

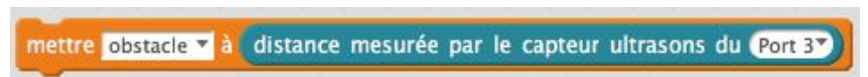
La programmation des **objets connectés** nécessite la gestion de **situations complexes** : déplacements, trajectoires, mesures des capteurs... Pour résoudre ces problèmes plus **évolués**, les programmeurs utilisent deux types de **variables informatiques** dans leurs **algorithmes** : les **variables dites « statiques »** et **« dynamiques »**.

- **Les variables statiques** sont des **valeurs constantes**. Elles sont **stockées** dans la mémoire de l'objet connecté.



Variable statique permettant de régler la vitesse d'un robot.

- **Les variables affectées à des capteurs** sont **dynamiques** car leurs **valeurs changent dans le temps** en fonction de la **variation de la mesure du capteur**. Ces variables sont **stockées** dans la **mémoire**.

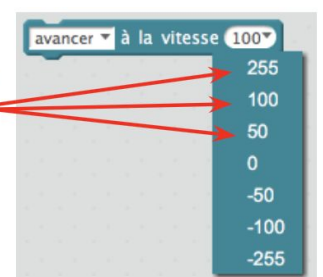


Variable dynamique permettant de stocker la valeur mesurée par le capteur à ultrasons.

- **Comment utiliser des variables statiques pour régler la vitesse d'un robot ?**

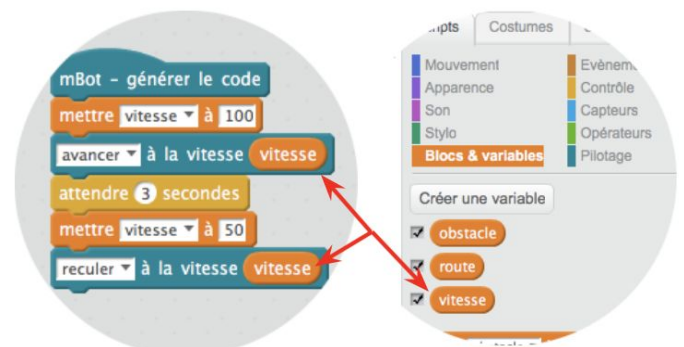
Cas 1 - La plupart du temps, les **variables** sont **prédéfinies** dans le logiciel de programmation.

Le programmeur a le **choix entre plusieurs valeurs** pour **régler la vitesse** de son robot.



Cas 2 - Les **variables** peuvent aussi **être créées** par le programmeur.

Elles **porteront un nom précis** en fonction du langage de programmation (exemple : var, B0, B1, vitesse...)



- Comment utiliser des variables dynamiques pour gérer l'arrêt d'un robot devant un obstacle ?

Un capteur à ultrasons va mesurer le changement de distance entre le capteur et l'obstacle.

Dans le programme, on stocke la valeur mesurée par le capteur à ultrasons dans la variable « obstacle » qui va varier quand le robot se déplacera.

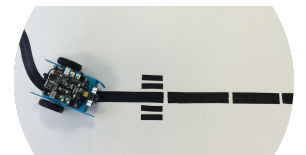
```
mettre obstacle à distance mesurée par le capteur ultrasons du Port 3
```

```
répéter jusqu'à obstacle < 10
  avancer à la vitesse vitesse
  avancer à la vitesse 0
```

Quand il sera à moins de 10 cm de l'obstacle, il s'arrêtera.

- Comment utiliser des variables dynamiques pour réaliser le comptage des intersections d'une route ?

1 - Le robot suit la route et passe devant les intersections (ruptures de la route).



2 - Une variable nommée « intersection » sera incrémentée quand le capteur de ligne du robot détectera la rupture de la route.

```
mettre intersection à 0
répéter indéfiniment
  si suiveur de ligne Port 2 Côté gauche est blanc et suiveur de ligne Port 2 Côté droit est blanc alors
    ajouter à intersection 1
    avancer à la vitesse 100
    attendre 0.3 secondes
```

3 - On compare la variable « intersection » à un nombre précis de carrefour (ici 3) pour donner l'ordre de tourner.

```
si intersection = 3 alors
  tourner à gauche à la vitesse 100
  attendre 0.7 secondes
  reculer à la vitesse 100
  attendre 1 secondes
  avancer à la vitesse 0
  attendre 5 secondes
```

Les variables sont des éléments qui associent un nom (qui sera implantée dans la mémoire du système programmé) à une valeur qui peut varier au cours de l'exécution du programme. Elles peuvent être statiques (temps, vitesse ...) ou dynamiques (détection de distances, de couleurs ...) et peuvent servir au comptage.

