

# Listes

Comme dans le langage courant, ROBO Pro Coding contient une liste d'éléments ordonnés, tels qu'une liste « à faire » ou une liste d'achats. Les éléments d'une liste peuvent être de n'importe quel type et la même valeur peut apparaître plusieurs fois dans une liste.

## Création d'une liste

### Créer une liste avec

Le bloc **Créer une liste avec** permet de saisir les valeurs initiales dans une nouvelle liste. Dans cet exemple, une liste de mots est créée et stockée dans une variable appelée **lettres** :

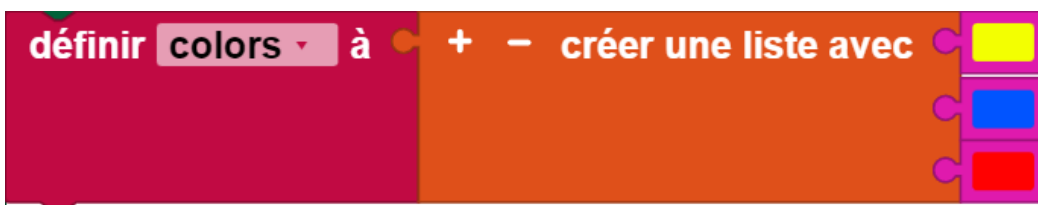


Nous appelons cette liste [« alpha », « bêta », « gamma »].

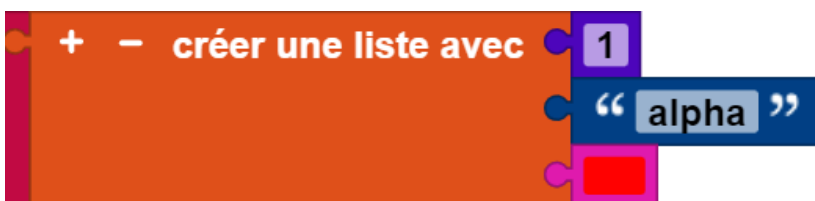
Ceci montre la création d'une liste de **chiffres** :



Pour créer une liste de **couleurs** :



Il est moins courant, mais il est possible de dresser une liste de valeurs de différents types :



### Modifier le nombre d'entrées

Pour modifier le nombre d'entrées, cliquez ou appuyez sur l'icône de la roue dentée. Ceci ouvre une nouvelle fenêtre. Vous pouvez faire glisser des sous-blocs d'éléments du côté gauche de la fenêtre vers le bloc de liste du côté droit pour ajouter une nouvelle entrée. :

Si le nouvel élément a été ajouté dans cet exemple, il peut être ajouté n'importe où. De même, les sous-blocs d'éléments indésirables peuvent être tirés vers la gauche à partir du bloc de liste.

## Créer une liste d'éléments

Le bloc **Créer une liste avec un élément** permet de créer une liste contenant le nombre indiqué de copies d'un élément. Par exemple, les blocs suivants ajoutent des **mots** à la liste [« très », « très », « très »].



## Contrôle de la longueur d'une liste

### Vide

La valeur d'un bloc **vide** est **vraie** si sa saisie est la liste vide et **fausse** si c'est autre chose. Cette entrée est-elle **vraie** ? La valeur du bloc suivant serait **fausse** car la variable Couleurs n'est pas vide : elle comporte trois éléments.



Notez la similitude avec le bloc **vide** pour le texte.

### Longueur de

La valeur du bloc **Longueur de** est le nombre d'éléments de la liste utilisée comme entrée. La valeur du bloc suivant serait par exemple 3, car **la couleur** comporte trois éléments :



Notez que le bloc **Longueur de** indique le nombre d'éléments contenus dans la liste et non le nombre d'éléments différents contenus dans la liste. Par exemple, ce qui suit la valeur 3, bien que les **mots** soient composés de trois copies du même texte :



Notez la similitude avec le bloc **Longueur de** pour le texte.

## Recherche d'éléments dans une liste

Ces blocs trouvent la position d'un élément dans une liste. L'exemple suivant a la valeur 1 parce que la première occurrence de "très" figure en tête de la liste de mots ([« très », « très », « très »]).

dans la liste words trouver la première occurrence de l'élément "very"

Le résultat de ce qui suit est 3, parce que la dernière occurrence de « très » en **mots** est en position 3.

dans la liste words trouver la dernière occurrence de l'élément "very"

Si l'élément n'apparaît nulle part dans la liste, le résultat est 0, comme dans cet exemple :

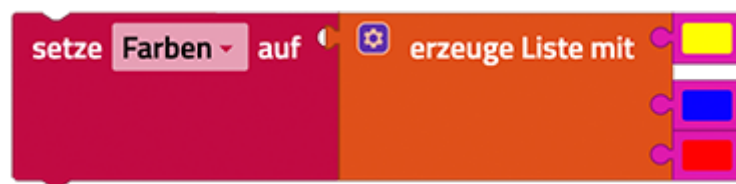
dans la liste words trouver la dernière occurrence de l'élément "unicorn"

Ces blocs se comportent de la même manière que les blocs permettant de trouver des lettres dans le texte.

