

PRÉAMBULE

Généralités

Tous nos tutoriels permettant de créer des exercices utilisent la **version 5** de GeoGebra.

Dans chacun de ces tutoriels, vous trouverez :

- dans la marge de droite de la feuille, l'objectif qui est expliqué dans le paragraphe en regard, ce qui vous permettra de retrouver facilement les manipulations à effectuer en cas de besoin ;
- un renvoi vers un autre tutoriel vous permettant d'ajouter un score à l'activité ;
- un renvoi vers un autre tutoriel vous permettant de fixer le nombre total de questions ;
- un renvoi vers un autre tutoriel expliquant comment insérer l'activité dans une leçon GeoGebra.

Objectif spécifique de ce tutoriel

L'objectif de ce tutoriel est d'apprendre à créer un exercice dans lequel les élèves choisissent leur réponse parmi les propositions d'une liste déroulante. Pour illustrer cet objectif, nous avons choisi de proposer à l'utilisateur de se prononcer sur la divisibilité par 2, 3, 5, 9 ou 10 d'un nombre entier.

Cadrage

Si on propose à l'utilisateur un nombre entier choisi au hasard entre 2 et 1 000 et qu'on teste sa divisibilité par l'un de ces 5 nombres, il est fort probable d'aboutir sur une non-divisibilité, ce qui ne donne pas beaucoup d'intérêt à l'exercice. Nous faisons donc le choix de prendre un nombre qui est soit un multiple de 3, soit un multiple de 5. On obtient ainsi les probabilités de divisibilité suivantes,

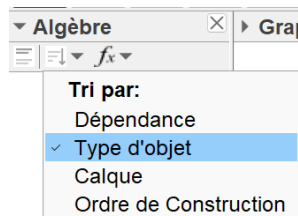
par 2 : $\frac{665}{1\,332} \approx 49,9\%$	par 5 : $\frac{133}{222} \approx 59,9\%$	par 10 : $\frac{133}{444} \approx 30\%$
par 3 : $\frac{2}{3} \approx 66,7\%$	par 9 : $\frac{133}{600} \approx 22,1\%$	

La probabilité que l'entier soit divisible par le nombre choisi est alors environ égale à 0,457, c'est à dire 45,7%.

1 - PRÉPARATION DE L'INTERFACE

Commencer par afficher la fenêtre « Algèbre », soit en cliquant sur « Affichage » puis sur « Algèbre », soit avec la combinaison de touches **Ctrl** + **Maj** + **A**.

Afin de retrouver plus facilement les objets qui seront construits, dérouler le menu « Algèbre », puis cliquer sur  et ensuite sur « Type d'objet »,



Dans la fenêtre « Graphique », si les axes et/ou la grille sont visibles, les cacher en effectuant un clic-droit dans la fenêtre graphique et en cliquant sur « Axes » ou sur « Grille ».

2 - CRÉATION DU NOMBRE ALÉATOIRE À ÉTUDIER

Nous allons créer deux listes de nombres : l'une contenant les multiples de 3 non nuls inférieurs à 1 000 et l'autre contenant les multiples de 5 non nuls inférieurs ou égaux à 1 000.

Pour chaque question, nous déterminerons aléatoirement l'une des deux listes, puis un entier appartenant à cette liste.

Construisons ces listes avec la commande « Séquence »¹.

Saisie : `m3=Séquence(3k,k,1,333)`

Saisie : `m5=Séquence(5k,k,1,200)`

Pour déterminer la liste à utiliser, on va tirer un élément au hasard dans la liste {3,5}.

Saisie : `c=ElémentAuHasard({3,5})`

La liste `m` retenue sera donc définie en fonction de la valeur de `c`.²

Saisie : `m=Si(c==3,m3,m5)`

Le nombre à étudier est défini par la commande suivante.

