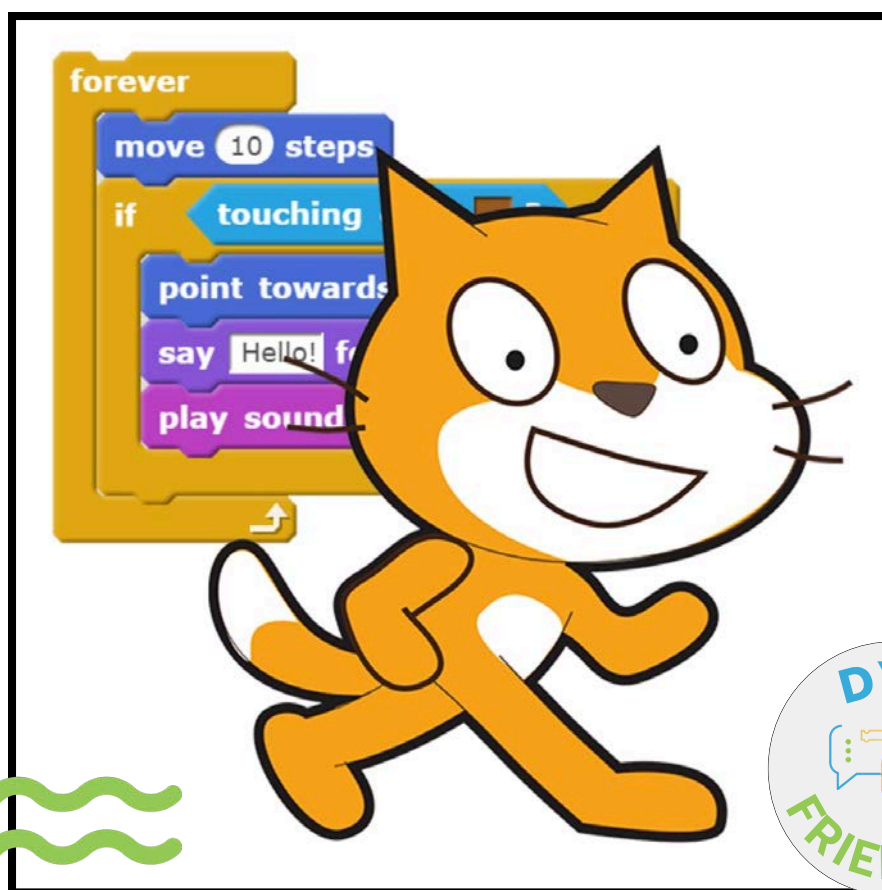


# Scratch

Je m'initie à la programmation



Carnet de l'enseignant

CodeNPlay

# Sommaire



Introduction & Accessibilité .....	p.3
Le Pair Programming .....	p.5
Concepts fondamentaux de la programmation.....	p.7
Découverte avec Code.org.....	p.9
Partie 1 : Je découvre Scratch.....	p.10
Catégories de blocs dans Scratch.....	p.11
Compétences développées avec Scratch.....	p.15
Fiche 1 : Découvrir l'environnement Scratch.....	p.16
Fiche 2 : Les Challenges Scratch.....	p.21
Fiche 3 : Utiliser les costumes.....	p.25
Fiche 4 : Faire nager et faire sauter un sprite.....	p.27
Fiche 5 : Faire du bruit et faire défiler un obstacle....	p.33
Partie 2 : Créer son premier jeu vidéo.....	p.37

## CodeNPlay : Un engagement pour l'accessibilité

Chez Codenplay, nous avons à cœur de rendre nos jeux et contenus accessibles à tous les enfants, y compris ceux qui rencontrent des troubles de l'apprentissage.

Pour ce faire, nos dispositifs ont été relus par Noémie Brans, orthopédagogue et enseignante dans l'enseignement spécialisé et Tatiana Matmat, collaboratrice des Pôles Territoriaux.

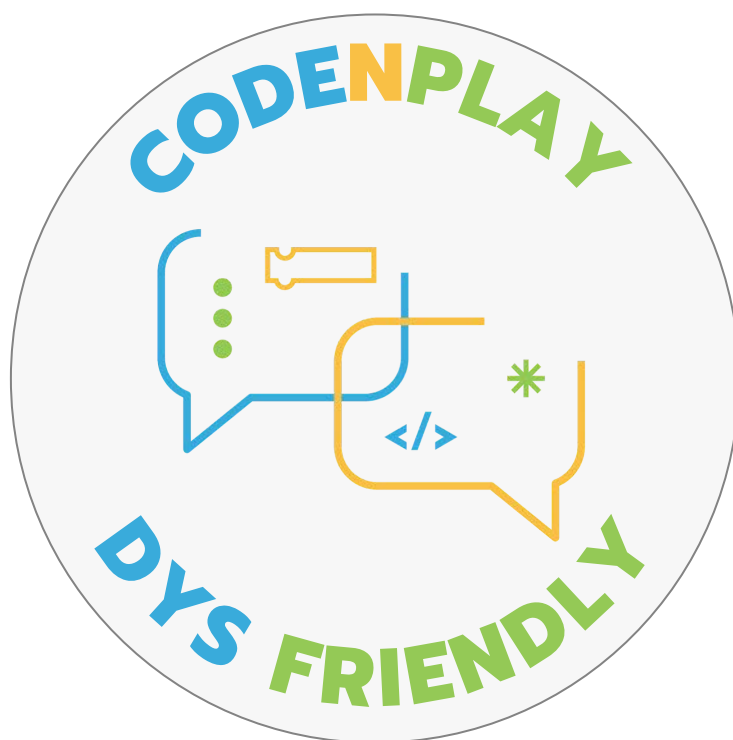
Concrètement, cela se traduit par :

- L'utilisation d'une police d'écriture claire et lisible, sans empattement et de taille suffisamment grande pour faciliter la lecture.
- Un interligne large pour aérer le texte et éviter la fatigue visuelle.
- La mise en gras des mots importants et des actions pour une meilleure compréhension.
- Simplifier le vocabulaire utilisé et à définir les mots difficiles.
- Structurer les textes de manière claire et concise, avec des paragraphes courts et des titres explicites.
- Proposer des contenus variés et stimulants, qui motivent les enfants et favorisent leur apprentissage.

