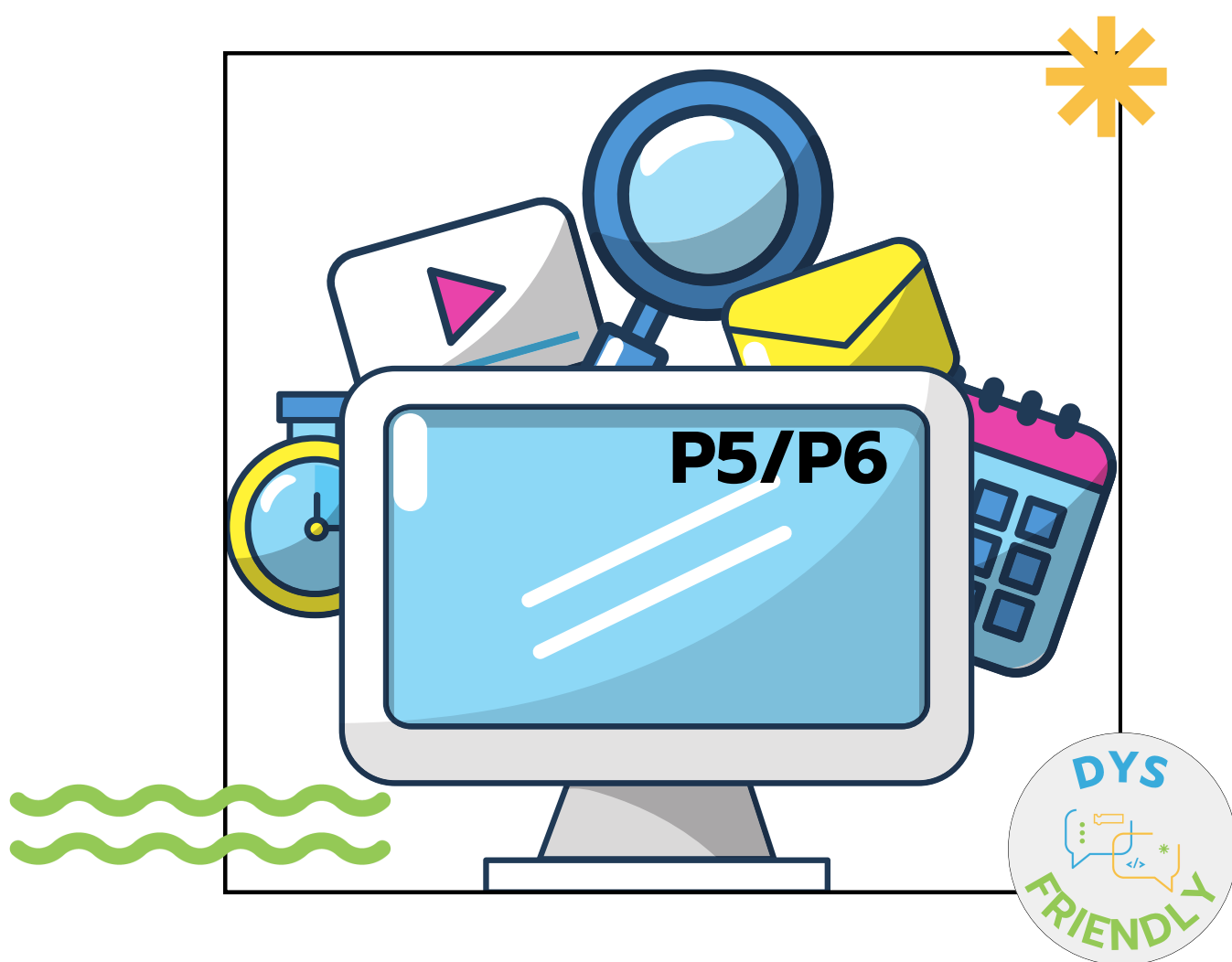


L'ordinateur, internet et moi !



Carnet de l'enseignant

CodeNPlay

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

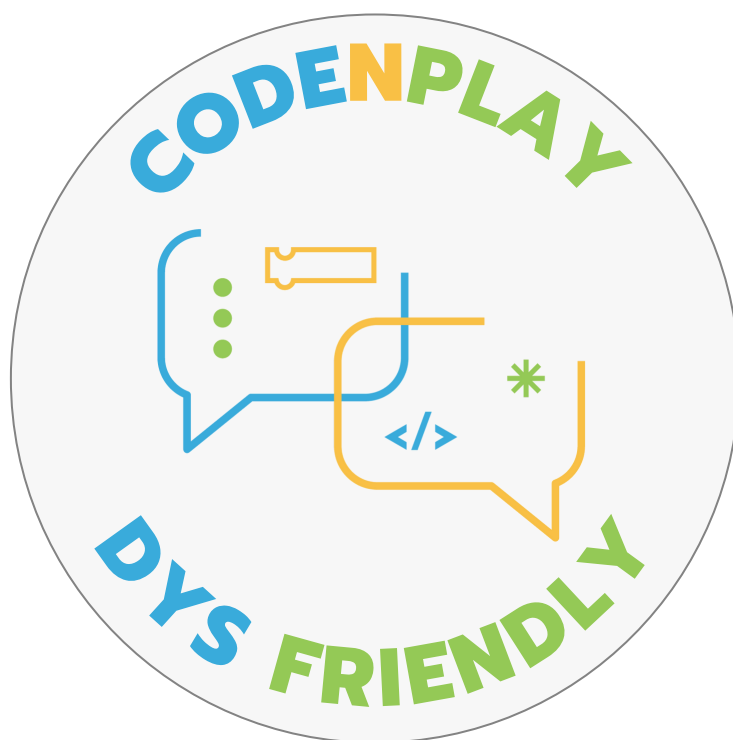
Concrètement, cela se traduit par :

- 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.



Création de contenus



CodeNPlay

Travailler en binôme

Pourquoi travailler à deux ?

Cela nous aide à trouver de meilleures solutions car, en général, il n'y a pas de manière « correcte » de résoudre un problème. Avec le travail en binôme, chaque élève apporte sa perspective, de sorte à construire une solution basée sur une combinaison d'idées.

