

# Initiation à la programmation

Mercredi 8/11/2023

# Le déroulé de la formation

1

- Que disent les textes?

2

- Mise en situation et expérimentation

3

- Synthèse

4

- Des ressources

# C'est quoi la programmation...



<https://digistorm.app/p/7916290>

# Que disent les textes?

## **Socle commun de connaissances, de compétences et de culture**

### ➤ *Domaine 1 : les langages pour penser et communiquer [...]*

Il sait que des langages informatiques sont utilisés pour programmer des outils numériques et réaliser des traitements automatiques de données. Il connaît les principes de base de l'algorithmique et de la conception des programmes informatiques. Il les met en œuvre pour créer des applications simples. [...]

## **CRCN – Domaine n°4 « Création de contenus »**

### *Niveau 1:*

- Programmer, réaliser un programme simple.

### *Niveau 2:*

- Lire et construire un algorithme qui comprend des instructions simples.

### *Niveau 3:*

- Développer un programme pour répondre à un problème à partir d'instructions d'un langage de programmation.
- Modifier un algorithme simple en faisant évoluer ses éléments de programmation.
- Mettre au point et exécuter un programme simple en commandant un système réel ou système numérique.

# Que disent les textes?

## Programmes C2 – Domaine « Mathématiques »

### Espace et géométrie

*Attendus de fin de cycle: (Se) repérer et (se) déplacer en utilisant des repères et des représentations.*

- Réaliser des déplacements dans l'espace et les coder pour qu'un autre élève puisse les reproduire
- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran.

## Programmes C3 – Domaine « Mathématiques »

### Espace et géométrie

*Attendus de fin de cycle: (Se) repérer et (se) déplacer en utilisant ou en utilisant des représentations.*

- Programmer les déplacements d'un robot ou ceux d'un personnage sur un écran en utilisant un logiciel de programmation.

## Programmes C3 – Domaine « Sciences et technologie »

### Matériaux et objets techniques.

*Attendus de fin de cycle: Décrire le fonctionnement d'objets techniques, leurs fonctions et leurs constitutions*

Repérer et comprendre la communication et la gestion de l'information.

Les élèves découvrent l'algorithme en utilisant des logiciels d'applications visuelles ou ludiques.

# Quels enjeux?

1

- Formation du citoyen

2

- Compréhension du monde

3

- Réfléchir, raisonner, collaborer

4

- Structuration de la pensée



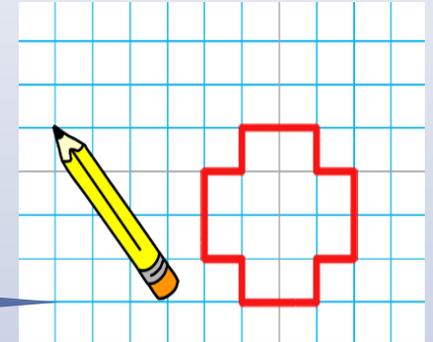


# Créer un langage pour communiquer



Vous disposez d'une feuille quadrillée et d'un crayon.

Vous allez devoir inventer des instructions (consignes, ordres) de façon à contrôler le crayon pour lui faire tracer la croix rouge.



Point de départ

La pointe du crayon est initialement en position d'écriture, sur un nœud du quadrillage appelé "Point de départ".

- 1 - Inventez un **JEU D'INSTRUCTIONS** pour contrôler les déplacements du crayon.
- 2 - Utilisez vos instructions pour écrire un **PROGRAMME** permettant de réaliser le tracé de la croix rouge ci-dessus.



# Des exemples d'instructions



Jeux d'instructions	Jeux d'instructions	Jeux d'instructions	Jeux d'instructions
<b>HDBG</b>	haut, bas, droite, gauche, poser, lever	H B D G C	Déplacement vers le haut Déplacement vers le bas Déplacement vers la gauche Déplacement vers la droite Pointe stylo posée Pointe stylo levée
Programme	Programme	Programme	Programme
<b>DDHDHHGHGGBGBBDB</b>	lever, droite(5), poser, droite(2), haut, droite, haut(2), gauche, haut, gauche(2), bas, gauche, bas(2), droite, bas	C D5 C D2 H D H2 G H G2 B G B2 D B.	Trop long à écrire !



# Synthèse

Vous venez d'inventer **un langage** pour contrôler une machine spécialisée dans la réalisation de tracés sur un quadrillage.

