



Se repérer et programmer des déplacements

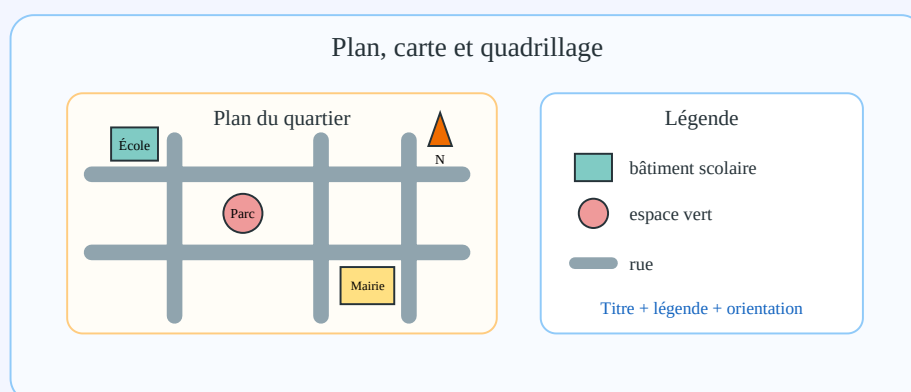
Version imprimable — SC@LPA

Se repérer sur un plan, une carte ou un quadrillage

Un **plan** ou une **carte** est une représentation simplifiée d'un lieu vu de dessus. Il permet de situer des lieux et de préparer un trajet.

Pour lire correctement un plan, on repère :

- son **titre**, qui indique le lieu représenté ;
- sa **légende**, qui explique les symboles et les couleurs ;
- son **orientation**, souvent indiquée par le nord ;
- les rues, les bâtiments et les autres **points de repère**.



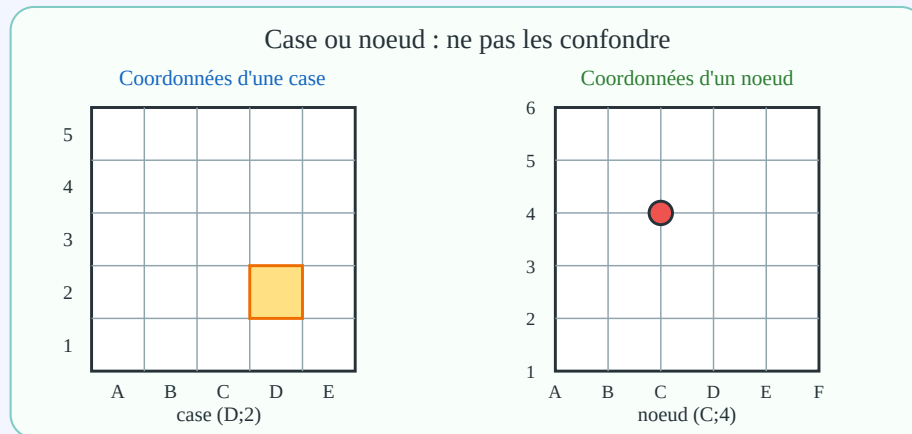
Un quadrillage facilite le repérage : ses colonnes et ses lignes permettent de localiser précisément une case ou un point.

Lire les coordonnées d'une case ou d'un noeud

Les axes d'un quadrillage sont codés avec des lettres et des nombres. On lit généralement d'abord la **colonne**, puis la **ligne**.

La case située dans la colonne D et sur la ligne 2 a pour coordonnées **(D ; 2)**.

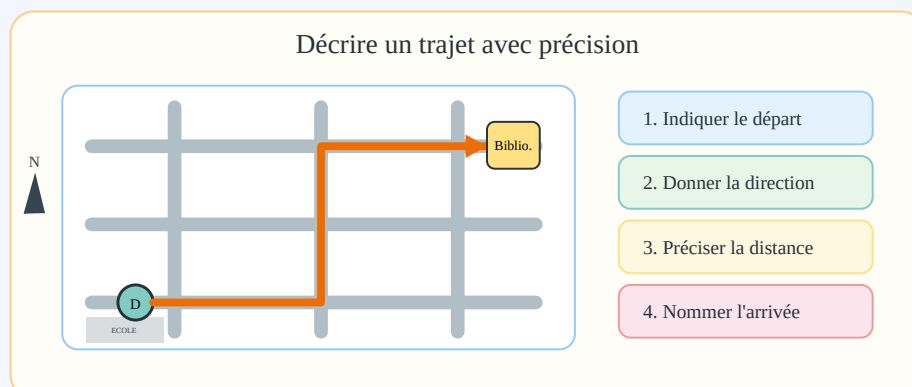
Attention : une **case** est une surface, tandis qu'un **noeud** est le point où se croisent deux lignes du quadrillage.