Saisie : `a=ElémentAuHasard(m)`

Générer une liste

Tirer aléatoirement un élément dans une liste

1. La commande **Séquence**(*<Expression e>*,*<Variable k>*,*<de a>*,*<à b>*) permet de créer une liste d'objets en utilisant l'expression *e* qui contient l'indice *k* variant du nombre *a* au nombre *b*.

2. La commande **Si**(*<Condition>*,*<Alors>*,*<Sinon (optionnel)>*) retourne une copie de l'objet « Alors » si la condition est réalisée. Si la condition n'est pas réalisée et que le « Sinon » est précisé, la commande retourne une copie de l'objet « Sinon ». Sinon « non défini » est retourné.

Pour effectuer une test d'égalité, on utilise l'enchaînement « == » qui correspond à « $\stackrel{?}{=}$ ».

3 - CRÉATION DU CHOIX DU DIVISEUR À ÉTUDIER

Créer une liste nommée « tables » des diviseurs possibles.

Saisie : `tables={2,3,5,9,10}`

Choisir maintenant au hasard un élément de cette liste.

Saisie : `b=ElémentAuHasard(tables)`

4 - CRÉATION DU TEXTE À COMPLÉTER

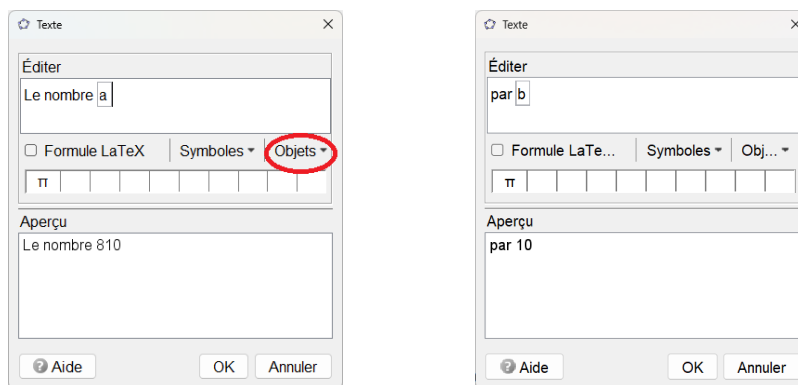
La phrase va être découpée en trois parties, à savoir :

« Le nombre a » « est/n'est pas divisible » « par b. ».

a. Deux textes dynamiques :

La première et la troisième partie sont des textes dynamiques faisant appel aux variables a et b.

Créer deux textes comme dans les aperçus suivants (dans les captures ci-dessous, la variable a vaut 810 et la variable b vaut 10).



Créer un
texte
dynamique

Les variables a et b sont appelées en cliquant sur le bouton « Objet ».

Une fois ces textes créés, dans leurs propriétés, onglet « Texte », régler leurs tailles sur « Grande » et les disposer sur une même ligne (on pourra, pour cela, aimer la grille).

b. Un objet liste :

La partie centrale de la phrase sera sous la forme d'une liste déroulante dans laquelle il y aura trois propositions :

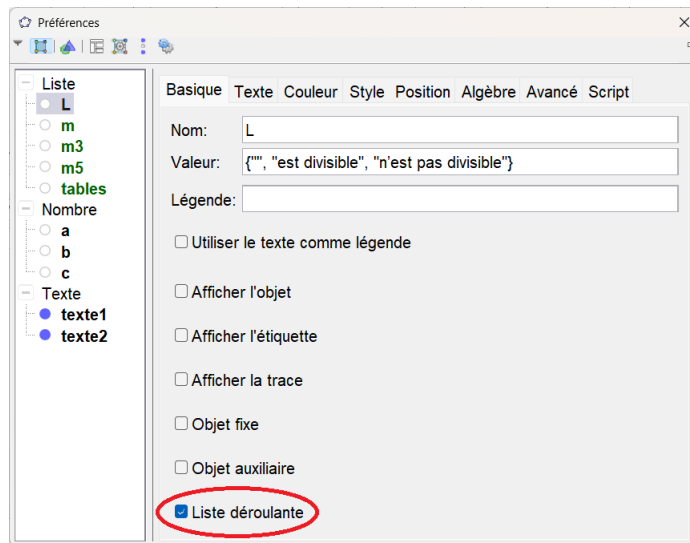
- absence de texte (solution affichée par défaut),
- « est divisible »,
- « n'est pas divisible ».