## Récupération d'éléments d'une liste

### Récupération d'un élément unique

Rappelez-vous de la définition de la liste **Couleurs** :



Le bloc suivant reçoit la couleur bleu parce qu'il est le deuxième élément de la liste (compté à partir de la gauche) :

dans la liste colors obtenir n° 2

Celui-ci reçoit du vert parce qu'il est le deuxième élément (compté depuis l'extrémité droite) :

dans la liste colors obtenir n° depuis la fin 2

Celui-ci reçoit le premier élément, rouge :

dans la liste colors obtenir premier

Celui-ci reçoit le dernier élément, jaune :

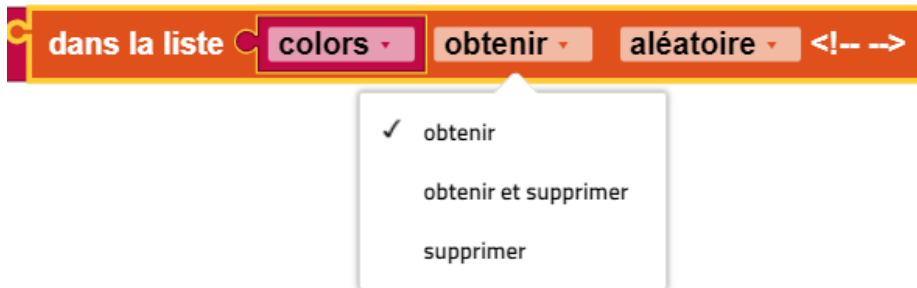
dans la liste colors obtenir dernier

Ceci sélectionne au hasard un élément de la liste en renvoyant avec la même probabilité l'un des éléments rouge, bleu, vert ou jaune.



## Récupération et suppression d'un élément

Le menu déroulant permet de modifier le bloc **Récupérer dans la liste ...** dans le bloc **Récupérer dans la liste ... et supprimer** qui fournit le même résultat, mais modifie également la liste :



Cet exemple place la variable **Première lettre** sur « alpha » et laisse les lettres restantes ([« beta », « gamma »]) dans la liste.



## Supprimer une entrée

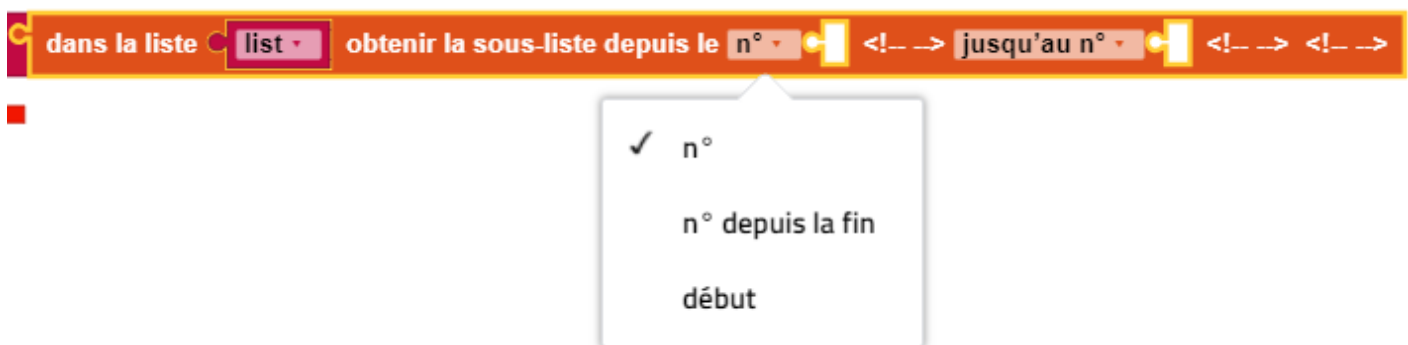
Si vous sélectionnez **Supprimer** dans le menu déroulant, le nez disparaît du bloc à gauche :



Cela supprime le premier élément des **lettres**.

## Afficher une sous-liste

Le bloc **Dans la liste ... Afficher la sous-liste** ressemble au bloc **Récupérer dans la liste ...** à la différence qu'il extrait une sous-liste et pas un élément individuel. Il existe plusieurs options pour indiquer le début et la fin de la sous-liste :



Dans cet exemple, une nouvelle liste de **première lettre** est établie. Cette nouvelle liste comporte deux éléments : [« alpha », « bêta »].



Remarquez que ce bloc ne change pas la liste d'origine.