De plus, avoir deux cerveaux à l'affût des erreurs signifie que les élèves prennent moins de temps à corriger les erreurs éventuelles.

Objectifs :

- créer une charte pour améliorer le travail en binôme
- distinguer la tâche de chacun

Les élèves feront appel à des savoirs, des savoirs-faire et des compétences de français pour cette activité.

Savoir(s) :

- Stratégies/habiletés de compréhension

Savoir-faire :

- Assigner un but à sa lecture.

Compétence(s) :

- Manifester sa compréhension d'un document dont l'intention est d'enjoindre.

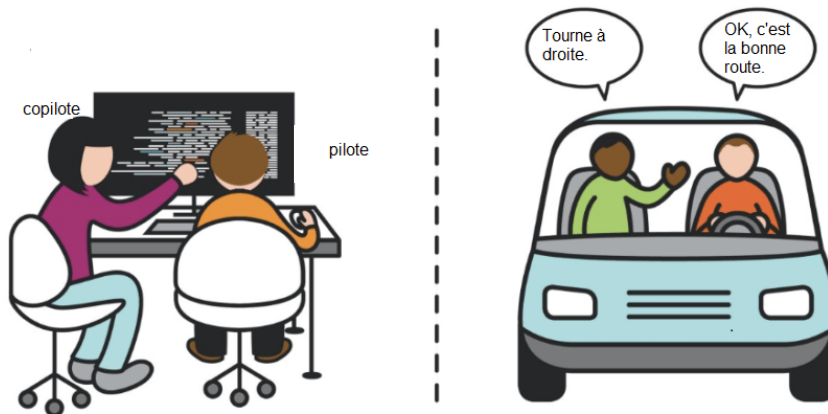
Document(s) annexe(s) :

- Fiche élève : Travailler en binôme
- Annexe 1

Déroulement de l'activité

Situation mobilisatrice : Annoncer aux élèves que lorsque nous travaillerons sur des ordinateurs ou sur des tablettes, nous allons régulièrement travailler en binôme.	
Questionnement	Quels sont les avantages de travailler à deux ?
Mise en commun	Réponses attendues : collaboration, aide, trouver des solutions à deux, ...
Observation	Montrer cette illustration (v. annexe 1) aux élèves et la faire décrire : Faire émerger que chaque membre du binôme a un rôle. Dans un binôme, il y a le rôle de pilote et le rôle de copilote.
Recherche	- Distribuer la fiche "Travailler en binômes". - Proposer toutes les tâches mélangées et demander de les placer au bon endroit. - Correction collective.
Prolongement	A l'aide d'un logiciel de traitement de texte ou d'un outil de mise en page (ex: Canva), créer une affiche pour la classe.

Travailler en binôme



COPILOTE

- 1 Assieds-toi à côté du pilote.
- 2 N'utilise pas la souris, le clavier/ la tablette.
- 3 Réponds aux questions du pilote.
- 4 Signale les problèmes.
- 5 Fais des suggestions.
- 6 Accorde-toi avec le-la pilote sur le travail réalisé.

PILOTE

- 1 Assieds-toi devant l'ordinateur.
- 2 Utilise la souris, le clavier/ la tablette.
- 3 Effectue les tâches demandées
- 4 Dis ce que tu fais à voix haute.
- 5 Écoute les conseils du-de la copilote.
- 6 Accorde-toi avec le-la copilote sur le travail réalisé.

Logigramme et programmation : Activités débranchées

Objectifs :

- Apprendre à décomposer une tâche en étapes séquentielles
- Savoir représenter ces étapes sous forme de logigramme

Référentiel FMTTN**Création de contenus****Savoir : Programmation et logigrammes****Attendus :**

- Représenter les symboles conventionnels d'un logigramme.
- Associer une activité du quotidien à un algorithme et vice-versa
- Expliquer les termes dont algorithme, logigramme, condition, boucle, langage de programmation.
- Différencier algorithme et programme.

Savoir-faire : Lire et écrire un algorithme simple

Attendus : Verbaliser un extrait ou l'entièreté d'un logigramme séquentiel de déplacement, écrire un logigramme séquentiel de déplacement.

Document(s) annexe(s) :

- Logigramme et programmation (exercices et synthèse)
- Annexe 2

Déroulement de l'activité

Situation mobilisatrice :

Demander aux élèves si un robot est intelligent.

Réponse attendue : Non

Les robots ne peuvent suivre que des instructions claires et précises. Ils suivent exactement les instructions, même si elles sont erronées.

Le jeu du robot

- Choisir un enfant pour être "le robot" et un autre pour être le "programmeur".
- Le "programmeur" doit donner des instructions claires au "robot" pour accomplir une tâche simple comme se rendre d'un point A à un point B dans la classe.

Exemples d'instructions : avancer de 3 pas, tourner à gauche, ...

- Faire l'exercice avec 2 à 3 groupes.
- Discuter collectivement de l'importance de donner des instructions claires et précises.

Ecrire son premier algorithme

Demander aux élèves de rédiger les instructions les plus claires et les plus précises pour que le robot arrive au point B. (voir annexe 1). Dire aux enfants que le robot est éteint.

- Faire des groupes de 2.
- Travail des élèves
- Faire une mise en commun. (Plusieurs chemins sont possibles, mais si les élèves n'ont pas précisé qu'il faut allumer et éteindre le robot, le code est erroné.)

Exercices

- Lire les consignes de la fiche élève.
- Travail individuel
- Correction collective

Synthèse

- Lire collectivement la synthèse écrite pour fixer les concepts travaillés.

Prolongements

- Découverte de la programmation par blocs.