Notre objectif est de permettre à tous les enfants de profiter pleinement de nos dispositifs d'apprentissage.

En plus de ces adaptations, nos contenus sont tous lisibles avec un logiciel de lecture vocale tel que Speechify.

Nous sommes convaincus que l'accessibilité est un enjeu majeur et nous nous engageons à poursuivre nos efforts en ce sens pour que tous les enfants puissent découvrir le numérique et la robotique avec Codenplay.



# PAIR PROGRAMMING :

Le pair programming est une pratique dans laquelle deux programmeurs travaillent ensemble sur un même ordinateur pour coder, concevoir et tester un logiciel. Voici pourquoi travailler de cette façon :

- Favoriser la collaboration et la communication : Le pair programming encourage les programmeurs à travailler ensemble, à partager des idées et à communiquer constamment. Cela peut aider à améliorer la qualité du code en permettant à chacun d'apporter sa propre expertise et son expérience.
- Améliorer la qualité du code : Le pair programming permet à deux programmeurs de travailler ensemble pour trouver des erreurs de code plus rapidement. Ils peuvent également discuter des bonnes pratiques de programmation et prendre des décisions ensemble sur la façon d'aborder un problème complexe.
- Augmenter la productivité : Le pair programming peut en fait augmenter la productivité des programmeurs. Travailler en équipe peut aider à résoudre les problèmes plus rapidement et à réduire le temps nécessaire pour tester et déboguer le code.

# PAIR PROGRAMMING :

- Favoriser l'apprentissage : Le pair programming peut également être un outil pédagogique utile pour les enseignants. En travaillant avec un élève, les enseignants peuvent aider les étudiants à comprendre les concepts de programmation et à développer leurs compétences en programmation.
- Encourager la résolution de problèmes : Le pair programming est une pratique qui encourage la résolution de problèmes en équipe. Les programmeurs doivent travailler ensemble pour trouver des solutions à des problèmes complexes et pour prendre des décisions importantes.

En résumé, le pair programming est une pratique utile pour les enseignants car elle favorise la collaboration, la communication, la qualité du code, la productivité et l'apprentissage. Les enseignants peuvent encourager les étudiants à pratiquer cette méthode en travaillant en binôme pour résoudre des problèmes de programmation ou pour travailler sur des projets de groupe.

# LES DIFFÉRENTS CONCEPTS DE PROGRAMMATION :

## Algorithmes

Un algorithme est une série d'étapes qui résolvent un problème ou accomplissent une tâche. Par exemple, pour faire une tartine, vous pouvez suivre un algorithme simple : prendre une tranche de pain, la toaster, ajouter du beurre et de la confiture. Dans la programmation, les algorithmes sont utilisés pour résoudre des problèmes et effectuer des tâches.

## Boucles

Une boucle est une instruction qui répète une action plusieurs fois. Par exemple, pour compter jusqu'à 10, vous pouvez utiliser une boucle qui commence à 1 et se termine à 10. Dans la programmation, les boucles sont utilisées pour exécuter des instructions répétitives.

## Conditions

Une condition est une instruction qui vérifie si une condition est vraie ou fausse. Par exemple, si vous voulez savoir si un nombre est pair, vous pouvez utiliser une condition qui vérifie si le nombre est divisible par 2. Dans la programmation, les conditions sont utilisées pour exécuter des instructions différentes en fonction de la situation.

# LES DIFFÉRENTS CONCEPTS DE PROGRAMMATION :

## Variables

Une variable est un espace de stockage qui peut contenir une valeur. Par exemple, si vous voulez stocker votre âge, vous pouvez utiliser une variable appelée "âge" et y stocker la valeur 8. Dans la programmation, les variables sont utilisées pour stocker des données et les utiliser plus tard dans le programme.

## Fonctions

Une fonction est un ensemble d'instructions qui résolvent un problème ou accomplissent une tâche. Par exemple, pour dessiner un carré, vous pouvez créer une fonction qui dessine un côté, puis appeler cette fonction quatre fois pour dessiner les quatre côtés du carré. Dans la programmation, les fonctions sont utilisées pour organiser le code et le rendre plus facile à comprendre et à gérer.



# CODE.ORG (1)

## SÉQUENCE 1 : CODE.ORG

### Objectifs :

- Comprendre la disposition des environnements de tutoriel/ d'environnement de programmation.
- Programmer par blocs.

### Déroulement de l'activité

Code.org	<ul style="list-style-type: none"><li>• Amener les élèves sur le site <a href="http://www.code.org">www.code.org</a></li></ul>
Découverte	<ul style="list-style-type: none"><li>• Proposer le cours : le labyrinthe classique : <a href="https://studio.code.org/hoc/1?lang=fr-FR">https://studio.code.org/hoc/1?lang=fr-FR</a></li><li>• Découvrir l'environnement collectivement.</li><li>• Mettre en avant les différentes zones de l'environnement : zone aperçu, zone de progression, zone de consigne et zone de travail.</li><li>• Réaliser un exercice avec l'ensemble de la classe.</li><li>• Créer des binômes de travail : lancer une discussion sur l'avantage de travailler à deux (<b>pair programming*</b>) et découvrir la charte : "travailler en binôme".</li><li>• Demander aux élèves de travailler au moins jusqu'au niveau 6.</li><li>• Découvrir le bloc "répétition".</li><li>• Faire manipuler pour aider les élèves à bien comprendre l'utilisation de ce bloc.</li><li>• Au niveau 9, le bloc gris est obligatoire</li></ul>

répéter 5 fois  
faire

répéter 3 fois  
faire

# PARTIE 1: JE DÉCOUVRE SCRATCH

## Qu'est-ce que Scratch ?

Scratch est un langage de programmation visuel et une plateforme éducative conçue pour initier les élèves à la programmation.