Votre langage est caractérisé par :

- un jeu d'instructions qui constitue le **VOCABULAIRE** du langage ;
- des règles de **SYNTAXE** qui indiquent comment assembler les instructions pour écrire un programme.



Votre langage doit pouvoir :

- Servir de **MOYEN DE COMMUNICATION** entre l'humain – le programmeur – et la machine à contrôler, mais aussi entre différents humains. Il doit donc être compréhensible à la fois par l'humain et par la machine.
- Permettre la **REALISATION DE TOUTES LES TACHES** que la machine est censée effectuer.

# Missions programmation



A vous de programmer!

# Missions: les solutions

## Mission Crêpier psychorigide

L'algorithme permettant de résoudre le problème du crêpier est le suivant :

1. amener la plus grande crêpe en haut de la pile
2. mettre la face brûlée vers le haut
3. retourner toute la pile - la crêpe est rangée
4. recommencer en ignorant les crêpes rangées

## Mission Pixel Art

**1/** Les lettres JO sont coloriées en noir.

**2/** 2 bits: Nord/Sud/Est/Ouest ( $2^2=4$ )

Gauche/Droite/Haut/Bas

00 : Nord / 01: Sud / 10: Est / 11: Ouest

**3/** 3 bits: les jours de la semaine ( $2^3=8$ )

000: lundi

001: mardi

010: mercredi

011: jeudi

100: vendredi

101: samedi

110: dimanche

111

# Solution, Mission créative Scratch Jr

The image shows the Scratch Jr interface. At the top, the Scratch Jr logo and a home icon are visible. Below the logo, there are several asset categories: 'Chat', 'Champignon' (mushrooms), and a plus sign. The main stage displays a house with a green roof, a red door, and a yellow cat character standing on the grass. To the right of the stage, there is a preview window showing the current scene and a plus sign. At the bottom, there is a toolbar with various icons for actions like chat, movement, and sound. Below the toolbar, there are two rows of programming blocks. The first row shows a sequence of blocks for the cat character: a green flag, a 'when green flag clicked' block, a 'say Hello! for 7 seconds' block, a 'run' block, an 'up 1 step' block, a 'turn right 3 degrees' block, a 'down 1 step' block, a 'turn left 3 degrees' block, a 'say Super! for 4 seconds' block, a 'run' block, a 'say Hello! for 4 seconds' block, an 'up 1 step' block, a 'turn right 3 degrees' block, a 'down 1 step' block, a 'turn left 3 degrees' block, a 'say Super! for 4 seconds' block, a 'run' block, and a 'say Hello! for 4 seconds' block. The second row shows a sequence of blocks for the mushroom character: a green flag, a 'when green flag clicked' block, a 'run' block, a 'say Hello! for 4 seconds' block, and a 'pop' sound block.

# Solution, Mission créative Scratch

Pour chaque lutin choisissez le mode de rotation : 



Implémentez le code suivant pour chaque lutin :



# Mission: Oups, j'ai bugué...Solution Labyrinthe 1

```
quand [drapeau] est cliqué
  Réinitialiser
  avancer de 300 pas
  attendre 0.5 secondes
  tourner ↻ de 90 degrés
  avancer de 200 pas
  attendre 0.5 secondes
  tourner ↻ de 90 degrés
  avancer de 100 pas
  attendre 1 secondes
  tourner ↻ de 90 degrés
  avancer de 150 pas
  s'orienter à 90
```

```
définir Réinitialiser
  aller à x: -161 y: -93
  s'orienter à 90
  attendre 0.5 secondes
```

```
quand [drapeau] est cliqué
  Réinitialiser
  avancer de 300 pas
  attendre 0.5 secondes
  tourner ↻ de 90 degrés
  avancer de 200 pas
  attendre 0.5 secondes
  tourner ↻ de 90 degrés
  avancer de 100 pas
  attendre 1 secondes
  tourner ↻ de 90 degrés
  avancer de 150 pas
  s'orienter à 90
```

```
définir Réinitialiser
  aller à x: -161 y: -93
  s'orienter à 90
  attendre 0.5 secondes
```

