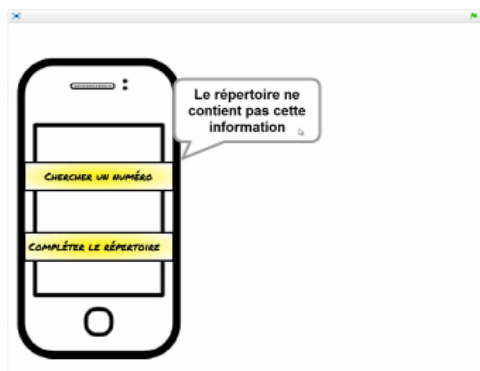


Les listes: création d'un répertoire

→ Ouvrez le fichier "[Telephone.sb2](#)"

Nous allons programmer un répertoire téléphonique, comme illustré par [la vidéo](#) ci-dessous.

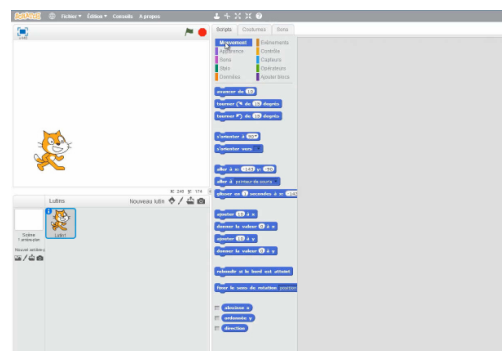


Pour cela, nous allons avoir besoin d'un nouveau type de variable: **la liste**.

« Mais comment utilise-t-on une liste ? »



Dans le menu «Données», il est possible de créer une liste: c'est ici une colonne dans laquelle on peut ranger autant d'informations que souhaitées (enfin...tant qu'il reste de la mémoire à l'ordinateur). Chaque information possède un rang indiqué par sa position dans la liste. Ci-contre, [une vidéo](#) explicative.



Ce programme comporte trois lutins: un téléphone, un bouton "Nouveau" pour compléter le répertoire, et un bouton "Chercher" qui permet d'interroger ce répertoire. Chaque lutin possède un script de placement en début de programme.

Exercice 1: Compléter le répertoire

Avant de commencer le script, créez deux listes: "Liste de noms" qui contiendra les noms des personnes et "Liste de numéros" qui contiendra leur numéro de téléphone.

Nous allons ici programmer le lutin "Nouveau" selon l'algorithme suivant:

Quand on clique sur ce lutin:

- il demande d'entrer un nouveau nom,
- il ajoute la réponse à la liste "Liste de noms",
- il demande d'entrer un numéro de téléphone,
- il ajoute la réponse à la liste "Liste de numéros".

→ Programmez le lutin "Nouveau" en utilisant les blocs suivants :



→ Lancez le programme et enregistrez les numéros de téléphone de 5 personnes.

→ Cachez les listes pour qu'elles n'apparaissent plus dans la scène.

« Je me suis trompé ! Comment faire pour supprimer un élément ? »



Il existe un bloc qui permet de vider une liste, ou de supprimer un élément en connaissant son rang. Si vous voulez supprimer un ou tous les éléments, utilisez le script suivant :



Exercice 2: Vérifier qu'une liste contient un élément demandé.

→ Créez une variable “Nom à chercher”.

Nous allons programmer le lutin “Chercher” pour que lorsqu’il est cliqué :

- il demande quel nom doit être cherché,
- il stocke la réponse dans la variable “Nom à chercher”,
- il exécute le test suivant: **Si** le “Nom à chercher” est dans la “Liste de noms”, **alors** il affiche le message “Ce nom est dans le répertoire”, **sinon** il affiche “Ce nom n’est pas dans le répertoire”.

→ Utilisez les blocs ci-dessous pour programmer cet algorithme.



Exercice 3: Trouver le rang d'un élément.

→ Créez une variable “Rang”.

Nous allons utiliser le script suivant:

“Rang” est la position de l’élé-

ment observé dans la “Liste de noms”

On commence par l’élément en position n°1.

Tant que l’élément observé n’est pas celui

cherché, on ajoute 1 à “Rang” pour observer l’élément suivant. Cette boucle doit forcément s’arrêter, car elle ne doit se lancer que si l’on sait auparavant que le “Nom à chercher” est bien dans la “Liste de noms”.



→ Recopiez ce script au bon endroit du programme pour obtenir le résultat attendu.

Exercice 4: Trouver et afficher le numéro correspondant.

→ Créez une variable “Numéro trouvé”.

Le “Rang” trouvé à l’exercice 3 est la position du “Nom à chercher” dans la “Liste de noms”.

Mais c’est aussi celle du numéro de cette personne dans la “Liste des numéros”, car ces deux éléments ont été créés au même moment et ont été ajoutés aux listes en même temps.

Il suffit donc de stocker dans la variable “Numéro trouvé” l’élément de la “Liste de numéros” situé à la position indiqué par “Rang”. Il faudra enfin construire une phrase de conclusion donnant le nom de la personne associée à ce numéro.

→ A l’aide des blocs ci-dessous, modifiez le programme pour qu’il corresponde à cet algorithme.