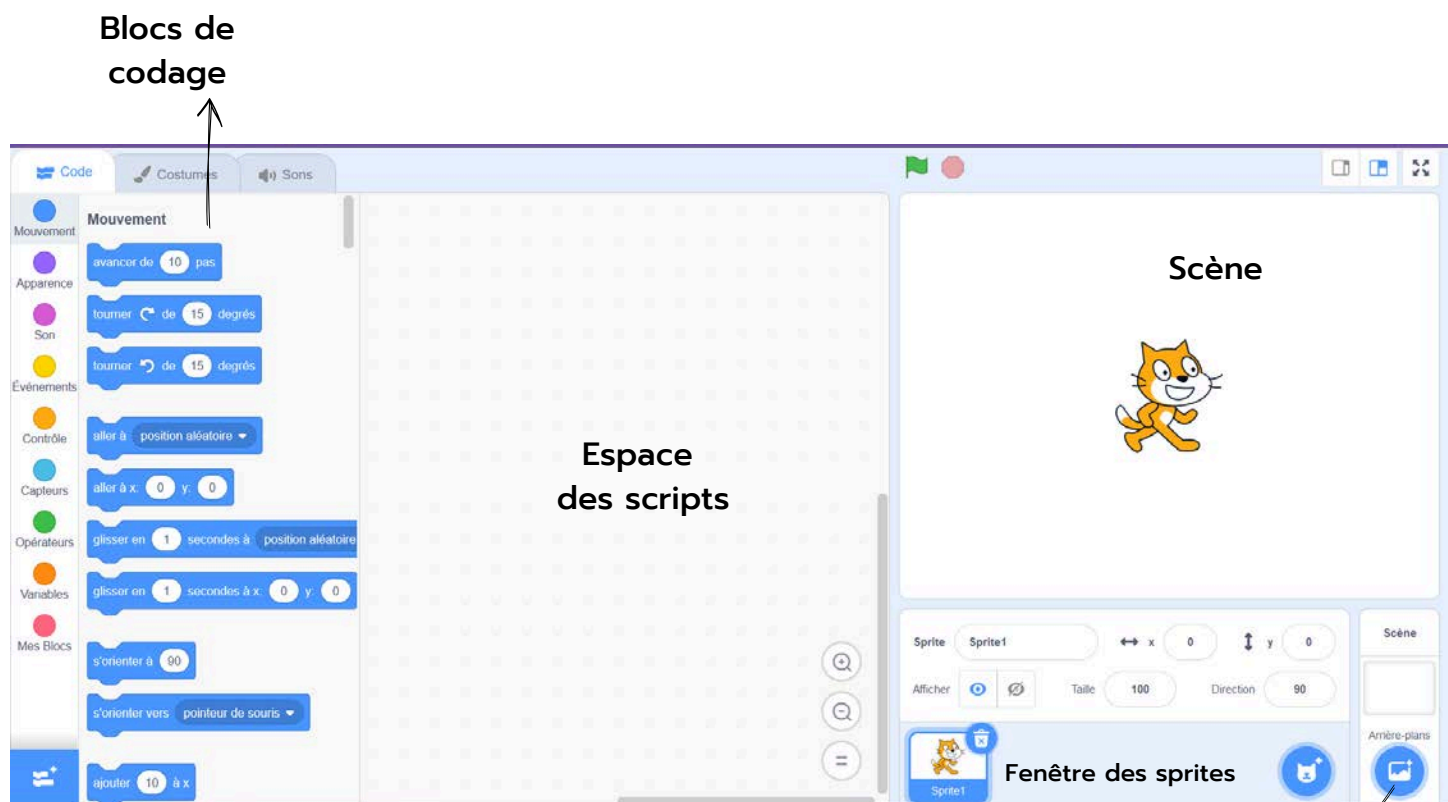
Il est souvent utilisé dans les écoles pour enseigner les concepts de base de la programmation, tels que la logique, la séquence, la boucle et la condition.

Avec Scratch, les enseignants peuvent aider les élèves à créer des programmes interactifs, des histoires animées, des jeux, des simulations et bien plus encore, en utilisant une interface graphique simple et intuitive.

Les enseignants peuvent utiliser Scratch pour enseigner une variété de sujets, tels que les mathématiques, les sciences, les langues et l'histoire, en créant des projets interactifs qui renforcent la compréhension des concepts et stimulent l'imagination des élèves.

Scratch est disponible gratuitement en ligne et est pris en charge sur de nombreuses plateformes.

## L'environnement scratch



## Les blocs de mouvement

avancer de  pas

tourner  de  degrés

tourner  de  degrés

aller à

aller à x:  y:

ajouter  à x

ajouter  à y

direction

ordonnée y

abscisse x

s'orienter en direction de

s'orienter vers

mettre x à

mettre y à

rebondir si le bord est atteint

fixer le sens de rotation

glisser en  secondes à x:  y:

glisser en  secondes à

# Les blocs "Apparence"

de l'arrière-plan

du costume

ajouter  à la taille

arrière-plan suivant

basculer sur l'arrière-plan

basculer sur le costume

dire  pendant  secondes

dire

mettre la taille à  % de la taille initiale

mettre l'effet  à

montrer

penser à  pendant  secondes

ajouter  à l'effet

aller à l'  plan

annuler les effets graphiques

cacher

costume suivant

déplacer de  plans vers l'

penser à

taille

## Les blocs "Son"



## Les blocs "Événement"



## Les blocs "Contrôle"

attendre  secondes

attendre jusqu'à ce que 

créer un clone de

quand je commence comme un clone


répéter  fois

répéter  fois

supprimer ce clone

répéter indéfiniment

répéter jusqu'à ce que 

si  alors

si  alors

stop



# COMPÉTENCES TRAVAILLÉES GRÂCE À SCRATCH

**Français :** Scratch peut aider à développer les compétences en français en créant des histoires interactives ou des dialogues dans des projets. Les élèves peuvent travailler sur la narration, la description, la compréhension écrite et orale, ainsi que l'expression orale et écrite.

**Mathématiques :** Scratch peut être utilisé pour explorer les concepts mathématiques et pour résoudre des problèmes mathématiques en créant des projets interactifs. Les élèves peuvent travailler sur la géométrie, la mesure, les statistiques, l'algèbre, la logique et le raisonnement mathématique.

**Technologie :** Scratch est un outil technologique qui peut aider les élèves à développer leur compréhension des concepts informatiques tels que les variables, les boucles, les conditions, les événements et les fonctions. Les élèves peuvent également apprendre les bases de la programmation informatique en utilisant Scratch.

**Travail en groupe :** Scratch est un outil collaboratif qui peut encourager les élèves à travailler ensemble sur des projets. Les élèves peuvent travailler en équipe pour planifier, concevoir et réaliser des projets Scratch, ce qui peut aider à développer des compétences en communication, en collaboration.