Créer une
liste
déroulante

Pour créer cette liste déroulante, nous commençons par créer une simple liste :

Saisie : `L={"", "est divisible", "n'est pas divisible"}`

Dans les propriétés de cette liste, onglet « Basique », on va maintenant cocher « liste déroulante ».



Régler la taille de la liste sur « Grande » dans l'onglet « Texte ».

Adapter la position des textes et de la liste déroulante pour qu'ils semblent ne former qu'une seule phrase, en veillant à laisser l'espace nécessaire pour le plus long texte de la liste. Pour déplacer la liste, il faut maintenir le clic-droit.

5 - TEST DE LA DIVISIBILITÉ DE a PAR b

Nous allons créer un booléen qui teste si le reste de la division euclidienne de a par b est égal à 0.

Créer un booléen

Saisie : `divisible=Reste(a,b)=0`

6 - VÉRIFICATION DE LA PROPOSITION DE L'UTILISATEUR

Pour vérifier la proposition de l'utilisateur, il nous faut la comparer avec la réponse attendue. Cette comparaison se fera grâce à un booléen qu'on appellera « correct » et dont la valeur dépendra de la divisibilité de a par b ET de la réponse proposée par l'utilisateur.

Saisie : `correct=(divisible&&PositionSélectionnée(L)==2)||
(!divisible&&PositionSélectionnée(L)==3)`

⚠ Ce script a été écrit sur deux lignes pour faciliter la lecture mais il faut l'écrire sans retour à la ligne sur GeoGebra !

Cette définition est en deux parties séparées par le symbole « || » qui est obtenu en utilisant deux fois de suite la combinaison de touches `Alt Gr` + `6` sur un clavier français ou en insérant un symbole « V » à l'aide du bouton  à droite de la barre de saisie.³

3. Ces symboles « || » et « V » correspondent au connecteur logique « ou ».

- La première partie « $\text{divisible} \&\& \text{PositionSélectionnée}(L) == 2$ »⁴ teste si a est divisible par b ET si la position sélectionnée par l'utilisateur dans la liste déroulante L est 2, c'est-à-dire si c'est la deuxième proposition (« est divisible ») qui a été choisie.
- La deuxième partie « $! \text{divisible} \&\& \text{PositionSélectionnée}(L) == 3$ » teste si a n'est pas divisible par b (le symbole « ! » devant un booléen appelle sa négation) ET si la position sélectionnée par l'utilisateur dans la liste déroulante L est 3, c'est-à-dire si c'est la troisième proposition (« n'est pas divisible ») qui a été choisie.

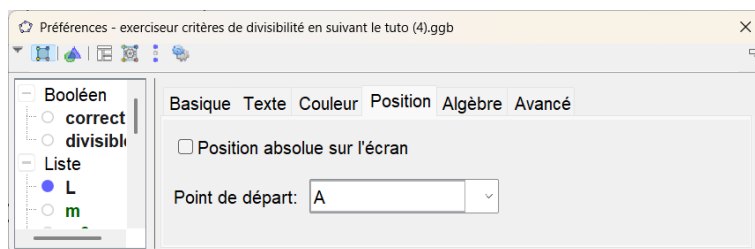
L'ensemble de ces deux tests connectés par un « ou » retourne une valeur « true » (vrai) ou « false » (faux) qui est affectée au booléen « correct ».

7 - AFFICHAGE D'UN MESSAGE DE VÉRIFICATION

Créer deux textes « Bravo » et « Dommage ».