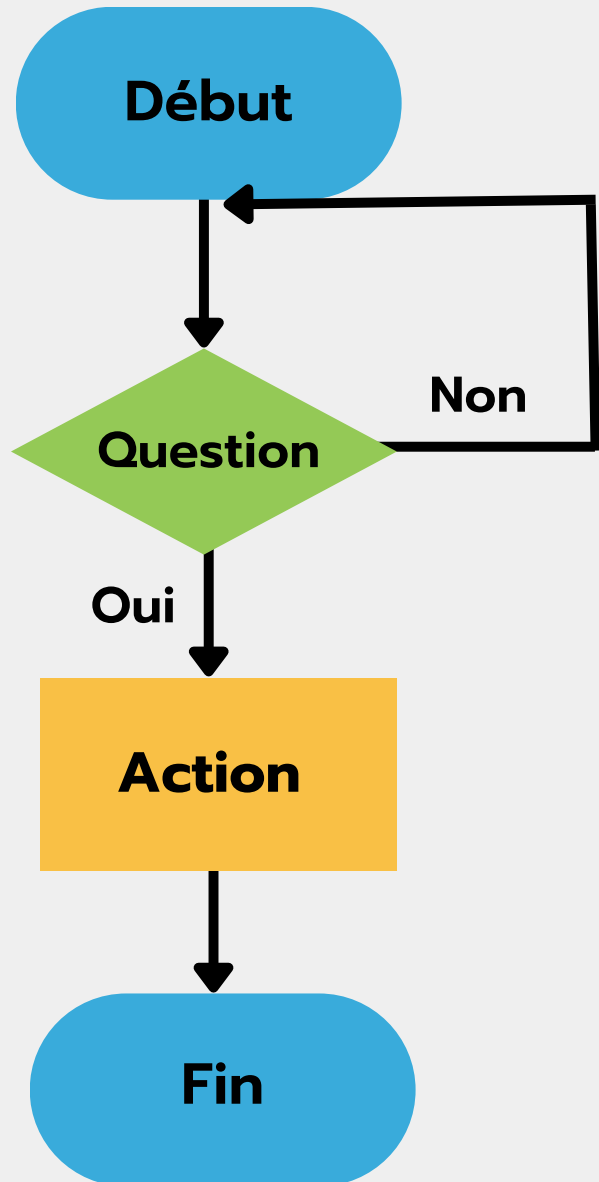
Note théorique

Un logigramme est une représentation visuelle d'un ensemble d'opérations qui formalise un algorithme. En d'autres termes, il s'agit d'un schéma graphique qui illustre un algorithme de manière claire et accessible à tous.

Un logigramme commence toujours par un début et une fin.
On utilise une forme ovale.

Un losange représente une question qui nécessite une réponse par oui ou par non.

Le rectangle fait référence à l'action à effectuer.



Comprendre les boucles**Objectifs :**

- Comprendre le concept de boucle en programmation.

Référentiel FMTTN**Création de contenus****Savoir : Programmation et logigrammes****Attendus :**

Expliquer les termes dont algorithme, logigramme; condition, boucle, langage de programmation.

Document(s) annexe(s) :

- Blocs de construction (Lego, Duplo, Kapla, ...)
- Cartes d'instruction (imprimées et plastifiées)

Déroulement de l'activité

Situation mobilisatrice :

Demander aux élèves de rappeler la définition d'un algorithme.

Expliquer que l'on va réaliser une activité pour mieux comprendre un concept lié à la programmation

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.

Le jeu de la tour de blocs de construction

Le but : reconstituer la tour qui est éloignée, hors de la vue de la zone de construction.

- Constituer des binômes

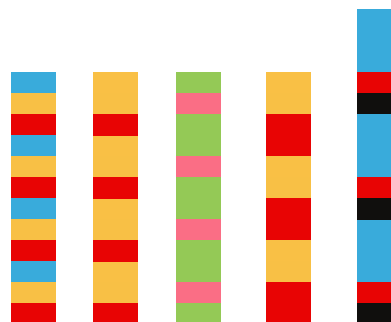
Pour chaque binôme, deux rôles :

Le constructeur : construit la tour grâce aux indications de son partenaire, ne voit pas le modèle.

Le messenger : va voir le modèle et explique à son partenaire comment construire.

- Première phase du jeu
- Phase de structuration : faire une mise en commun : "Quelles sont les informations importante à transmettre à son partenaire ? Comment optimiser le message donné ?
- Faire déconstruire la tour pour faire émerger les notions de motif et de répétition.

Exemples de tours :



Exercices de décodage	• Pour pouvoir communiquer avec un robot ou un ordinateur, il faut utiliser un langage informatique. • Chaque élève reçoit une carte avec un code. (v. annexe 3). Individuellement, il va devoir construire la tour en respectant le code. • Distribuer la fiche élève. • Demander de réaliser les exercices individuellement. • Faire une correction collective. • Définir collectivement ce qu'est une boucle : <i>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.</i>
-------------------------------------	--

Activité créée à partir de ce travail :

[*Scénario pédagogique MOTIF.. MOTIF.. Marielle Léonard, Yann Secq, Yvan Peter, Julian Alvarez, Cédric Fluckiger - Université de Lille](#)

Découvrir la programmation par blocs**Objectifs :**

- Comprendre les concepts de base de la programmation par blocs.
- Apprendre à décomposer une tâche en étapes simples et à les organiser.

Référentiel FMTTN**Création de contenus****Savoir-faire : Lire un programme simple**

Attendus : Lire un programme séquentiel de déplacement
Lire un programme intégrant une boucle et une condition

Savoir-faire : Écrire un programme simple

Attendus : Traduire un programme séquentiel de déplacement, à l'aide de blocs de codes proposés.

Savoir-faire : Identifier des éléments relatifs à la programmation et aux logigrammes :

Attendus : Identifier une suite d'opérations qui peut être remplacée par une boucle.

Document(s) annexe(s) :

- Fiche élève : Je m'initie à la programmation par blocs

Matériel :

- Un ordinateur, tablette pour 2 élèves
- Connexion internet

Au préalable, mettre en favori le lien du parcours d'initiation à la programmation. <https://studio.code.org/hoc/1?lang=fr-FR>

Pourquoi apprendre à coder ?

Résoudre des problèmes dans un cadre motivant :

Le codage permet de trouver des solutions innovantes à divers problèmes, ce qui peut être extrêmement satisfaisant et stimulant.