## FICHE 1 : DÉCOUVRIR L'ENVIRONNEMENT SCRATCH

### Objectifs :

- Découvrir l'environnement Scratch : les différentes zones de travail, les menus et fonctions principales.

### Déroulement de l'activité

<b>Introduction</b>	<ul style="list-style-type: none"><li>• Découverte de l'interface collectivement.</li><li>• Rechercher les différentes zones de travail,...</li></ul>
<b>Exploration</b>	<ul style="list-style-type: none"><li>• Former des binômes.</li><li>• Les élèves explorent librement l'environnement. Ils n'ont pas de consignes précises mais les inviter à observer les différents blocs.</li><li>• Mettre en évidence les différentes catégories de blocs.</li></ul>
<b>Mise en commun</b>	<ul style="list-style-type: none"><li>• Faire un retour rapide avec le groupe-classe : Qu'avez-vous découvert ? Quels blocs avez-vous testé ? Quelles sont les catégories de blocs ? ...</li></ul>
<b>Défis simples</b>	<ol style="list-style-type: none"><li>1. Faire avancer le chat de 10 pas.</li><li>2. Faire avancer le chat de 20 pas.</li><li>3. Remettre le chat au centre de la scène.</li></ol> <p>Cet exercice est important car tous les élèves vont être rapidement confrontés au fait que le sprite va sortir de la scène au bout d'un certain nombre de manipulations.</p>



<p><b>Exercices</b></p>	<ul style="list-style-type: none"> <li>• Distribuer la fiche aux élèves, donner les consignes.</li> <li>• Faire collectivement le premier exercice.</li> <li>• Rechercher avec les élèves la signification du bloc</li> <li>• Défi 4 : Faire avancer le chat de 20 pas et lui faire dire bonjour.</li> <li>• Défi 5 : Répéter 3 fois : Faire avancer le chat de 20 pas et lui faire dire bonjour.</li> <li>• Défi 6 : Répéter indéfiniment : Faire avancer le chat de 20 pas et lui faire dire bonjour.</li> <li>• Défi 7 : Même chose lorsque on clique sur le drapeau.</li> </ul>
<p><b>Exercices en binôme</b></p>	<ul style="list-style-type: none"> <li>• Former des binômes et faire réaliser les 4 défis de l'exercice 2</li> <li>• Faire une correction collective au tableau.</li> </ul>



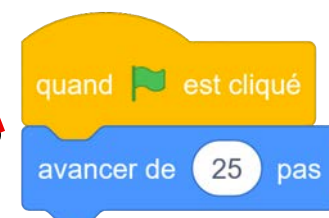
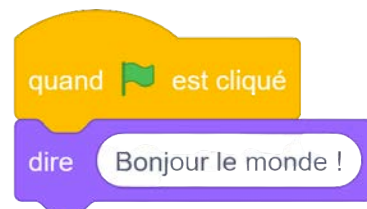
# METTRE LE CHAT "AU PAS"

Tu viens de découvrir l'espace de programmation Scratch, où tu créeras bientôt tes premières histoires, et même ton premier jeu vidéo.

Pour te familiariser à ce nouvel environnement, **relève** les défis suivants !

**Exercice 1 : Relie chaque instruction avec son bloc d'instruction sur Scratch.**

- Faire avancer le chat de 10 lorsque le drapeau vert est cliqué.
- Faire avancer le chat de 25 pas lorsque le drapeau vert est cliqué.
- Faire reculer le chat de 25 pas lorsque le drapeau vert est cliqué.
- Ramener le chat au centre de la scène lorsque le drapeau vert est cliqué.
- Faire monter le chat de 10 pas vers le haut lorsque le drapeau vert est cliqué.
- Afficher une bulle de texte qui dit "Bonjour le monde" lorsque le drapeau vert est cliqué.




- A quoi sert la première instruction

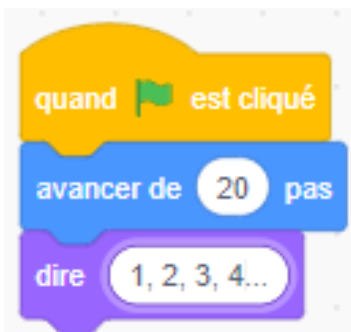


Ce bloc déclenche la lecture de tous les scripts lorsqu'on clique sur le drapeau vert.

**Exercice 2 :** Réalise les défis suivants pour mettre le chat au pas !

Colorie en vert si tu as réussi le défi.

<b>DÉFI 1</b>	Faire déplacer le chat de 20 pas et lui faire afficher une bulle disant "1, 2, 3, 4..."	
---------------	---	---



<b>DÉFI 2</b>	Répéter 3 fois le défi 1, en utilisant un bloc de "Contrôle".	
---------------	---	---



**DÉFI 3**

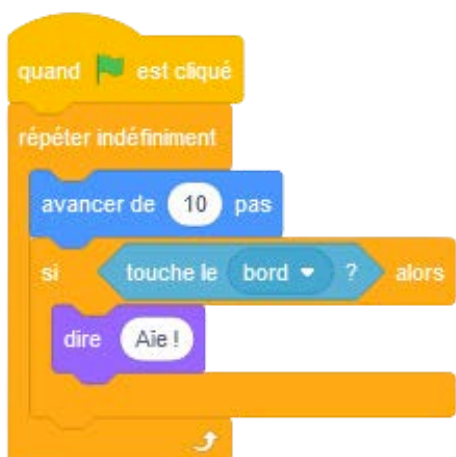
Répéter indéfiniment le défi 1.

**DÉFI 4**

Faire en sorte que le code démarre en cliquant sur la barre d'espace, et non plus sur le drapeau vert.



Faire en sorte que le chat dise "aïe" lorsque le bord est atteint !



## FICHE 2 : LES CHALLENGES

### **Objectifs :**


- Réaliser les challenges données.
- Remettre en ordre des blocs de programmation pour réaliser une action spécifique.
- Comprendre la fonctionnalité de chaque bloc proposé dans les challenges.

