scratch project URL or ID:

Or load a project from a file:

 Aucun fichier sélectionné.

## Output

Waiting for input...

## Credits

Snapinator 3.0 is written and maintained by [Deborah Servilla](#), inspired by the original Snapin8r by [Kartik Chandra](#).

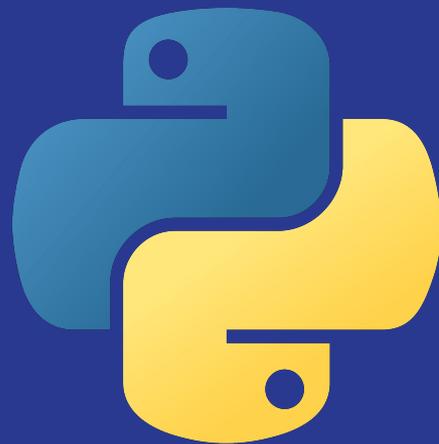
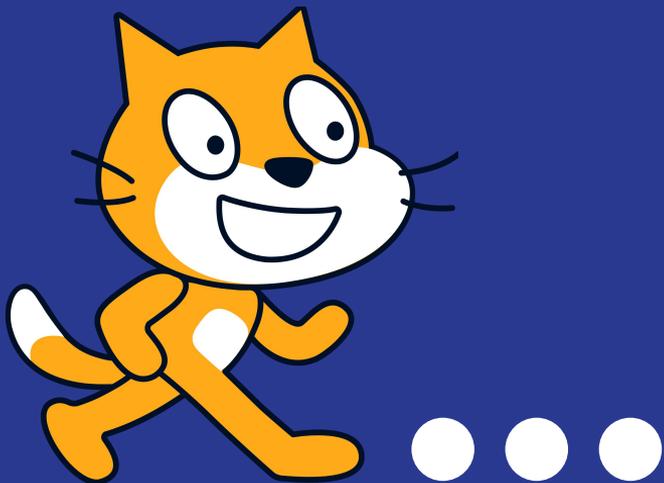
Snapinator is [free software](#), and you can view its [source code](#) on GitHub. It depends on numerous other free/open source projects, including [unzipit](#), [Preact](#), and parts of Scratch 3.0.

Snapinator couldn't exist without Scratch or Snap! ❤️

<https://snapinator.github.io/>

# Passer de Scratch à Python

Ou plutôt de la programmation par blocs à un langage textuel



# Puzzles

- Algorea Serious Game
- Algorea Parcours pour SNT
- Pyrates

# Projet

Ou plutôt “Bac à sable”

- Block2Py
  - Vittascience
-

# Block2Py

Éditer en Blockly, visualiser le programme Python correspondant

- Instructions simples
- Instructions composées
- Expressions arithmétiques
- Expressions logiques
- Expressions de chaînes
- Variables entières
- Variables logiques
- Variables chaînes

```
message = "Hello world"  
print ( message )
```

<https://declercq-c.univ-nantes.io/block2py/>

Entier Booléen Chaîne

OUVRIER ENREGISTRER

```
1 from random import randint  
2  
3  
4 message = "Hello world"  
5 print(message)  
6
```

Exécuter

Hello world

# Vittascience

Éditer avec des blocs ou en python

The screenshot displays the Vittascience web editor interface. At the top left, the logo "vitta science" is visible. The top navigation bar includes links for "Programmer", "Ressources", "Classe", and "Matériel", along with user profile, help, and settings icons. Below the navigation bar, there is a toolbar with a "Nouveau projet" button and several icons for file management and execution. The main workspace is divided into two sections: a block-based editor on the left and a Python code editor on the right. The block-based editor shows a grid with a block labeled "afficher" containing the text "Bonjour !". The Python code editor shows two lines of code: `1 print('Bonjour !')` and `2`. A sidebar on the left contains a search bar and a list of categories: Affichage, Turtle, Graphiques, Numpy, IA, Logique, Boucles, Math, Texte, Variables, and Listes. At the bottom of the workspace, there is a message: `>>> Cliquez sur le bouton Exécuter pour lancer votre programme !`. A URL is overlaid on the workspace: <https://fr.vittascience.com/python/?m ode=mixed&console=bottom>.

# Pyrates

The image shows a screenshot of the 'Pirates' game interface. On the left, there is a sidebar with a title 'PYRATES' in a stylized font. Below the title is a 'Guide de démarrage' section with a 'Niveaux' (Levels) list containing 8 levels, with level 1 selected. A 'Sauvegarder' (Save) button is visible. The 'Objectif' (Objective) is to collect a key and open a chest. The 'Contraintes' (Constraints) state that the program must not exceed 10 lines. The 'Fonctions de contrôle' (Control Functions) list includes: `avancer()` (move forward), `gauche()` (turn left), `droite()` (turn right), and `ouvrir()` (open chest). Below this is a 'Mémo programmation' (Programming Memo) section with a 'python' icon and a list of topics: 'Notions de base', 'Variable', 'Conditionnelle', 'Boucle for', 'Boucle while', and 'A propos du projet'. The main area on the right shows a game scene with a character on a platform, a key, a chest, and a brick wall. The bottom of the interface features an 'Exécuter' (Run) button and an 'Arrêter' (Stop) button, along with a code editor containing the following Python code:

```
1 # Ecrire le code de son programme ci-dessous
2 avancer()
3
```

<https://py-rates.fr/>

Merci de votre écoute !

Contact : [marielle.leonard@univ-lille.fr](mailto:marielle.leonard@univ-lille.fr)