Avant de répondre, vérifie donc si l'on te demande les coordonnées d'une **case** ou d'un **point**.

Décrire un trajet avec précision

Pour décrire un trajet, il faut indiquer clairement le point de départ, les directions suivies, les distances parcourues et le point d'arrivée.



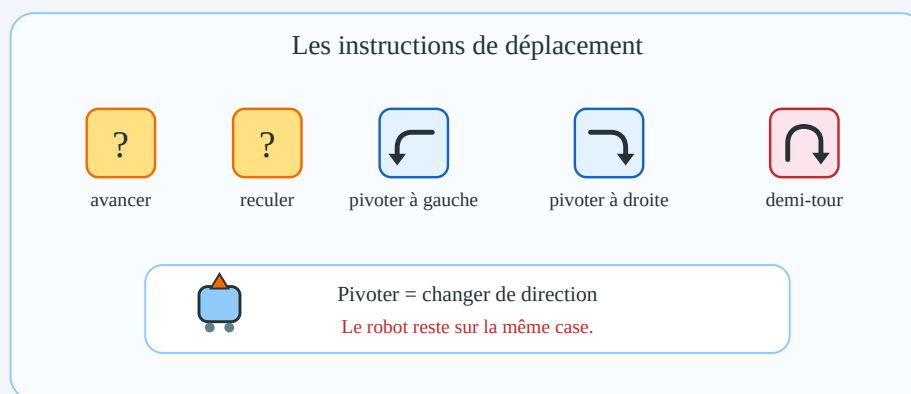
Exemple : « Sors de l'école et tourne à droite, avance de deux rues vers l'est, tourne à gauche, prends la deuxième rue à droite puis continue jusqu'à la bibliothèque. »

Sur une carte réelle, on peut aussi préciser le nom des rues, la direction d'une ligne de transport et les changements nécessaires.

Coder un déplacement avec des instructions

Un déplacement peut être codé avec des mots, des flèches ou des blocs :

- **avancer** d'une ou plusieurs cases ;
- **reculer** d'une ou plusieurs cases ;
- **pivoter à gauche** ou **pivoter à droite** ;
- faire un **demi-tour**.



Quand un robot pivote, il change de direction mais **reste sur la même case**.

Pour exécuter un programme, place-toi mentalement à la place du robot et regarde toujours dans la même direction que lui.

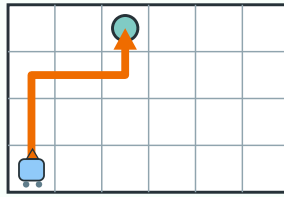
Exécuter une suite d'instructions dans l'ordre

Un programme est une **suite ordonnée d'instructions**. Le robot exécute la première instruction, puis la deuxième, et ainsi de suite.

Exécuter le programme pas à pas



Lire de gauche à droite



Après chaque instruction :

1. repérer la case
2. vérifier l'orientation
3. poursuivre le programme

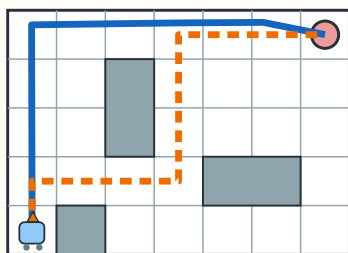
Changer l'ordre de deux instructions peut conduire à une arrivée différente ou provoquer une collision avec un obstacle.

Pour vérifier un programme, exécute-le pas à pas en marquant la position et l'orientation du robot après chaque instruction.

Éviter les obstacles et chercher un trajet efficace

Plusieurs trajets peuvent conduire au même point. Un trajet est souvent considéré comme plus efficace s'il utilise moins de déplacements ou moins de rotations.

