

# Logoplus - Catalogue des mots-clés (primitives).

V06.08.11.45.19 et ultérieures

<http://logoplus.pagesperso-orange.fr>

[Vers l'index des primitives >>](#)

Classement des primitives par rubrique :

Pour définir le travail de LOGO:

POUR

Pour dessiner avec la tortue:

AVANCE , BAISSECRAYON , CACHETORTUE , COULEURFOND , COULEURTORTUE ,  
EFFACECRAN , LEVECRAYON , MONTRETORTUE , NETTOIE , ORIGINE , POINT , RECULE ,  
RVB , TRAIT

Pour orienter la tortue:

CAP , COMPAS , FIXECAP , TOURNEDROITE , TOURNEGAUCHE , VERS

Pour positionner la tortue :

AUCUNE , DEROULE , ENROULE , FIXEX , FIXEXY , FIXEY , FORME , OBEIS , LIEU ,  
LIMITES? , RAMENE , SITUATION , TORTUES? , TOUTES , VA , XCOORD , YCOORD

Pour contrôler les mouvements de la tortue:

CAPTEURAVANT? , CAPTEURDROIT? , CAPTEURGAUCHE? , COULEURTORTUE? ,  
VISIBLE?

Pour calculer:

[ + - / \* ^ ] , [ . : ] , ARRONDIR , DEC , DIFFERENCE , DIV , ENTIER , INC , MAX? , MIN? ,  
MOD (%) , PI , PRODUIT , PUISSANCE , QUOTIENT , RESTE (%) , SOMME

**Pour travailler sur les nombres:**

<, <=, >, >=, \$, =, [ eE ], [ , ], [ ( ) ], CARTEPOINTS

**Pour travailler sur les phrases:**

[ ], CARDINAL, CHOIX, DERNIER, INSERE, ITEM, LISTE, LISTEVARS, MEMBRE?, OCCURRENCES, PERMUTATIONS, PHRASE, PREMIER, REPLACE, SAISIS, SAUFPREMIER, SAUFDERNIER, ROTATION, SCINDE, SWAP, TRIEITEMS, VICEVERSA .

**Pour travailler sur les mots:**

[ " ], DERNIERELETTRE, MOT, PREMIERELETTRE, SAUFDERNIERELETTRE, SAUFPREMIERELETTRE, SEPARE, TRIECAR .

**Pour vérifier un calcul:**

DIFFERENTS?, EGALITE?, NEGATIVE?, SI, SIVRAI, SIFAUX (SINON), TESTE, ZERO?

**Pour vérifier plusieurs choses:**

ET(&), OU(!), LESDEUX?, UNDESDEUX?

**Pour faire écrire LOGO:**

AFFICHE, DESSINETEXTE, ECRIS, ECRISLIGNE, EFFACETEXTE, COLORIELETTRE, STYLELETTRE, TAILLELETTRE, xx, "", ||, /e /E, x", ""x

**Pour imprimer le travail de LOGO:**

IMPRIMEEcran, IMPRIMETEXTE

**Pour faire lire un nombre, un mot, une phrase:**

EFFACELIGNE, LISCAR, LISLISTE, LISMAT, LISNOMBRE, TOUCHE?

**Pour effectuer une consigne plusieurs fois:**

REPETE , TANTQUE (TANT QUE) , BOUCLE

Pour donner un nom à une valeur:

DONNE / SOIT , DEFINIS .

Pour connaître la valeur d'une donnée:

[ : ] , CHOSE , COPIE , PARTAGE , PERMUTE

Pour répondre à un calcul:

REPONDRE

Pour arrêter le travail de LOGO:

FIN , RETOURNE , STOP

Pour accélérer ou ralentir le travail de LOGO:

ACCELERE , ATTENDS , RALENTIS , SUSPENDRE

Pour utiliser la souris:

CLIC? , SOURISX , SOURISY , SURECRAN?

Pour vérifier le texte de l'éditeur:

CONTENU , PROCEDURES?

Lorsqu'il n'y a que deux réponses possibles:

BOOLEEN , FAUX , LISTE? , MOT? , NOMBRE? , VIDE? , VRAI

Pour varier le travail de LOGO:

FIXEHASARD , HASARD , RANDDEC , RANDOM

Pour suivre le travail de LOGO:

PASAPAS , TRACE , SILENCE

Pour faire se déplacer un lutin:

AVANCELUTIN , CAPLUTIN , CAPLUTIN? , CACHELUTIN , ENDORS , FIXELUTINX ,  
FIXELUTINXY , FIXELUTINY , LUTINVERS , LUTIN VX , LUTIN VY , LUTINX? , LUTINY? ,  
MONTRELUTIN , POSITIONLUTIN? , REVEILLE , RECULELUTIN , TOURNELUTINDROITE ,  
TOURNELUTINGAUCHE , VELOCITE? , VITESSE

Pour colorier un lutin:

COLORIELUTIN

Lorsque plusieurs lutins se rencontrent:

COLLISIONS?