# Mission: Oups, j'ai bugué...Solution Labyrinthe 2

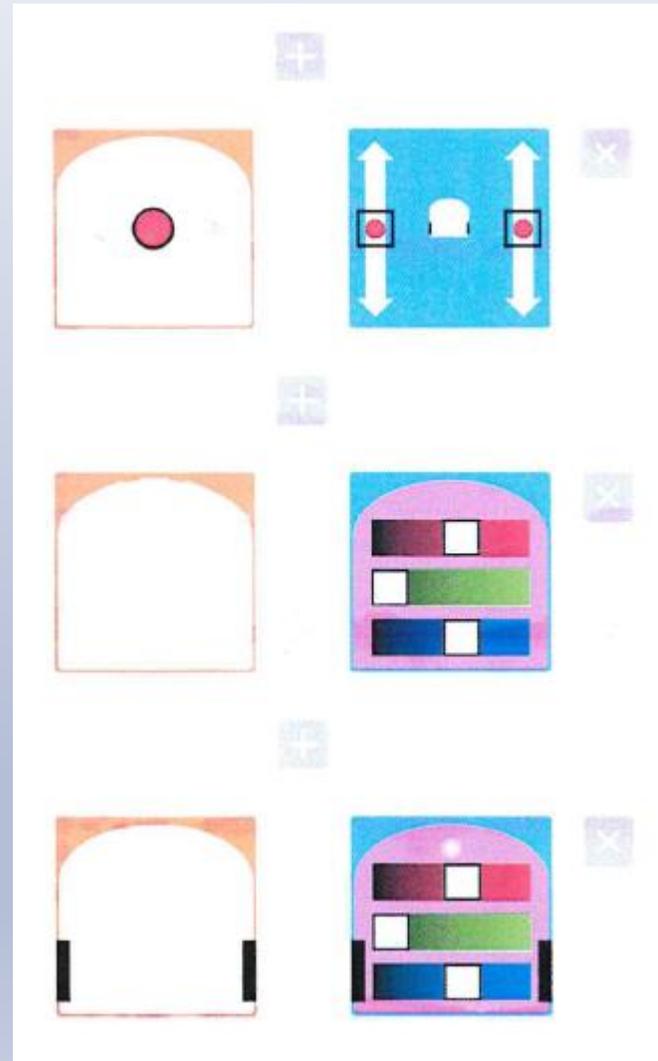
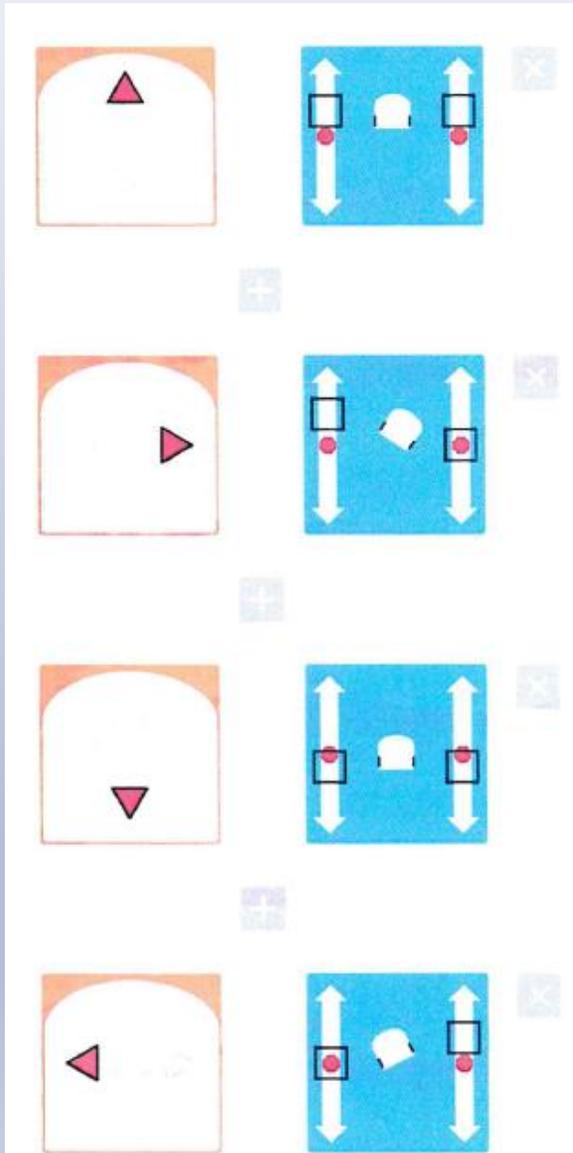
```
quand est cliqué
  Réinitialiser
  attendre 2 secondes
  répéter 4 fois
    avancer de 30 pas
    tourner de 90 degrés
    attendre 0.5 secondes
    avancer de 40 pas
    tourner de 90 degrés
    attendre 0.5 secondes
  avancer de 50 pas
  tourner de 90 degrés
  attendre 0.5 secondes
  répéter 4 fois
    avancer de 30 pas
    tourner de 90 degrés
    attendre 0.5 secondes
    avancer de 40 pas
    tourner de 90 degrés
    attendre 0.5 secondes
  s'orienter à 90
```

```
définir Réinitialiser
  aller à x: -100 y: -53
  s'orienter à 90
```

```
quand est cliqué
  Réinitialiser
  attendre 2 secondes
  répéter 4 fois
    avancer de 40 pas
    tourner de 90 degrés
    attendre 0.5 secondes
    avancer de 40 pas
    tourner de 90 degrés
    attendre 0.5 secondes
  avancer de 50 pas
  tourner de 90 degrés
  attendre 0.5 secondes
  répéter 4 fois
    avancer de 40 pas
    tourner de 90 degrés
    attendre 0.5 secondes
    avancer de 40 pas
    tourner de 90 degrés
    attendre 0.5 secondes
  s'orienter à 90
```

```
définir Réinitialiser
  aller à x: -100 y: -53
  s'orienter à 90
```