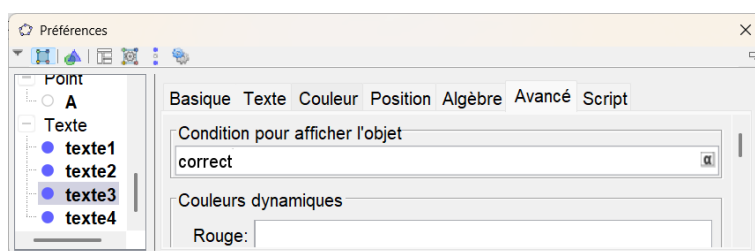
Placer ces deux textes au même endroit . Pour cela, créer un point A à l'endroit où les textes doivent être placés puis, dans les propriétés des deux textes, dans l'onglet « Position », choisir comme point de départ A.

Positionner
plusieurs
textes au
même
endroit



Enlever l'affichage du point A.

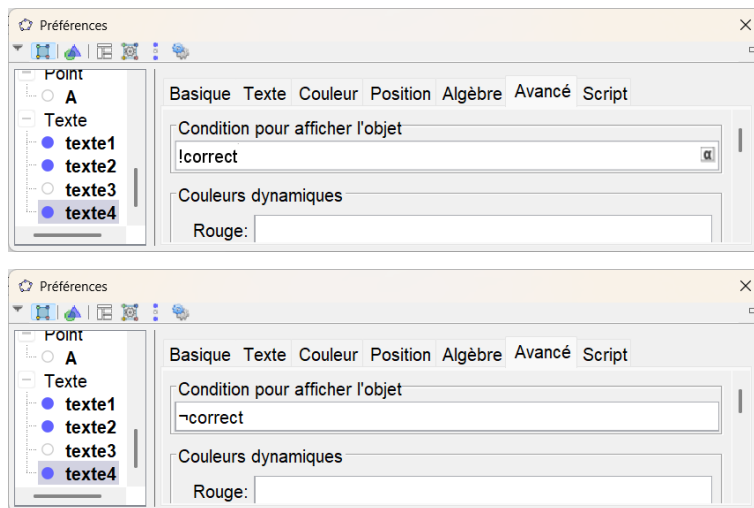
Le texte « Bravo » doit s'afficher quand la proposition est correcte. Dans l'onglet « Avancé » des propriétés du texte, sur la ligne « Condition pour afficher l'objet », taper `correct`.



De même, le texte « Dommage » doit s'afficher quand la proposition est incorrecte. L'affichage de ce texte doit être conditionné par `!correct` que GeoGebra transformera en « $\neg \text{correct}$ »⁵ (voir captures d'écran suivantes).

4. Les symboles « && » et « $\wedge$ » correspondent au connecteur logique « et ».

5. De façon générale, le point d'exclamation devant un booléen transforme celui-ci en son contraire. Il peut être apposé également devant un signe comme = ou < ou > pour signifier sa négation, par exemple « $\neq$ » signifie « différent » et « $>$ » signifie « inférieur ou égal ».



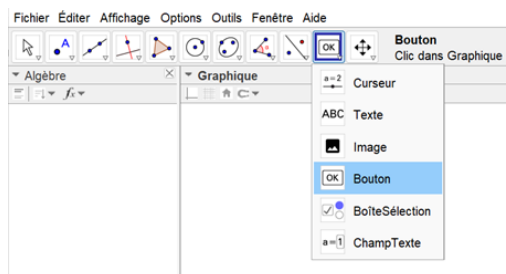
On peut changer la couleur de l'arrière-plan du texte « Bravo » en vert et celle du texte « Dommage » en orange dans les propriétés de chaque objet et régler leur taille sur « Grande ».

8 - INTERDIRE À L'UTILISATEUR LA CORRECTION EN CAS D'ERREUR

Dans l'état actuel, la validation se fait dès que l'utilisateur a choisi sa réponse dans la liste déroulante. Il est alors possible pour l'utilisateur de se corriger pour donner l'autre réponse en cas d'erreur.