Pour colorier une région de l'écran:

COLORIE , PIXELS?

Pour vérifier une couleur:

COULEUR? , COULEURFOND? , COULEURLUTIN? , COULEURPOINT?

Pour jouer de la musique:

JOUE , TEMPO

Pour mettre en valeur le travail de LOGO : les couleurs.

[ [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#) ]

Voir également la liste des codes [couleurs Windows](#).

Pour commenter le texte de l'éditeur:

[ [/\\* \\*/](#) ] , [//](#)

Les primitives spécialisées en mathématiques.

[ABS](#) , [ARCCOS](#) , [ARGCOSH](#) , [ARCCOT](#) , [ARGCOTH](#) , [ARCCSC](#) , [ARGCSCH](#) , [ARCSEC](#) , [ARGSECH](#) , [ARCSIN](#) , [ARGSINH](#) , [ARCTAN](#) , [ARGTANH](#) , [COMPLEXE](#) , [CONJUG](#) , [CPLXADD](#) , [CPLXPROD](#) , [COSH](#) , [COSEC](#) , [COSINUS](#) , [COTAN](#) , [COTH](#) , [CSCH](#) , [DEGRE](#) , [DETERMINANT](#) , [DIVISEURS](#) , [ECART\\_TYPE](#) , [EXP](#) , [FVECT](#) , [HEXA?](#) , [IMAGE](#) , [INTEGRALE](#) , [LN](#) , [LOG](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [M\\_ALEA](#) , [M\\_DIAG](#) , [M\\_DIFF](#) , [M\\_IDENT](#) , [M\\_INVERSE](#) , [M\\_NORME](#) , [M\\_NULLE](#) , [M\\_SOMME](#) , [M\\_SUPPR](#) , [M\\_TRACE](#) , [M\\_TRANSPOSE](#) , [M\\_TRIINF](#) , [M\\_TRISUP](#) , [NON](#) , [OPPOSE](#) , [POLYSOLVE](#) , [PRODEXT](#) , [PRODINT](#) , [RACINE2](#) , [RACINE3](#) , [RADIAN](#) , [REEL](#) , [SECANTE](#) , [SECH](#) , [SENHORAIRE](#) , [SENSTRIGO](#) , [SINUS](#) , [SINH](#) , [TAN](#) , [TANH](#) , [TRANSLATE](#) , [VARIANCE](#) , [VCOS](#) , [VDIM](#) , [VPROD](#) , [VSIN](#) , [V\\_PROPRES](#) , [V\\_UNIQUE](#) , [ZEROS?](#)

Pour dessiner autrement:

[ARC](#) , [BARRES](#) , [CHAMPVECT](#) , [ELLIPSE](#) , [FONDECRAN](#) , [GRAPHE](#) , [GRAPH2D](#) , [GRAPH3D](#) , [GRAPHEUR](#) , [HISTO](#) , [INITGRAPH](#) , [PLOT](#) , [PLOTPOL](#) , [SEGMENT](#) , [SEGMENT3D](#) , [SURFACE](#) , [SYSPLOT](#) , [SYSPLOTPOL](#) , [ZMAX](#) , [ZMAX?](#) , [ZPOINT](#) , [ZPOINT?](#) , [ZSEGMENT](#)

Pour situer un événement:

[DATE](#) , [TEMPS](#)

Pour agir sur l'extérieur (port série):

[CACHESPOTS](#) , [MONTRESPOTS](#) , [SPOTS](#) , [LISPORT](#) , [OUVREPORT](#) , [PORT](#) , [PORTS?](#)

Pour agir sur l'extérieur (port USB):

[CABLEUSB?](#) , [CANAL?](#) , [CANAL\\_MAX](#) , [CANAL\\_MIN](#) , [CANAUX?](#) ,  
[CANAUX\\_MAX](#) , [CANAUX\\_MIN](#) , [CARTEUSB](#) , [CONNECTE](#) , [CONNEXIONS](#) , [CPUOK?](#) ,  
[DECONNECTE](#) , [EXECDLL](#) , [FERMECANAL](#) , [FERMECANAUX](#) , [FERMECARTE](#) , [FERMEDLL](#) ,  
[FIXECANAL](#) , [FIXECANAL\\_L](#) , [FIXECANAUX\\_A](#) , [FIXECANAUX\\_L](#) , [FIXEMODULATION](#) ,  
[FIXEREBOND](#) , [INIT\\_COMPTEUR](#) , [MODULATION?](#) , [OUVERT?](#) , [OUVERTS?](#) , [OUVERTS\\_A?](#) ,  
[OUVRECANAL](#) , [OUVRECANAUX](#) , [OUVREDLL](#) , [TOPS?](#) , [VERSION?](#) , [WINUSB1](#) , [WINUSB2](#)