## Ajout d'éléments à une liste

### Remplacer les éléments dans une liste

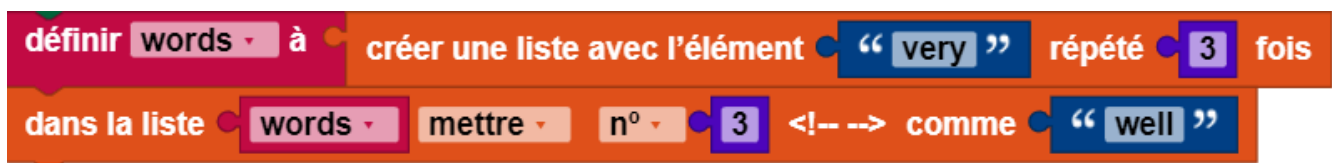
Le bloc **Remplacer dans la liste ...** remplace l'élément à un emplacement défini d'une liste par un autre élément.



Vous trouverez dans la section précédente la signification des différentes options de menu déroulant.

L'exemple suivant fait deux choses :

1. La liste **mots** est composée de 3 éléments : [« très », « très », « très »].
2. Le troisième élément de la liste est remplacé par « bien ». La nouvelle valeur des **mots** est [« très », « très », « bien »]



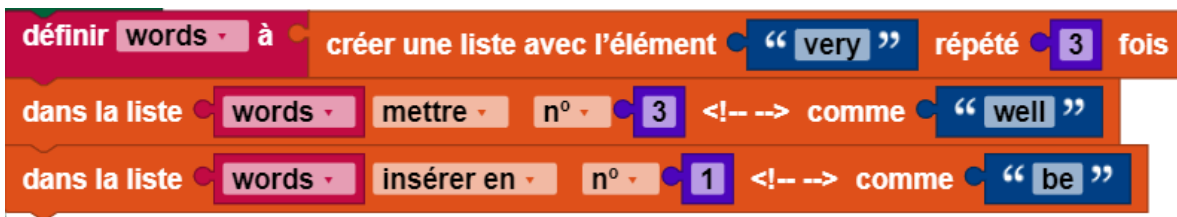
### Insérer des éléments à un emplacement donné dans une liste

Le bloc **Ajouter dans la liste ... pour** est affiché via le menu déroulant du bloc **Remplacer dans la liste ...** :



Il insère un nouvel élément dans la liste à l'emplacement indiqué, devant l'élément précédemment présent à cet emplacement. L'exemple suivant (tiré d'un précédent exemple) fait trois choses :

1. La liste **mots** est composée de 3 éléments : [« très », « très », « très »].
2. Le troisième élément de la liste est remplacé par « bien ». La nouvelle valeur des **mots** est donc [« très », « très », « bien »].
3. Le mot « être » est ajouté au début de la liste. La valeur finale des **mots** est donc [« être », « très », « très », « bien »].



## Diviser les chaînes et fusionner les listes

### Créer une liste à partir d'un texte

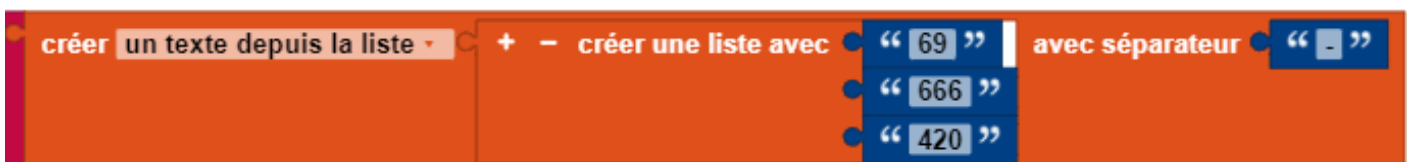
Le module **Créer une liste à partir de texte** décompose le texte spécifié en parties à l'aide d'un caractère de délimitation :



Dans l'exemple ci-dessus, une nouvelle liste contenant trois morceaux de texte est renvoyée : « 311 », « 555 » et « 2368 ».

### Créer un texte à partir d'une liste

Le module **Créer un texte à partir d'une liste** fusionne une liste en un seul texte à l'aide d'un séparateur :



Dans l'exemple ci-dessus, un nouveau texte est renvoyé avec la valeur : « 311-555-2368 ».

## Blocs apparentés

### Impression d'une liste

Le module **Imprimer** de la catégorie Texte peut émettre des listes. Le résultat du programme suivant est l'édition de console représentée :



Console

Démarrage du programme ...

['alpha', 'beta', 'gamma']

Programme terminé.

## Exécuter quelque chose pour chaque élément dans une liste

Le bloc **pour chacun** dans la catégorie Commande exécute une opération pour chaque élément dans une liste. Par exemple, ce bloc imprime individuellement chaque élément de la liste :



Ceci ne supprime pas les éléments de la liste d'origine.

Voir aussi les exemples pour les [blocs de rupture de boucle](#).

Revision #8

Created 17 November 2021 20:44:54 by Admin

Updated 21 February 2022 12:22:14 by Admin