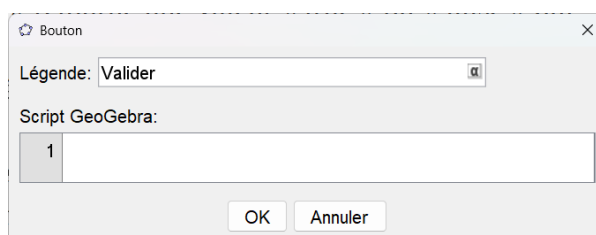
Pour éviter cela, nous allons créer un bouton permettant à l'utilisateur de demander la validation de sa proposition.

Création
d'un bouton



Une fois l'outil « Bouton » activé, cliquer dans la fenêtre « Graphique » à l'endroit où doit être placé le bouton.

Dans la légende, taper **Valider** et laisser le script vide pour l'instant. Cliquer sur « OK ».



Par défaut, GeoGebra n'affiche pas les boutons dans la fenêtre Algèbre. Pour pouvoir les visualiser, il faut cliquer sur l'icône « Objets auxiliaires »  pour l'activer, elle doit être encadrée en bleu pâle : .

Ce bouton est automatiquement nommé « Bouton1 ». À l'aide d'un clic droit, le renommer « Valider » pour plus de clarté.

Ce bouton aura deux missions :

- signaler que l'utilisateur a donné et validé sa réponse pour permettre l'affichage du texte de validation,
- empêcher l'utilisateur de corriger sa réponse.

Pour ce faire, nous allons créer un booléen « valid » initialisé à « false ».

Saisie : `valid=false`

Ce booléen prendra la valeur « true » lorsque l'utilisateur aura validé sa proposition. Dans les propriétés du bouton « Valider », dans l'onglet « Script », sous-onglet « Par clic », ajouter la commande `SoitValeur(valid,true)`.

Ce booléen « valid » va aussi conditionner l'affichage des messages de validation.

Modifier les conditions d'affichage du texte « Bravo » par `valid&&correct` et celles du texte « Dommage » par `valid&&!correct`.

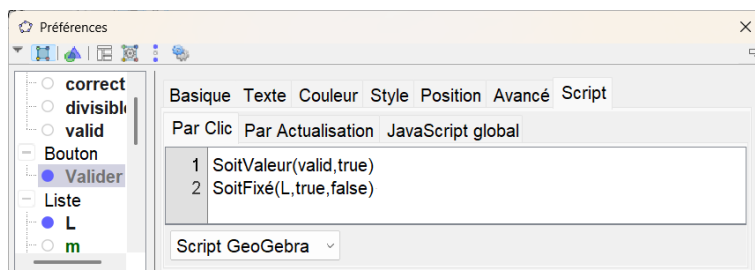
Pour empêcher l'utilisateur de modifier sa réponse, nous allons utiliser la commande `SoitFixé`.⁶

Dans les propriétés du bouton, onglet « Script », ajouter la commande

`SoitFixé(L,true,false)`

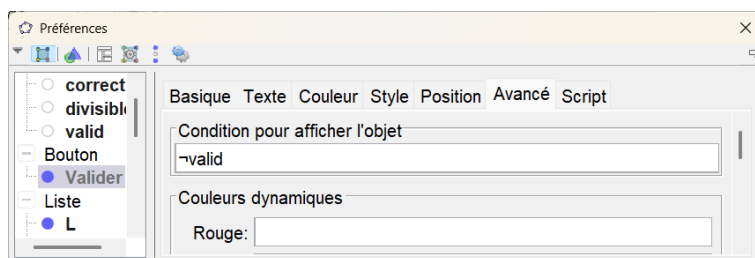
Cette commande permet non seulement de fixer la position de la liste déroulante (pour que l'utilisateur ne la déplace pas par mégarde) mais aussi de la rendre non sélectionnable. L'utilisateur ne pourra donc plus la dérouler et changer sa réponse.

Interdire le déplacement et la sélection d'un objet



⚠ Il faut toujours fermer la fenêtre « Propriétés... » d'un bouton pour pouvoir le tester !