### **Déroulement de l'activité**

<b>Introduction</b>	<ul style="list-style-type: none"><li>• Rappel de l'activité précédente : redécouvrir l'environnement Scratch, revoir les différentes catégories de blocs.</li></ul>
<b>Les challenges partie 1</b>	<ul style="list-style-type: none"><li>• Partie 1 :<ul style="list-style-type: none"><li>◦ Lire collectivement les quatre premiers challenges et vérifier la compréhension de chacun. Au besoin, faire bouger soi-même le sprite pour montrer ses déplacements.</li><li>◦ Former des binômes, chaque groupe réalise à son rythme les challenges.</li><li>◦ Vérifier au fur et à mesure le travail des enfants.</li><li>◦ Il est envisageable de proposer des fiches autocorrectives pour une plus grande autonomie des élèves.</li></ul></li></ul>

## FICHE 2 : LES CHALLENGES

### Déroulement de l'activité

<p><b>Les challenges partie 2</b></p>	<ul style="list-style-type: none"><li>• Partie 2<ul style="list-style-type: none"><li>◦ Une fois que les 4 premiers exercices sont faits, faire une mise en commun.</li><li>◦ Les élèves sont-ils tous à l'aise avec le fait de mettre des blocs dans d'autres blocs?</li></ul></li></ul>  <ul style="list-style-type: none"><li>◦ Lancer les élèves sur les quatre exercices suivants.</li><li>◦ Vérifier au fur et à mesure le travail des enfants.</li></ul>
---	--



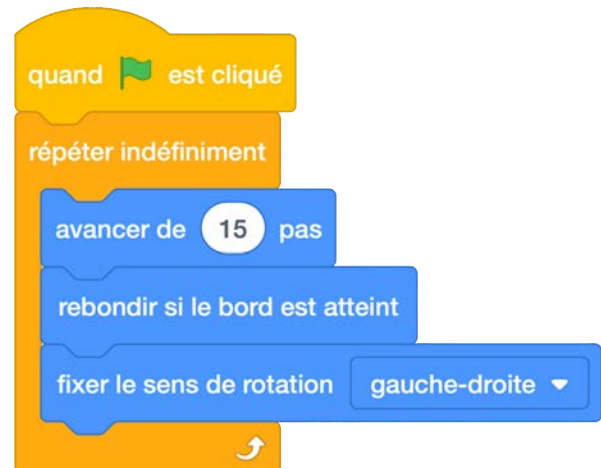
# DES CHALLENGES ...

## PARTIE 1

Faire faire un carré au sprite de 100x100 sur la scène.



Faire faire des allers-retours sur la scène au sprite.



Faire sauter un sprite quand la barre d'espace est pressée



Faire faire des allers-retours sur la scène au sprite et lui faire dire "Aïe" quand le bord est atteint.





# DES CHALLENGES ...

## PARTIE 2

Faire déplacer le sprite quand l'utilisateur appuie sur les flèches du clavier.



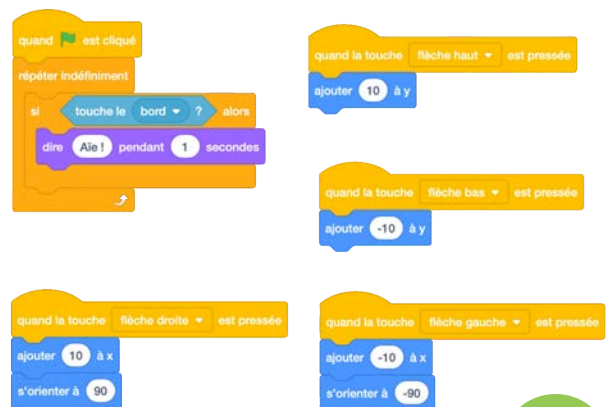
Faire suivre le pointeur de la souris au sprite.



Faire suivre le pointeur de la souris au sprite et lui faire dire "aïe" si on touche le bord.



Faire déplacer le sprite quand l'utilisateur appuie sur les flèches du clavier et lui faire dire "Aïe" s'il touche le bord.





## FICHE 3 : UTILISER LES COSTUMES

### **Objectifs :**

- Apprendre à utiliser les costumes.
- Écrire un programme de déplacement.
- Apprendre à donner une illusion de mouvement dans une animation.

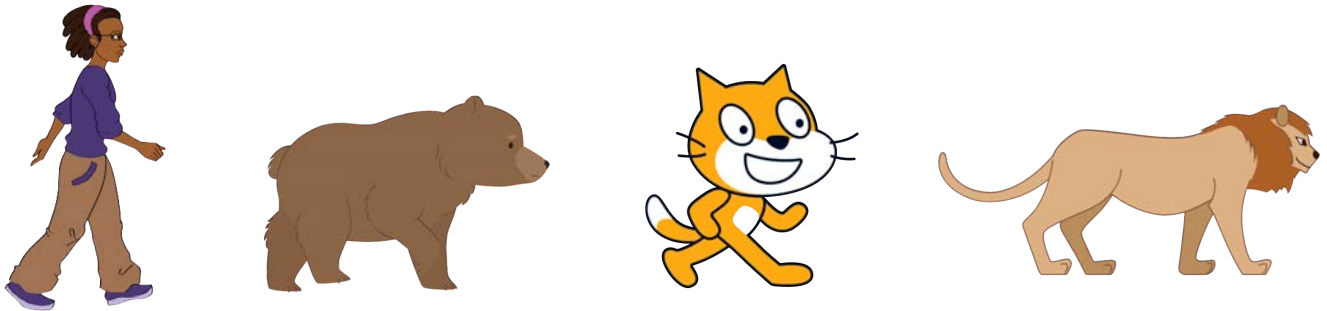
### **Déroulement de l'activité**

<b>Introduction</b>	<ul style="list-style-type: none"><li>• Rappel de l'activité précédente : redécouvrir l'environnement Scratch, est-ce que tout le monde se souvient bien des boucles?</li></ul>
<b>Découvrir les folioscopes</b>	<ul style="list-style-type: none"><li>• Lire le texte dans l'encadré jaune, si possible, montrer la vidéo se trouvant dans le QR code pour montrer aux enfants cette activité.</li></ul>
<b>Créer son folioscope</b>	<ul style="list-style-type: none"><li>• Découper autour des pointillés pour garder les deux positions du chat.</li><li>• Avec un marqueur ou un bic, les enfants peuvent maintenant faire comme dans la vidéo pour créer une illusion de mouvement.</li></ul>
<b>Mouvement dans scratch</b>	<ul style="list-style-type: none"><li>• Faire l'activité "faire marcher son sprite".</li><li>• Demander aux élèves comment faire pour que le sprite se déplace plus vite/plus lentement.</li><li>• Une fois le déplacement fait, essayer que le sprite fasse des allers-retours.</li><li>• S'il reste du temps, les enfants peuvent créer un décor pour leur sprite.</li></ul>

## LES ACTIVITÉS SUR SCRATCH : FAIRE MARCHER SON SPRITE

- Choisis un sprite avec au moins 2 costumes.