Comparer des trajets et éviter les obstacles



trajet A : 10 avancées, 1 pivot

trajet B : 10 avancées, 3 pivots

Comparer avant de choisir.

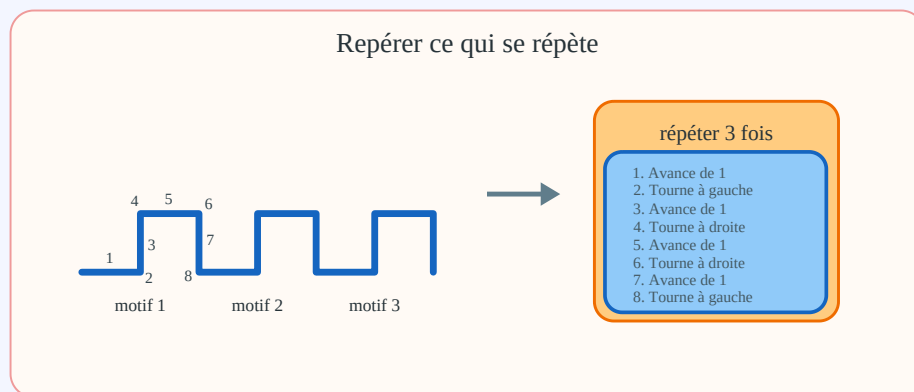
Pour choisir un trajet :

1. repère le départ, l'arrivée et les obstacles ;
2. imagine plusieurs chemins possibles ;
3. compte les déplacements et les rotations ;
4. vérifie que chaque case traversée est autorisée.

Les deux trajets atteignent la cible sans toucher les obstacles. On compare ensuite le nombre de déplacements et de pivots pour choisir le trajet le plus efficace.

Utiliser une boucle pour répéter

Une **boucle** permet de répéter plusieurs fois la même instruction ou le même groupe d'instructions.

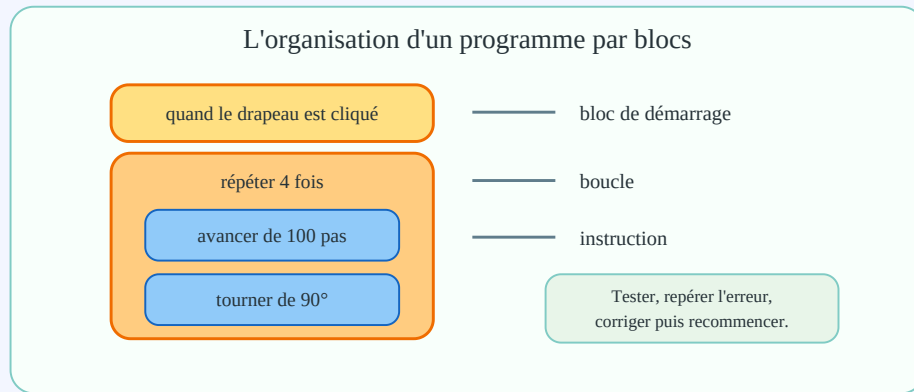


Au lieu d'écrire vingt-quatre instructions « avancer, pivoter... », on peut écrire : **répéter 3 fois** ces huit instructions. → On a donc 9 instructions au lieu de 24.

Avant de créer une boucle, repère exactement la partie du programme qui se répète et compte le nombre de répétitions.

Lire, produire et corriger un programme par blocs

Dans un logiciel comme Scratch, les blocs s'assemblent pour former un programme. Chaque bloc correspond à une action, un test ou une répétition.



Pour corriger un programme qui ne produit pas le résultat attendu :

1. vérifie le bloc de démarrage ;
2. exécute les instructions une à une ;
3. repère l'instruction absente, inutile ou mal placée ;
4. corrige puis teste à nouveau.

Cette recherche d'erreur s'appelle le **débogage**.

Tester un algorithme de déplacement avec Scratch

Observe le quadrillage, le départ, l'arrivée et les obstacles. Trois algorithmes sont proposés : choisis celui qui te semble correct, puis teste ta réponse.

1. clique sur la proposition A, B ou C ;
2. vérifie que ton choix est bien enregistré ;
3. clique sur **Tester mon choix** ;
4. observe le trajet du robot et lis le résultat affiché.

Cette activité Scratch est interactive et doit être réalisée dans la version en ligne de la leçon.