Se familiariser avec la méthode essai et erreur :

Le processus de codage implique souvent des tests et des corrections, ce qui aide à développer une approche résiliente et méthodique pour surmonter les obstacles.

Développer l'anticipation :

Programmer exige de prévoir les comportements du code et de prévenir les erreurs, renforçant ainsi la capacité à anticiper et planifier.

Structurer la pensée :

Le codage aide à organiser les idées de manière logique et séquentielle, améliorant ainsi les compétences de raisonnement et de planification.

Développer ses capacités de séquençage :

Apprendre à coder nécessite de suivre des étapes précises pour atteindre un objectif, ce qui améliore la capacité à ordonner et structurer les actions de manière cohérente.

Renforcer l'imagination et la créativité :

La programmation n'est pas seulement technique, elle permet aussi de créer des projets originaux et innovants, stimulant ainsi la créativité.

Développer son sens logique :

La résolution de problèmes et l'élaboration de programmes nécessitent une pensée logique rigoureuse, ce qui renforce cette compétence.



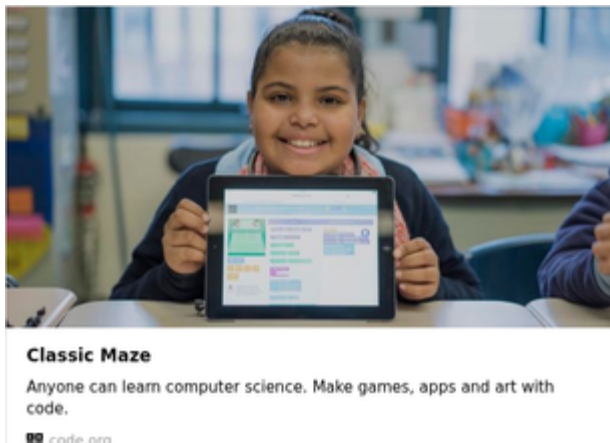
Qu'est-ce que le code ?

C'est la transcription d'une information en échangeant l'écriture courantes en signes conventionnels (ex. : binaire)

Faire ses premiers pas en programmation

Si vous n'avez jamais pratiqué la programmation par blocs, il est recommandé de commencer à vous initier en utilisant les plateformes suivantes :

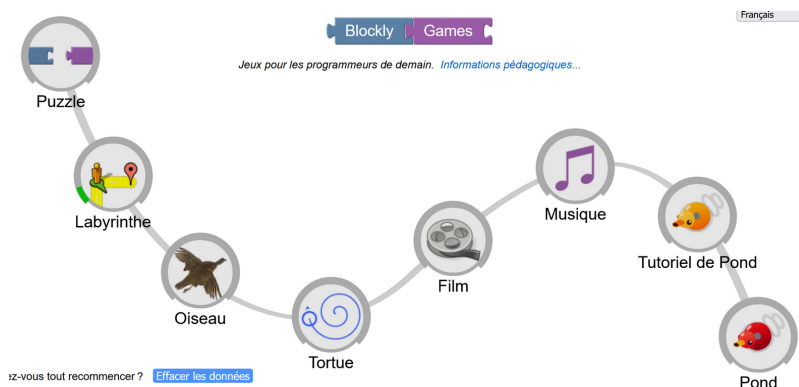
Un heure de code pour découvrir la programmation par blocs



<https://studio.code.org/hoc/1?lang=fr-FR>



Blockly Games est une série de jeux éducatifs qui enseignent la programmation



Déroulement de l'activité

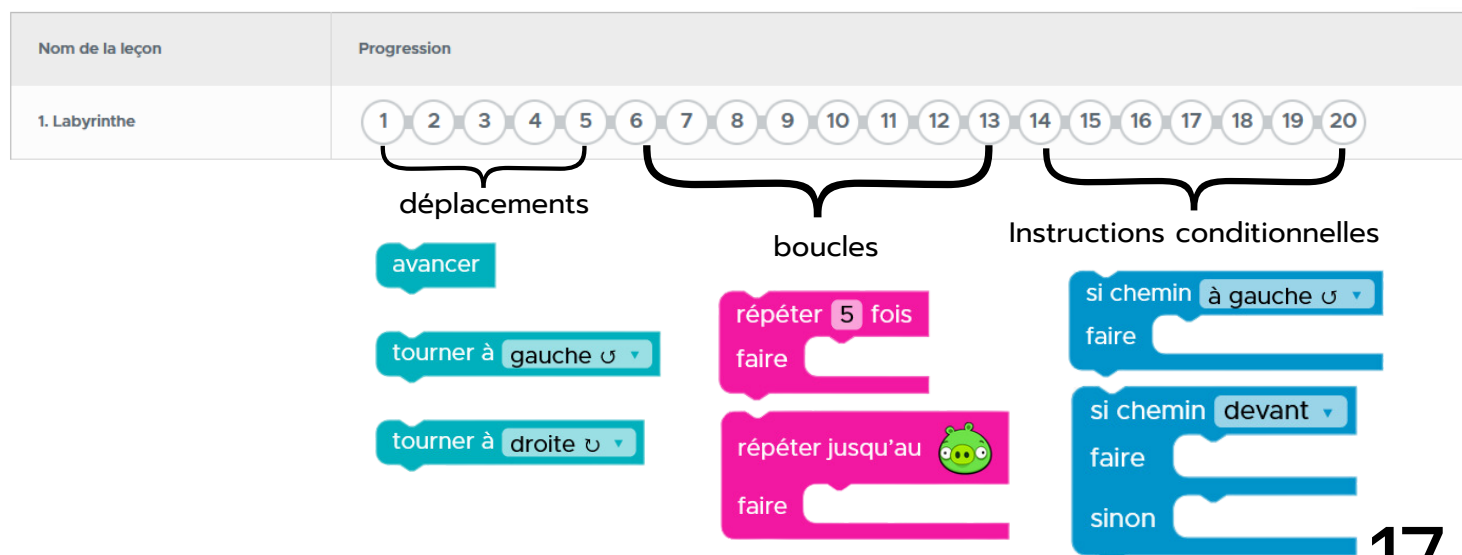
Situation mobilisatrice :