Voici quelques exemples :



- Remets ces blocs en ordre pour faire marcher le sprite.



- Fais faire des allers-retours sur la scène au sprite.



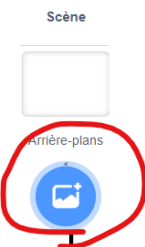
## FICHE 4 : FAIRE NAGER ET FAIRE SAUTER LE SPRITE

### Objectifs :

- Apprendre à déplacer un sprite.
- Écrire différents programmes de déplacement.
- Changer le fond d'écran d'un jeu/d'une animation sur scratch.

### Déroulement de l'activité

<b>Introduction</b>	<ul style="list-style-type: none"><li>• Rappel de l'activité précédente : comment faire pour donner une illusion de déplacement.</li><li>• Se demander s'il n'y aurait pas un autre moyen de faire des déplacements que tout droit (certains ont déjà été vus dans "les challenges")</li></ul>
<b>Faire nager le sprite</b>	<ul style="list-style-type: none"><li>• Demander aux élèves de changer le fond d'écran, ceci peut se faire avec le bouton en bas à droite.</li><li>• Demander aux élèves de sélectionner un des sprites qui leur est proposé.</li><li>• Comment faire pour que les sprites se déplacent? Les élèves peuvent s'aider de la fiche "Je découvre scratch : Faire nager le sprite"</li></ul>
<b>Faire sauter le sprite</b>	<ul style="list-style-type: none"><li>• Un autre déplacement très utile est le saut.</li><li>• Il est conseillé d'utiliser la version simplifiée pour des élèves débutants.</li></ul>

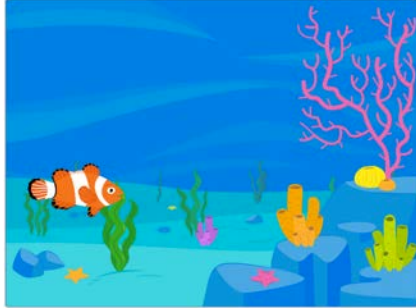




# JE DÉCOUVRE SCRATCH

## FAIRE NAGER LE SPRITE

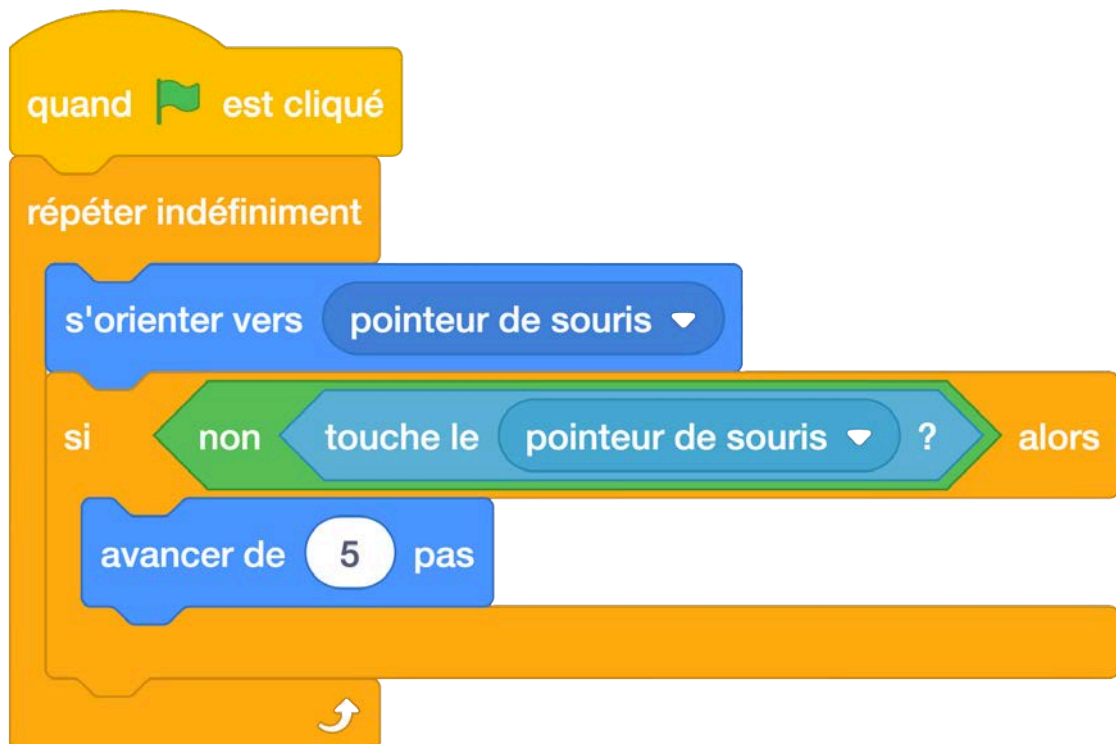
Cet exercice consiste à faire nager un sprite autour de l'écran, en suivant le pointeur de la souris



- Choisis un sprite. Voici quelques exemples :



- Choisis un arrière-plan du monde aquatique.
- Ajoute le code à ton programme; voilà ce que fait chaque bloc.

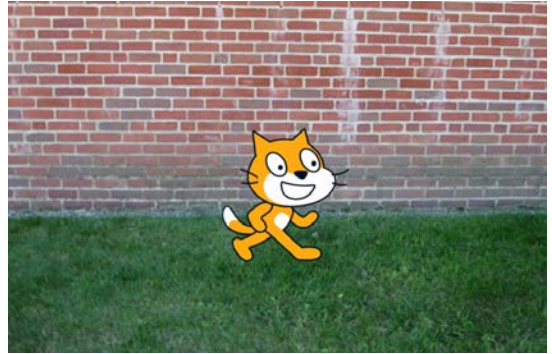




# JE DÉCOUVRE SCRATCH

## FAIRE SAUTER LE SPRITE (VERSION SIMPLIFIÉE)

Cet exercice consiste à faire sauter un sprite.