**Pour utiliser des fichiers ou des données:**

[CHANGEREP](#) , [CREEREP](#) , [RAPPELLE](#) , [SAUVE](#) , [SUPPREP](#) , [SUPPRIME](#)

**Les primitives moins utilisées:**

[ASCII](#) , [CARACTERE](#) , [DETRUIS](#) , [FAIRE](#) , [NOMBRE](#) , [PLEINECRAN](#) , [SAUVEECRAN](#) ,  
[TRANSFORME](#)

**Liste des abréviations :**

Primitive	Abrév.
AVANCE	AV
BAISSECRAYON	BC
CACHETORTUE	CT
CAPLUTIN	CPL
CARACTERE	CAR
CARDINAL	CARD
CACHELUTIN	CL
COSINUS	COS
COULEURFOND	FCFG
COULEURFOND?	CF
COULEURTORTUE	FCC
COULEURTORTUE?	CC
DEFINIS	DEF
DERNIER	DER
DERNIERELETTRE	DERL
DIFFERENCE	DIFF
DIFFERENTS?	DIFF?
DONNE	SOIT
ECRIS	EC
ECRISLIGNE	ECL
EFFACEECRAN	EFF

<b>EFFACELIGNE</b>	<b>EFFL</b>
<b>EFFACETEXTE</b>	<b>EFFTXT</b>
<b>ENROULE</b>	<b>ENR</b>
<b>FIXECAP</b>	<b>FCAP</b>
<b>FIXEMODULATION</b>	<b>FIXEPWM</b>
<b>FIXEPOSITION</b>	<b>FPOS</b>
<b>IMPRIMEECRAN</b>	<b>IM</b>
<b>IMPRIMETEXTE</b>	<b>IMTXT</b>
<b>LEVECRAYON</b>	<b>LC</b>
<b>LISLISTE</b>	<b>LL</b>
<b>MONTRELUTIN</b>	<b>ML</b>
<b>MONTRETORTUE</b>	<b>MT</b>
<b>PHRASE</b>	<b>PH</b>
<b>PREMIER</b>	<b>PREM</b>
<b>PREMIERELETTRE</b>	<b>PREML</b>
<b>PRODUIT</b>	<b>PROD</b>
<b>QUOTIENT</b>	<b>QUOT</b>
<b>RECULE</b>	<b>RE</b>
<b>REPONDR</b>	<b>RENDS</b>
<b>RESTE</b>	<b>% (infixé)</b>
<b>RETOURNE</b>	<b>RET</b>
<b>ROTATION</b>	<b>ROT</b>
<b>SAUFDERNIER</b>	<b>SD</b>
<b>SAUFDERNIERELETTRE</b>	<b>SDL</b>
<b>SAUFPREMIER</b>	<b>SP</b>
<b>SAUFPREMIERELETTRE</b>	<b>SPL</b>
<b>SINUS</b>	<b>SIN</b>
<b>TOURNEDROITE</b>	<b>TD</b>
<b>TOURNEGAUCHE</b>	<b>TG</b>
<b>TOURNELUTINDROITE</b>	<b>TLD</b>
<b>TOURNELUTINGAUCHE</b>	<b>TLG</b>
<b>VICEVERSA</b>	<b>VV</b>

**Codes Windows correspondant à une couleur nominale en LOGO:**

Nom :	Code Windows
BLANCH E	16777215
BLEUE	16711680
JAUNE	65535
NOIRE	0
ROUGE	255
VERTE	32768
TILLEUL	65280
EAU	16776960
FUCHSIA	16711935

SARCELLE	8421376
GRISE	8421504
OLIVE	32896
MARRON	12632256
ARGENT	128
ORANGE	8421631
ROSE	16744703
VIOLETTE	8388736

[Vers l'index des primitives >>](#)

[ ] :

Délimiteur de liste. Une liste est un ensemble d'items (mots, nombres ou listes) séparés par un espace, situés entre crochets. Exemple: [ un 2 3 [2+2] ] est une liste contenant quatre items (le quatrième item est aussi une liste).

n.b: Il existe des combinaisons de caractères utiles à connaître pour afficher

des caractères particuliers à l'intérieur d'une liste :

Touche(s) :	⌘⌘ ( <Alt gr + \$> x2)	⌘⌘⌘ ( <shift +touche ^ > x2)	⌘⌘ ( <Alt gr + 6> x2)
caractère affiché:	[	]	"

Touche(s) :	/e /E
caractère affiché:	(espace)

Touche(s) :	⌘⌘ <Alt gr> + (<touche ^> x2)	⌘⌘ (<touche ^> x2) + <Alt gr>
Action réalisée :	Effacement de la dernière ligne de l'afficheur.	Effacement du dernier caractère de l'afficheur.