Enfin, il faut que le bouton « Valider » ne soit plus visible après que l'utilisateur a cliqué dessus. Dans l'onglet « Avancé » des propriétés de ce bouton, conditionner son affichage par `!valid`.

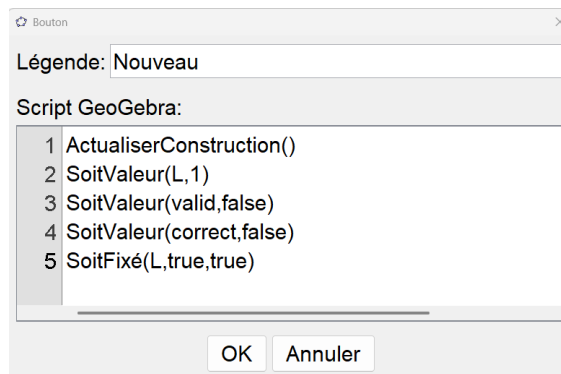


6. `SoitFixé(<Objet>,<fixé true/false>,<sélectionnable true/false>)` rend l'objet fixe (true) ou modifiable (false), sélectionnable (true) ou non (false).

9 - CRÉATION DU BOUTON PERMETTANT UNE AUTRE QUESTION

Créer un nouveau bouton dont la légende est « Nouveau » et lui affecter le script suivant.

```
1 ActualiserConstruction()
2 SoitValeur(L, 1)
3 SoitValeur(valid, false)
4 SoitValeur(correct, false)
5 SoitFixé(L, true, true)
```



Détaillons ce que fait ce script :

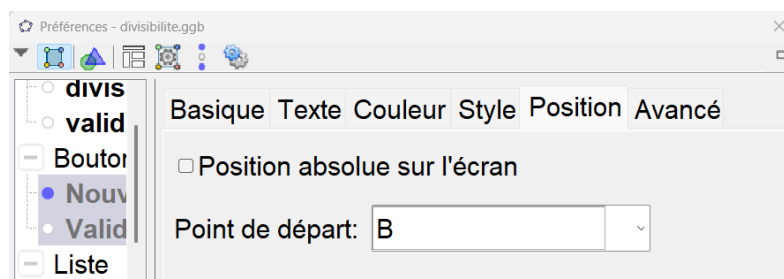
- La ligne 1, « ActualiserConstruction() » permet de recalculer toutes les valeurs aléatoires (a, b et c).
- Dans les lignes suivantes, on réinitialise L, valid et correct.
La commande « SoitValeur(L, 1) » permet de sélectionner le premier élément de la liste déroulante, c'est-à-dire le texte vide.
La commande « SoitFixé(L, true, true) » permet de rendre la liste déroulante à nouveau sélectionnable.

Renommer ce bouton « Nouveau ».

Conditionner l'affichage de ce bouton par le booléen « valid ».

À l'aide d'un nouveau point B, positionner les deux boutons au même endroit.

Il faut décocher la case « Position absolue sur l'écran » dans l'onglet « Position » des propriétés de chaque bouton.



Enlever l'affichage du point B.

10 - CRÉATION DU SCORE ET LIMITATION DU NOMBRE DE QUESTIONS

À ce stade, vous pourriez souhaiter :

- afficher un score sous la forme « nombre de réussites / nombre d'essais », pour cela vous pouvez suivre le **tutoriel dédié** ;
- fixer le nombre de questions accessibles via le bouton « Nouveau », pour cela vous pouvez suivre le **tutoriel dédié**.