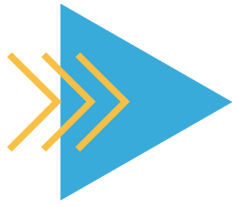
Faire observer aux élèves un programme simple utilisant des blocs (v. annexe 4).
Désigner un élève qui sera le "robot" et un autre qui lira le programme donné.
Faire verbaliser le parcours.
(On peut utiliser le carrelage en guise de quadrillage)

Activités débranchées	• Distribuer la fiche élève : " Je m'initie à la programmation par blocs". • Réaliser le premier parcours collectivement. • Travail individuel • Correction collective • Discussion autour de la différence entre un algorithme et un programme.
Activités branchées	• Maintenant que les élèves sont initiés à la programmation par blocs grâce aux activités débranchées. • Former des binômes. • Faire découvrir un environnement de programmation. • Proposer ce parcours : https://studio.code.org/hoc/1?lang=fr-FR • Observer collectivement l'interface. (v. annexe 5) • Faire une démonstration aux élèves. • Travail des élèves (aide aux élèves en difficulté)

Note à propos du parcours d'apprentissage :

Ce parcours est un didacticiel ludique où l'on y découvre les "boucles", les instructions conditionnelles et les algorithmes de base.





Pour aller plus loin ...

Le site <https://www.futureengineer.fr> offre des leçons interactives et des modèles pour apprendre la programmation et la robotique. Les cours sont structurés en chapitres et abordent différents aspects techniques et pratiques du codage. Les ressources sont spécialement conçues pour un usage éducatif, proposant des activités pratiques et des ressources pédagogiques pour faciliter l'acquisition de compétences numériques.

The screenshot displays a grid of lesson cards for Season 1 of the Future Engineer program. Each card represents an episode and includes a title, a brief description, the number of activities, the level, and a 'Lancer' button.

Episode	Titre	Description	Nombre d'activités	Niveau	Thèmes
Saison 1 - épisode 1	Séquences d'instructions	Dans ce premier épisode, découvre Citizen Code Python et manipule des instructions simples et basiques !	2	Débutant	Future Land
Saison 1 - épisode 2	Répéter une instruction	Il est maintenant l'heure de répéter des instructions. Tu vas découvrir les boucles ! Toujours plus loin, toujours plus haut, toujours plus fort.	2	Débutant	Go to Gate
Saison 1 - épisode 3	Répéter plusieurs instructions	Continue ton aventure en répétant cette fois plusieurs instructions. Ça commence à se corser !	5	Débutant	Code Park
Saison 1 - épisode 4	Plusieurs boucles	Ca ne rigole plus ! Concentre-toi bien pour associer plusieurs boucles et étendre tes activités de niveau intermédiaire !	9	Intermédiaire	FESTÀ
Saison 1 - épisode 5	Boucles imbriquées	Nouvel épisode, nouveau concept ! Tu connais les boucles imbriquées ? De monde d'un côté, c'est parti.	7	Intermédiaire	PLAGE PUBLIQUE
Saison 1 - épisode 6	Instructions conditionnelles	Ah, les conditions... Principe de base de programmation, tu vas pouvoir les manipuler dans ce dernier épisode !	2	Avancé	QUARTIER DES COINS
Saison 1 - épisode 7	Imbrications	La saison 1 touche à sa fin ! Mais il reste encore un peu de boulot. Et les imbrications, c'est pas un cadeau !	8	Avancé	Le Marché
Saison 1 - épisode 8	Open Badge	Tu as appris plein de choses ? Encore quelques activités et tu auras terminé la saison 1 et tu pourras récupérer ton openbadge !	9	Avancé	apprenti codeur Python, amazon future >> engineer

<https://app.futureengineer.fr/lessons/list/models?chapters=4104>

A la découverte de Scratch

Objectifs :

- Découvrir les bases de Scratch
- Apprendre à identifier les différentes catégories de blocs

Référentiel FMTTN**Création de contenus****Savoir-faire : Lire un programme simple**

Attendus : Lire un programme séquentiel de déplacement

Savoir-faire : Ecrire un programme simple

Attendus : Traduire un programme séquentiel de déplacement, à l'aide de blocs de codes proposés.

Matériel nécessaire

- Un ordinateur ou une tablette par groupe, avec Scratch installé ou accessible en ligne.
- Le jeu des 7 familles Scratch (imprimé et plastifié)

Déroulement de l'activité

Situation mobilisatrice :

Dans les séances précédentes, nous avons découvert la programmation par blocs dans un environnement tutoriel.

Expliquer que nous allons découvrir l'application "Scratch" où l'on pourra créer nos propres histoires.

Projeter l'interface de "Scratch", faire observer aux élèves afin d'identifier les différentes catégories de blocs.

Pour chaque catégorie, observer quelques blocs et réfléchir à l'utilité de chaque catégorie. (v. annexe 5)

Le jeu des 7 familles

Pendant le jeu :

- Pour se familiariser avec les blocs de programmation, proposer aux élèves de jouer au jeu des 7 familles.
- Lire collectivement les règles du jeu. Vérifier la compréhension de chacun.
- Diviser la classe en groupes de 4 à 5.
- Distribuer les cartes et laisser les élèves jouer.
- Pendant le jeu, aider et encourager les discussions sur les blocs.

Après le jeu :

- Demander à chaque groupe de présenter une famille de blocs qu'ils ont complétée.
- Encourager les élèves à expliquer comment chaque bloc peut être utilisé dans un programme Scratch.

Mise en pratique

- Former des binômes.
- Donner des défis simples aux élèves à réaliser sur l'application Scratch.

Exemples : faire avancer le chat de 10 pas, ... (v. annexe 7)

- Aider les élèves à commencer leur programme et les encourager à utiliser les cartes comme références.

Prolongement

- [Carnet : Je m'initie à la programmation avec Scratch](#)

Utiliser un tableur

Objectifs :

- Encoder des données dans un tableur.
- Réaliser un graphique avec un tableur.