- Choisis l'arrière-plan de ton choix.
- Ajoute le code à ton programme; voilà ce que fait chaque bloc.

**Quand la touche (espace) est enfoncée** : le code situé sous ce bloc s'exécutera à chaque fois que la touche espace est enfoncée.

**Costume suivant** : Le sprite aura l'air plus réaliste lorsqu'il sautera si son animation change.

**Répéter (10) fois** : Le code à l'intérieur de cette boucle s'exécutera 10 fois.

**Ajouter (10) à y** : Avec ce bloc, le sprite va se déplacer vers le haut.

**Costume suivant** : Comme avant, l'animation de notre sprite changera à nouveau au fur et à mesure qu'elle redescendra.

**Répéter (10) fois** : Le code à l'intérieur de cette boucle s'exécutera 10 fois.





# JE DÉCOUVRE SCRATCH

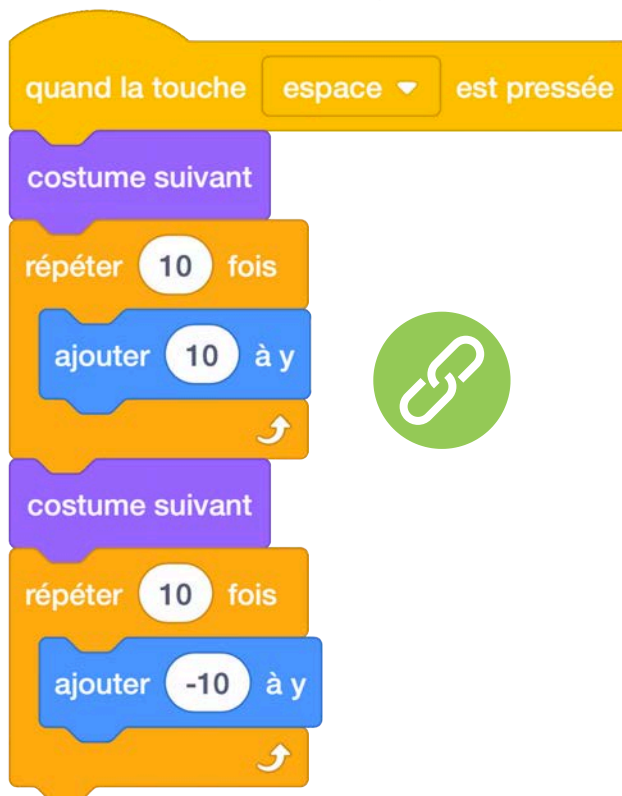
## FAIRE SAUTER LE SPRITE(VERSION SIMPLIFIÉE)

**Ajouter (10) à y** : Avec ce bloc, le sprite va se déplacer vers le haut.

**Costume suivant** : Comme avant, l'animation de notre sprite changera à nouveau au fur et à mesure qu'elle redescendra.

**Répéter (10) fois** : Le code à l'intérieur de cette boucle s'exécutera 10 fois.

**Ajouter (-10) à y** : Avec ce bloc, le sprite va se déplacer vers le bas.

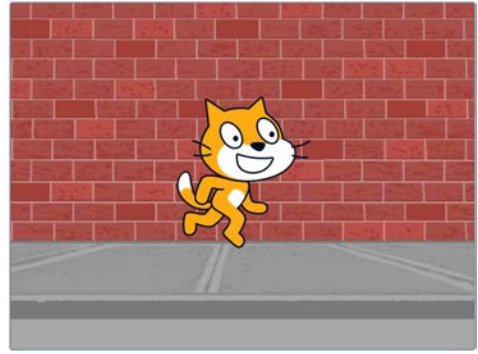




# JE DÉCOUVRE SCRATCH

## FAIRE SAUTER LE SPRITE

Cet exercice consiste à faire sauter un sprite.



- Choisis l'arrière-plan de ton choix.
- Ajoute le code à ton programme; voilà ce que fait chaque bloc.

**Quand la touche (espace) est enfoncée** : le code situé sous ce bloc s'exécutera à chaque fois que la touche espace est enfoncée.

**Costume suivant** : Le sprite aura l'air plus réaliste lorsqu'il sautera si son animation change.

**Glisser (0,2) secondes vers x : (position x) y : (position y + 50)** : Pour comprendre ce bloc de code, nous devons d'abord comprendre les coordonnées x et y. Le mouvement des sprites dans Scratch est contrôlé par une grille invisible, et vous pouvez déplacer des éléments autour de la grille en leur attribuant une valeur x et une valeur y. La valeur x contrôle le placement de gauche à droite et la valeur y contrôle le placement de haut en bas. Avec ce bloc de code, nous disons au sprite de ne pas changer d'emplacement de gauche à droite, et de monter de 50 unités.





# JE DÉCOUVRE SCRATCH

## FAIRE SAUTER LE SPRITE

**Costume suivant** : Comme avant, l'animation de notre sprite changera à nouveau au fur et à mesure qu'elle redescendra.

**Glisser (0,2) secondes vers x : (position x) y : (position y - 50)** : Ce bloc est à l'opposé de notre bloc de glissement précédent ; notre sprite reviendra à sa position d'origine



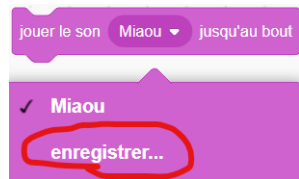


## FICHE 5 : FAIRE UN BRUIT ET FAIRE DÉFILER UN OBSTACLE

### Objectifs :

- Apprendre à utiliser l'option "son".
- Faire rebondir un objet contre le bord de l'écran.