Structurer les connaissances

Déclenchement d'une action par un évènement, instructions conditionnelles

CYCLE 4

>

» Début de cycle
» Milieu de cycle
» Fin de cycle

Pour résoudre différents problèmes et rendre les systèmes toujours plus autonomes, on utilise un algorithme avec des instructions qui peuvent être exécutées si des évènements sont détectés par des capteurs.

- Dans un algorithme ou un programme, les conditions qui déclenchent des actions à partir d'évènements sont des instructions conditionnelles. Elles permettent d'exécuter les instructions si les conditions sont vraies ou pas.

Différents évènements peuvent être détectés comme :

- la variation d'une grandeur physique (changement de luminosité, de chaleur, de couleur...)
- le déplacement d'un objet mesuré par un capteur du système
- la présence ou non d'une route
- ou simplement un évènement initial, permettant de lancer le programme.



- Les instructions peuvent être conditionnées par l'apparition d'évènements initiaux ou détectés par des capteurs.

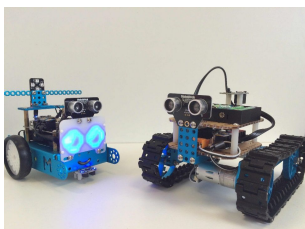
- Déclenchement d'une action par un évènement dit initial : "Quand évènement"

Exemple : « quand touche espace est pressée »



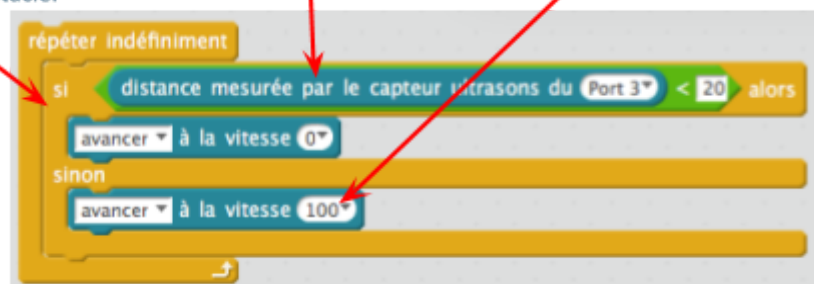
- Déclenchement d'une action par un évènement conditionnel : Si... Alors...Sinon

Exemple : le robot mBot doit détecter un obstacle.



*Si obstacle Alors
arrêt
Sinon avancer
Si obstacle Alors arrêt...*

Le robot mBot doit laisser la priorité, en détectant l'apparition d'un autre véhicule. Il s'arrête quand il détecte un obstacle devant lui et se remet en route quand il n'y a plus d'obstacle.



Dans un algorithme ou un programme, l'exécution des instructions peut être conditionnée par l'apparition d'un évènement. Celles-ci sont des instructions conditionnelles. Dans ce cas, l'instruction s'exécute SI l'évènement a lieu, SINON une instruction différente pourra aussi se réaliser.

Pour résoudre différents problèmes et rendre les systèmes toujours plus **autonomes**, le programmeur va devoir réaliser **des algorithmes** avec des **séquences d'instructions** et des **boucles**.

Pour écrire un programme, on va utiliser des **instructions qui peuvent être exécutées une seule fois ou répétées plusieurs fois**.

• Comment réaliser des séquences d'instructions ?

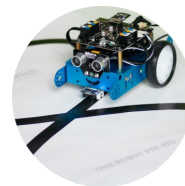
Les actions d'un système peuvent être déclenchées 1 fois en **séquences d'instructions** sans condition préalable : **avancer, tourner à gauche, à droite, reculer...**

Dans l'exemple, on pilote le robot pour qu'il avance pendant 5 secondes puis tourne 0,8 seconde et s'arrête.

Le robot aura avancé d'environ 50 centimètres et pivoté à droite de 90°

```

avancer à la vitesse 100
attendre 5 secondes
tourner à droite à la vitesse 100
attendre 0.8 secondes
avancer à la vitesse 0
    
```



• Comment réaliser des boucles ?

Les instructions peuvent être **répétées en boucles un certain nombre de fois** avant de passer à une autre action ou **répétées indéfiniment**.

Le robot en répétant 4 fois cette séquence, va réaliser un carré de 10 cm de côté environ.

```

répéter 4 fois
  avancer à la vitesse 100
  attendre 1 secondes
  tourner à droite à la vitesse 100
  attendre 0.8 secondes
  avancer à la vitesse 0
    
```



```

répéter indéfiniment
  avancer à la vitesse 100
  attendre 1 secondes
  tourner à droite à la vitesse 100
  attendre 0.8 secondes
    
```

Dans ce cas, l'exécution du programme ne s'arrêtera que lorsque l'opérateur éteindra le système.

Les **instructions** d'un **algorithme** peuvent être déclenchées :

- en **séquences** (séquences d'instructions) : les **ordres sont enchaînés les uns à la suite des autres** sans condition préalable (avancer, tourner...)
- et/ou **répétées en boucle un nombre précis de fois ou indéfiniment**.