Référentiel FMTTN**Création de contenus**

Savoir-faire : utiliser un tableur

Attendus : Encoder des données dans un tableur et réaliser un graphique

Matériel :

- un ordinateur pour deux
- un thermomètre ou un carte micro:bit
- un logiciel "tableur" (Microsoft Excel, Google Sheets, Apple Numbers, ...)

Document(s) annexe(s) :

- Utiliser un tableur.

Déroulement de l'activité

Situation mobilisatrice :

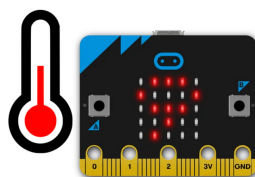
Avant de commencer l'activité, expliquez aux élèves ce que vous allez faire. Dites-leur qu'ils vont apprendre à collecter des données sur la température extérieure et à utiliser un logiciel tableur pour créer un graphique.

2 façons pour relever les températures.

Coder Micro:bit pour l'utiliser comme thermomètre.

- Proposer aux élèves de faire un thermomètre grâce à Micro:bit.

<https://microbit.org/fr/projects/make-it-code-it/thermometer/>



Utiliser un thermomètre

Utiliser un tableur pour le relevé

- Faire observer un logiciel "tableur" : quelles sont les particularités, à quoi cela peut-il servir, ...

Un logiciel tableur est un outil informatique qui permet de créer et de manipuler des tableaux de données. Il est composé de lignes et de colonnes, qui forment des cellules. Les données peuvent être saisies dans les cellules, puis modifiées, triées, filtrées, formatées, etc.

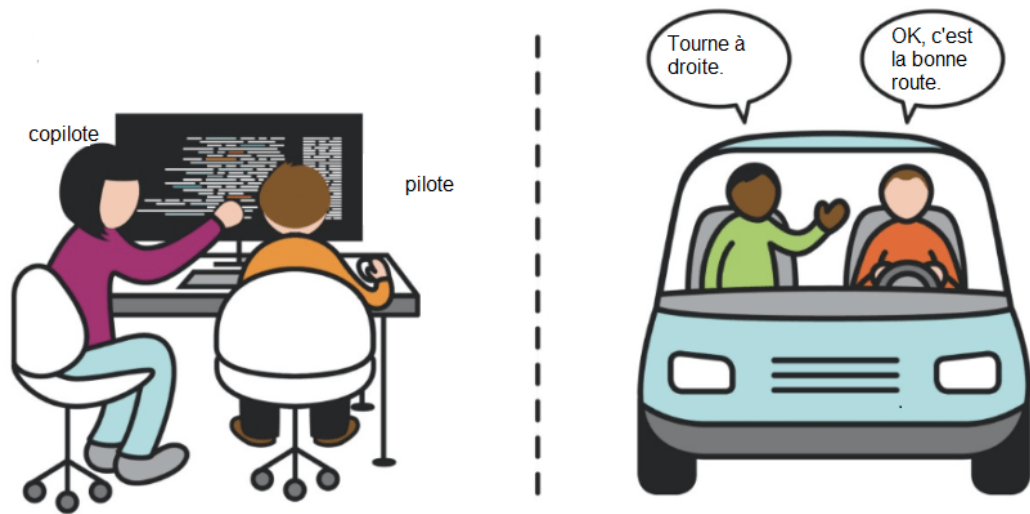
Relevé des données

- Tous les jours, pendant une semaine, demander aux élèves de noter la température extérieure tous les matins.
- À la fin de la semaine, demander aux élèves de saisir les données qu'ils ont collectées dans un logiciel tableur.
- Aider les élèves à créer un graphique en bâtons pour représenter les données

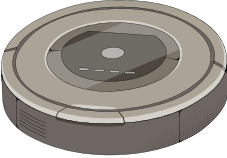
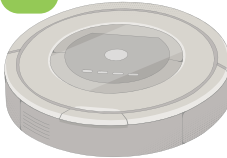
Piste de prolongement

- Faire le relevé de température durant un mois pour travailler les traitements de données : observation d'un graphique. (Exemples : trouver le jour le plus chaud, ...)
- Calculer la moyenne de la température de la semaine.
- Découvrir le calcul de moyenne dans un tableur.

Annexe 1



Annexe 2

A 				
			B 	

Répéter 3 fois

Placer 1 bloc rouge

Placer 2 blocs jaunes

Placer 1 bloc vert

Répéter 3 fois

Placer 1 bloc rouge

Placer 2 blocs jaunes

Répéter 2 fois

Placer 1 bloc vert

Placer 1 bloc jaune

Placer 1 bloc vert

Répéter 2 fois

Placer 3 blocs bleus

Placer 2 blocs jaunes

Répéter 4 fois

Placer 1 bloc noir

Placer 1 bloc rouge

Placer 1 bloc vert

Répéter 4 fois

Placer 2 blocs bleus

Placer 2 blocs verts

Annexe 4



Annexe 5

The image shows a programming interface for a game. On the left is a game grid with a red bird character and a green pig character. Below the grid is a 'Démarrer' button. On the right is a programming area with a 'Consignes' (Instructions) section at the top, a 'Blocs' (Blocks) section in the middle, and an 'Espace de travail' (Workspace) on the right. The 'Consignes' section contains a message from the red bird asking for help to catch the pig. The 'Blocs' section contains three blocks: 'avancer' (move forward), 'tourner à gauche' (turn left), and 'tourner à droite' (turn right). The 'Espace de travail' section contains one block: 'quand l'exécution commence' (when the execution begins). Annotations with arrows point to various parts of the interface:

- An arrow points to the 'Consignes' section with the text: "Lis bien la consigne."
- An arrow points to the 'Démarrer' button with the text: "Clique sur « Démarrer » pour tester ton programme."
- An arrow points to the 'Blocs' section with the text: "Les blocs de programmation"
- An arrow points to the 'Espace de travail' section with the text: "Zone de programmation : c'est la zone où tu peux déposer et assembler les blocs pour créer un programme."