### Déroulement de l'activité

<b>Introduction</b>	<ul style="list-style-type: none"><li>• Rappel de l'activité précédente : est-ce que tout le monde se souvient comment changer de fond ?</li><li>• C'est la dernière activité avant de créer un petit jeu !</li></ul>
<b>Faire du bruit</b>	<ul style="list-style-type: none"><li>• Demander aux élèves de changer le fond d'écran, ceci peut se faire avec le bouton en bas à droite.</li><li>• Demander aux élèves de sélectionner un des sprites qui leur est proposé.</li><li>• Le code est très facile, si vous le désirez les élèves peuvent aussi créer leur propre son grâce à cette icône.</li></ul> 

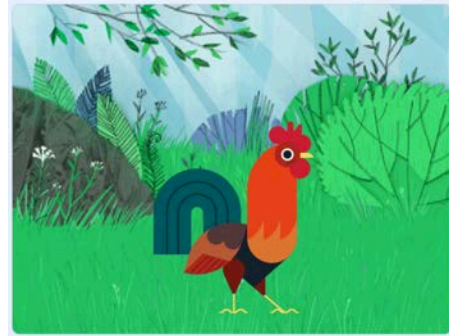
<b>Faire défiler un obstacle</b>	<ul style="list-style-type: none"><li>• Choisissez un sprite à faire rebondir.</li><li>• Les élèves peuvent adapter le décor en fonction du sprite sélectionné (eau pour le poisson, terrain pour la balle, etc.)</li><li>• L'option "rebondir si le bord est atteint" permet au sprite de se retourner une fois arrivé au bord, pour éviter qu'il avance à reculons.</li></ul>
----------------------------------	---



# JE DÉCOUVRE SCRATCH

## FAIRE DU BRUIT AVEC UN SPRITE EN UN CLIC

Cet exercice Scratch consiste à faire émettre un son à un sprite lorsqu'on clique dessus.



- Choisis un sprite. Voici quelques exemples :



**Le code de ce projet ne comprend que deux blocs. Jetons un coup d'œil à ce qu'ils font !**

- **Lorsque ce sprite a cliqué** : Le code sous ce bloc s'exécutera à chaque fois que vous cliquerez sur votre sprite.
- **Jouer du son (...) jusqu'au bout** : ce bloc jouera le son de ton choix.

quand ce sprite est cliqué

jouer le son **rooster** ▼ jusqu'au bout

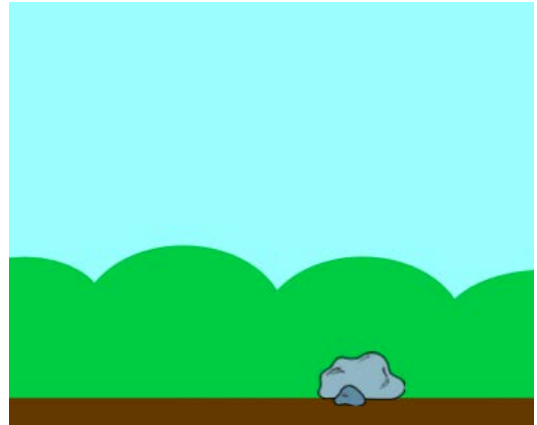




# JE DÉCOUVRE SCRATCH

## FAIRE DÉFILER UN OBSTACLE

Cet exercice Scratch consiste à faire défiler un obstacle sur la scène.



- Choisis l'arrière-plan de ton choix.
- Choisis un obstacle dans les sprites. Voici quelques exemples.



- Essaie de remettre de l'ordre dans ces blocs pour animer ton sprite.



# PARTIE 2: MON PREMIER JEU VIDEO

## **Objectifs :**

- Utiliser les notions apprises lors des activités précédentes et les combiner.
- Créer un jeu vidéo.

## **Déroulement de l'activité**

<b>Introduction</b>	<ul style="list-style-type: none"><li>• Rappel des activités précédentes : focus sur ces 3 animations :</li><li>• Comment sauter ?</li><li>• Comment faire des déplacements ?</li><li>• Comment faire un rebond sur un bord ?</li></ul>
<b>Créer le jeu</b>	<ul style="list-style-type: none"><li>• D'abord, nous allons choisir un objet à éviter et un personnage. Dans notre jeu, nous avons choisi Scratchy et le caillou, mais d'autres sont possibles, comme indiqué sur la feuille d'exercice.</li><li>• Ensuite, nous allons choisir un arrière-plan qui doit être cohérent avec notre jeu (ne pas prendre le fond marin si notre personnage n'est pas un personnage aquatique).</li><li>• Répétez dans l'ordre les étapes du cours pour faire sauter votre personnage, le faire se déplacer, puis rebondir lorsqu'il rencontre un obstacle.</li></ul>



# CRÉER SON PREMIER JEU VIDÉO

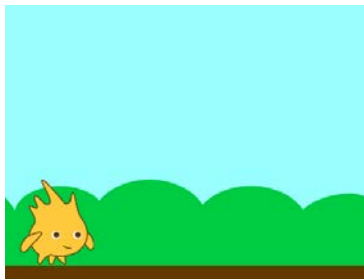
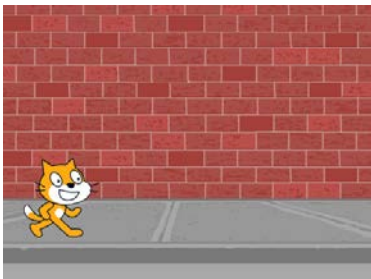
A l'aide de tes connaissances, tu es maintenant capable de créer ton premier jeu vidéo.

Voilà les différentes étapes pour la réalisation :

Réfléchis à l'histoire de ton jeu

- Qui est le personnage principal ?
- Quels sont les obstacles ?

Choisis un sprite (personnage principal) et un arrière-plan.



Choisis un obstacle : une balle ou un rocher (par exemple)

Programme le personnage principal

Programme l'obstacle

Teste ton jeu

**EXEMPLE DE JEU VIDÉO =>**



# SOURCES :



<https://numerisere.web.ac-grenoble.fr/>

Coder un jeu sur Scratch

Découverte de Scratch (1/8) 03:24

Déplacement du lutin (2/8) 03:38

Autres mouvements du lutin (3/8) 02:32

Ajout d'autres lutins (4/8) 03:23

Règles du jeu et musique (5/8) 02:48

Interaction entre lutins (6/8) 03:48

### 3 minutes pour coder

Le site national Lumni propose une série ludique et pédagogique pour apprendre à programmer dès l'école primaire.

🇫🇷 Primàbord / Jan 12, 2023