11 - FINITIONS

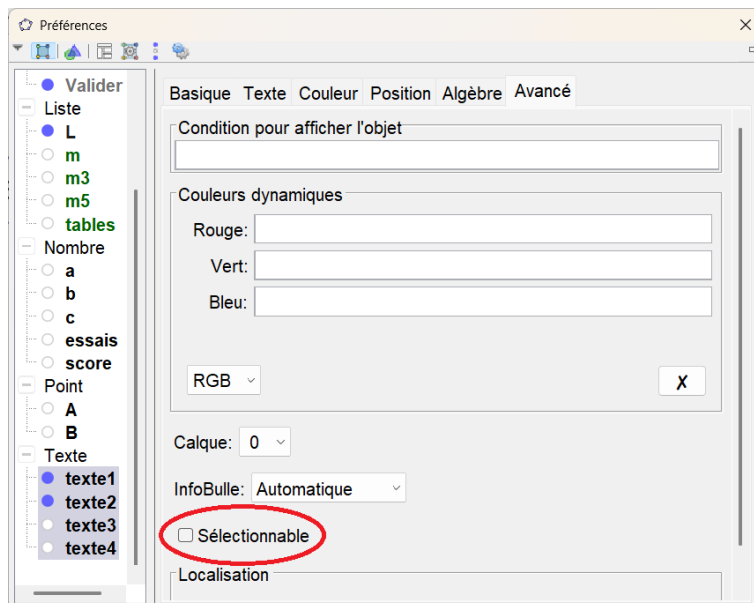
Notre exerciceur est presque terminé.

Si besoin, replacer tous les éléments pour créer quelque chose d'esthétique et d'ergonomique (éventuellement, rendre de nouveau visible temporairement les points A et B).

Pour éviter que l'utilisateur ne modifie ou ne supprime des éléments (volontairement ou non), il convient de rendre les textes non sélectionnables.

Sélectionner tous les textes (clic droit sur le mot « texte » dans la fenêtre « Algèbre ») puis, dans leurs propriétés, dans l'onglet « Avancé », décocher la case « Sélectionnable ».

Rendre un objet non sélectionnable



Avant d'enregistrer, fermer la fenêtre « Algèbre »...

Il ne vous reste plus qu'à sauvegarder votre fichier.

12 - DONNER L'EXERCISEUR AUX ÉLÈVES

Pour cela vous pouvez **enregistrer votre activité** dans votre compte GeoGebra et **créer une leçon**.



Exercice d'application

Modifier le script du bouton « Valider » pour que le message « Entrer une réponse avant de valider. » s'affiche tant que l'utilisateur n'a pas modifié la liste déroulante.

INDEX DES FONCTIONS UTILISÉES

Source : <https://geogebra.github.io/docs/manual/fr/>

- **ActualiserConstruction()**
Recalcule tous les objets utilisés dans le fichier GeoGebra.
- **AléaEntreBornes(<Entier minimum a>,<Entier maximum b>)**
Retourne un nombre entier aléatoire de l'intervalle [a ; b].
- **ElémentAuHasard(<Liste>)**
Retourne un élément choisi avec une probabilité uniforme dans la liste. Tous les éléments doivent être de même type.
- **Reste(<Dividende D>,<Diviseur d>)**
Calcule le reste de la division euclidienne de l'entier D par l'entier d.
- **Séquence(<Expression e>,<Variable k>,<de a>,<à b>)**
Retourne la liste des objets créés en utilisant l'expression e et l'indice k variant du nombre a au nombre b.
- **Si(<Condition>,<Objet a>,<Objet b>)**
Retourne une copie de l'objet a si la condition prend la valeur true (vrai), et une copie de l'objet b si elle prend la valeur false (faux).
- **SoitFixé(<Objet>,<fixé true|false>,<sélectionnable true|false>)**
Rend l'objet fixe (true) ou modifiable (false), sélectionnable (true) ou non (false).
- **SoitValeur(<Booléen b>,<false|true>) ou SoitValeur(<Booléen b>,<0|1>)**
Affecte au booléen b la valeur false (0) ou true (1).
- **SoitValeur(<Liste L>,<Nombre n >,<Objet O>)**
Affecte au n^e élément de la liste L la valeur actuelle de l'objet O. Le nombre n est au plus égal à Longueur(L)+1.