# Solution mission Thymio 2



# Synthèse

## Objectifs de cet enseignement :

- contribution aux programmes des C2 et C3 dans les domaines « mathématiques » et « sciences et technologie »
- formation du citoyen du demain
- structuration de la pensée.

## Des concepts à construire :

- langage ;
- programme / algorithme ;
- fonctions test « si...alors » et boucle.

## 2 types de programmation :

- « événementielle » avec des instructions selon une suite chronologique linéaire
- « conditionnelle » qui prend en compte d'éventuelles perturbations extérieures.

## 3 modalités :

- en débranché
- logicielle sur ordinateur ou tablette
- robotique.

## Des tâches variées pour les élèves :

- lire / appliquer un programme ;
- reconstituer un programme ;
- créer un programme ;
- corriger un programme ;
- optimiser un programme ;
- déduire un programme à partir de l'observation d'un fonctionnement.

## Glossaire :

**Le langage informatique** est un moyen de communication entre l'homme et la machine, il a son propre vocabulaire (les instructions) et sa propre syntaxe (agencement des instructions).

**Algorithme** : Un algorithme est la description d'une suite d'étapes permettant d'obtenir un résultat à partir d'éléments fournis en entrée. (CNIL)

**Programme** : c'est un ou plusieurs algorithmes destinés à être exécutés par une machine dans un langage qu'elle comprend.

# Introduction à la pensée informatique

Il est reconnu que la pensée informatique représente un ensemble de compétences essentielles à tout élève au 21<sup>e</sup> siècle, qu'il ou elle envisage ou non une carrière dans l'informatique. Elle implique d'appréhender le monde selon l'approche employée en programmation par les développeurs de logiciels.

Cette approche peut être scindée en **cinq grandes catégories** :

- appréhender un problème et sa solution à différents niveaux (**abstraction**),
- réfléchir aux tâches à accomplir sous forme d'une série d'étapes (**algorithmes**),
- comprendre que pour résoudre un problème complexe il faut le décomposer en plusieurs problèmes simples (**décomposition**),
- comprendre qu'il est probable qu'un nouveau problème soit lié à d'autres problèmes déjà résolus par l'élève (**reconnaissance de formes**),
- réaliser que la solution à un problème peut servir à résoudre tout un éventail de problèmes semblables (**généralisation**).

Par ailleurs, il existe quelques points fondamentaux à comprendre au sujet des ordinateurs :

- Les ordinateurs sont **déterministes** : ils font ce qu'on leur dit de faire.
- Les ordinateurs sont **précis** : ils font exactement ce qu'on leur demande de faire.
- Les ordinateurs peuvent donc être compris ; il s'agit simplement de machines au fonctionnement logique.

# Bilan de la formation

- Qu'est-ce que j'ai appris?
- Qu'est-ce que j'ai retenu?
- Si je ne devais conserver qu'un temps, une activité de la formation, ce serait lequel?
- Quels sont mes besoins?