Annexe 6

Les catégories de blocs

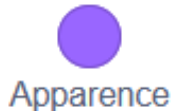
Qu'est-ce qu'un bloc ?

Comprendre les blocs de programmation sur Scratch

Les blocs de programmation sont des éléments clés pour créer des séquences d'instructions qui définissent le comportement d'un programme informatique. Chaque bloc représente une action spécifique, comme "Avancer" pour déplacer un élément ou "Dire" pour afficher du texte. En les assemblant de manière logique, tu construis des programmes pour indiquer à l'ordinateur comment exécuter des tâches spécifiques.



Les blocs « **mouvement** » concernent la position, l'orientation et le mouvement des sprites.



Les blocs « **Apparence** » permettent de créer des bulles de parole, de modifier l'apparence des sprites.



Les blocs « **son** » de faire emettre du son à l'ordinateur



Les blocs « **évènements** » permettent de déclencher les instructions placées en dessous d'eux.



Les blocs « **contrôle** » permettent d'attendre, de faire des boucles et des répétitions, de stopper un programme. Ils permettent d'organiser l'exécution des blocs



Les blocs « **capteurs** » renvoient une information de l'environnement. Ils permettent de savoir si une couleur est touchée, un bord est atteint, une touche est pressée. Les blocs s'insèrent dans d'autres blocs



Les blocs « **opérateurs** » permettent de faire du calcul. Les blocs s'insèrent dans d'autres blocs.



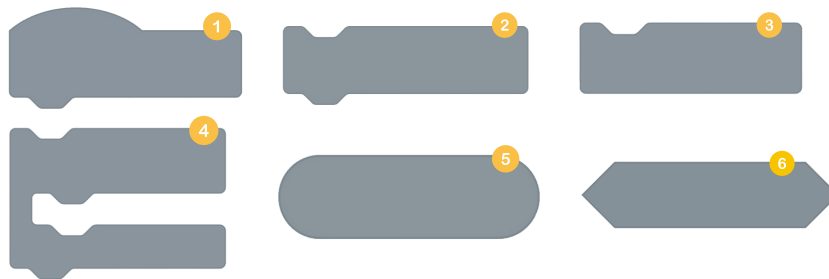
Les blocs « **variables** » permettent de créer, de nommer une nouvelle variable, de connaître sa valeur et de la modifier.



"**Mes blocs**" sont un ensemble de blocs personnalisés par l'utilisateur (extension, jumelage avec un robot)

Les formes des blocs

Les blocs dans Scratch ne sont pas tous identiques. La forme de chaque bloc détermine où il peut être placé dans le programme. Cette distinction permet d'éviter les erreurs qui pourraient altérer le fonctionnement de ton script.



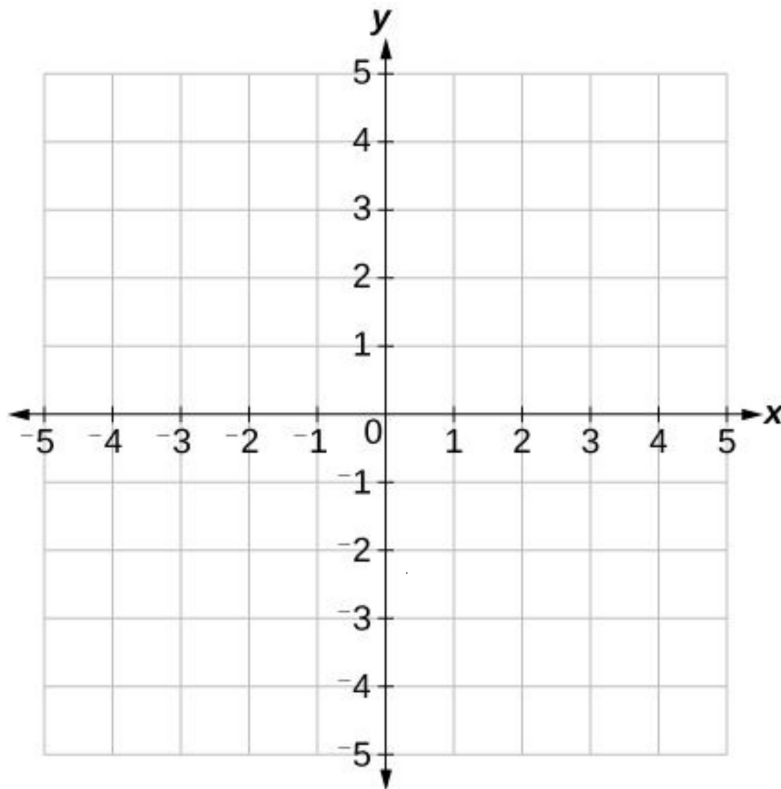
- 1 Les blocs de tête ont pour fonction de lancer le reste du script. Ils sont placés au début.
- 2 Les blocs d'empilement sont les blocs les plus communs, qui exécutent les commandes les plus basiques
- 3 Les blocs de fin sont des blocs qui se placent à la fin des scripts pour l'arrêter.
- 4 Les blocs en forme de C sont des blocs qui expriment une répétition (boucle) ou une condition (avec un bloc booléen). Ils se placent comme des blocs d'empilement, à la seule différence que l'on peut placer d'autres blocs d'empilement à l'intérieur.
- 5 Les blocs de valeur peuvent être des nombres ou du texte. Sur Scratch, il y a un peu plus de 30 blocs de valeur différents, dans plusieurs catégories, notamment les capteurs et les opérateurs. C'est également la forme des variables et des listes.
- 6 Les blocs booléens expriment si une condition est remplie ou non.

Explication « x et y »

Pour situer un objet sur un plan (sur une étendue) les mathématiciens se sont mis d'accord pour utiliser les mêmes outils. Il s'agit de deux droites graduées, "x" et "y" qui se croisent perpendiculairement.

Elles se croisent toutes les deux sur leur "0" (zéro).

Après le zéro, les chiffres sont positifs, avant le zéro, les chiffres sont négatifs.



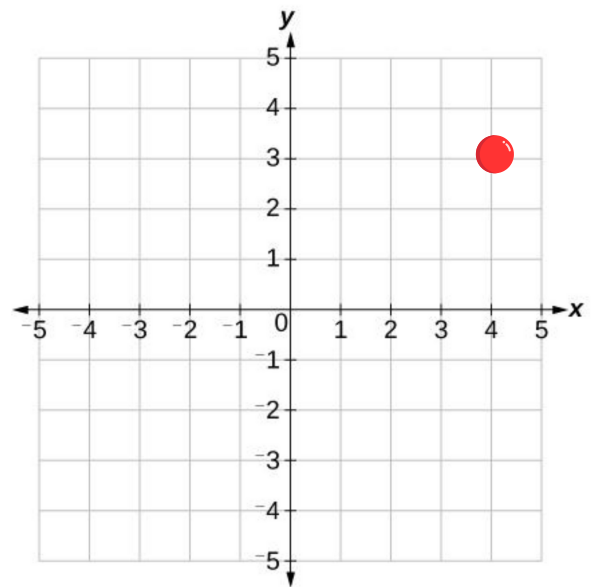
- Plus on se situe à la hauteur d'un grand nombre sur le x, plus l'objet se trouvera vers la droite. Et plus on se situe à la hauteur d'un nombre négatif sur le x, plus l'objet se trouvera sur la gauche.
- Plus on se situe à la hauteur d'un grand nombre sur le y, plus l'objet se trouvera en haut. Et plus on se situera à la hauteur d'un nombre négatif sur le y, plus l'objet se trouvera en bas.

Le y nous indique donc la hauteur de l'objet (en haut ou en bas) et le x nous indique si l'objet est à gauche ou à droite.

Exemple:

le x vaut 4: l'objet est sur la droite

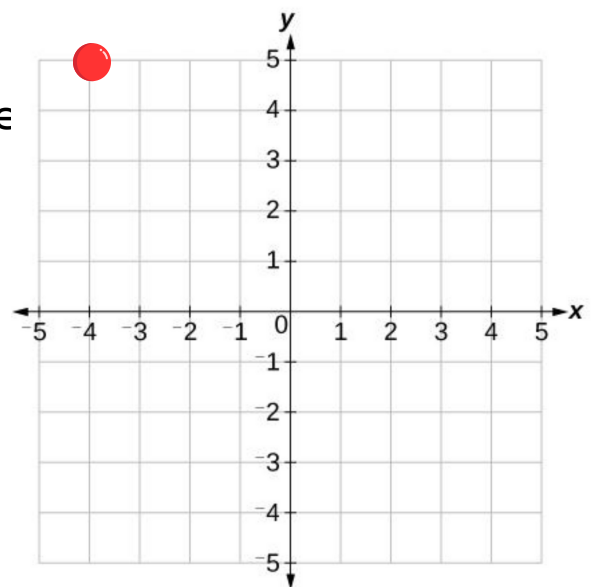
Le y vaut 3: l'objet est en haut



Exemple:

Le x vaut -4: l'objet est sur la gauche

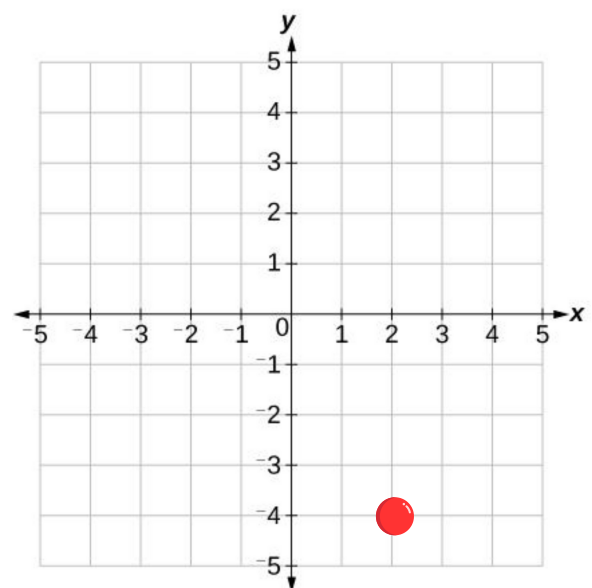
Le y vaut 5: l'objet est en haut



Exemple:

Le x vaut 2: l'objet est sur la droite

Le y vaut -4: l'objet est en bas



1

DÉFI SCRATCH

Faire avancer
le chat de 10 pas.



2

DÉFI SCRATCH

Faire avancer
le chat de 20 pas.



3

DÉFI SCRATCH

Faire avancer
le chat de 20 pas et lui
faire dire bonjour
pendant 2 secondes



4

DÉFI SCRATCH

Répéter 3 fois : faire
avancer le chat de 20
pas et lui faire dire
bonjour.



5

DÉFI SCRATCH

Répéter indéfiniment :
faire avancer le chat de
20 pas et lui faire dire
bonjour.



6

DÉFI SCRATCH

Remettre le chat au
centre de la scène.



DÉFI SCRATCH

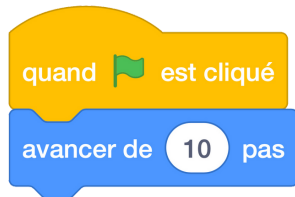
Faire faire des allers-
retours sur la scène au
chat.

Corrigé défi Scratch

1

DÉFI SCRATCH

Faire avancer
le chat de 10 pas.



2

DÉFI SCRATCH

Faire avancer
le chat de 20 pas.



3

DÉFI SCRATCH

Faire avancer
le chat de 20 pas et lui faire dire bonjour
pendant 2 secondes



4

DÉFI SCRATCH

Répéter 3 fois : faire avancer le chat de 20
pas et lui faire dire bonjour.



5

DÉFI SCRATCH

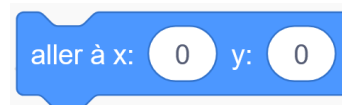
Répéter indéfiniment : faire avancer le chat
de 20 pas et lui faire dire bonjour.



6

DÉFI SCRATCH

Remettre le chat au
centre de la scène.



BONUS

DÉFI SCRATCH

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