/e ou /E :

Signet d'insertion d'espaces dans une liste, immédiatement suivi d'un nombre entier. Un signet crée lors de l'affichage de la liste autant d'espaces entre deux items consécutifs que souhaités. Les signets d'insertion d'espaces sont logiquement invisibles à l'exécution d'un programme LOGO et ne sont pas comptés en tant qu'items d'une liste.

Exemple: **ECRIS** [ item1 item2 /e8 item3 ] occasionnera l'affichage de:

item1 item2      item3      (avec 8 espaces entre item2 et item3.)

n.b.: **ECL CARD** [ item1 /e2 item2 /e4 item3 ] donnera 3 en résultat.

Voir [LISTE](#) , [LISTE?](#) , [ITEM](#) .

[Début du catalogue.](#)

␣ :

Signet d'effacement de la dernière ligne de l'afficheur, réalisé en appuyant sur <Alt gr + touche \$> et sur la touche <shift +touche ^> en début de liste ou de mot passé en argument à ECRIS ou ECRISLIGNE . Le ou les signets d'effacement doivent être placés en début de liste si liste il y a, accolés ou séparés.

Exemples:

**ECL** "␣␣␣Bonjour.

**ECL** [␣␣␣ Bonjour.]

**ECL** [␣ ␣␣ Bonjour.]

effaceront les deux dernières lignes de l'afficheur (si elles existent) avant d'afficher Bonjour.

Voir [LISTE](#) , [LISTE?](#) , [ITEM](#) .

[Début du catalogue.](#)

␣' :

Signet d'effacement du dernier caractère de l'afficheur, réalisé en appuyant sur <shift +touche ^> et sur la touche <Alt gr + touche \$> en début de liste ou de mot passé en argument à ECRIS ou ECRISLIGNE . Le ou les signets d'effacement doivent être placés en début de liste si liste il y a, accolés ou séparés.

Exemples:

**ECL** "'␣␣␣Bonjour.

**ECL** [␣␣␣␣ Bonjour.]

**ECL** [␣␣ ␣␣ Bonjour.]

effaceront les deux derniers caractères de l'afficheur (si ils existent) avant d'afficher Bonjour.

Voir [LISTE](#) , [LISTE?](#) , [ITEM](#) .

[Début du catalogue.](#)

Les parenthèses ( ) :

Délimiteurs d'expression logique.

Exemple: **DONNE** "MaQuantité ((5+7)\*2)

:MaQuantité reçoit 24 comme valeur.

## A retenir :

S'il est d'usage de considérer l' espace comme séparateur d'instructions (ou primitives), les parenthèses, elles, ont un rôle important en LOGO pour délimiter les calculs, notamment pour séparer plusieurs items numériques qui comportent un signe négatif, cela dans un appel de procédure ou dans une liste.

Exemple: `FIXEXY 50 (-80)`

[Début du catalogue.](#)

## Les guillemets " :

Délimiteur de mot ou indicateur de nom de variable. Voir [CHOSE](#), [MOT](#), [MOT?](#), [^^](#) || ....

Exemple: `DONNE "MonMot" "crayon" ECRIS :MonMot`

affichera le contenu de la variable :MonMot -> crayon .

[Début du catalogue.](#)

## [Vers l'index des primitives >>](#)

`:` : Indicateur du contenu d'une variable. Voir [CHOSE](#), [MOT](#) , [LISTE](#) , [NOMBRE](#) .

Exemple: `DONNE "MonMot" "crayon"`  
`ECRIS :MonMot`

affichera le contenu de la variable :MonMot -> crayon .

**A retenir :** Il est quelquefois possible d'utiliser des noms de données à l'intérieur d'une liste, pour rappeler leur contenu (ou valeur) ensuite. Dans ce cas, il est nécessaire de ne pas séparer par un espace le prédicat du nom de cette donnée ( : ) du reste de ce nom, ce qui aurait à la fois pour conséquence de créer deux items et de rendre le nom de la donnée illisible.

Exemple: `DONNE "donnée -8`  
`ECL ITEM [ 5 :donnée ] 2 // donnera -8`  
`ECL ITEM [ 5 : donnée ] 2 // donnera un autre résultat`

[Début du catalogue.](#)

`/* */`: Délimiteur de commentaires.

Exemple: `/* Ceci est un commentaire. */`

Ce qui est écrit entre les symboles /\* et \*/ est ignoré à l'exécution du programme LOGO. Ces délimiteurs fonctionnent toujours par paires.

### [Début du catalogue.](#)

// : Délimiteur de commentaires en fin de ligne.

Exemple: AV 50 TD 90 // Le reste de la ligne est ignoré.

### [Début du catalogue.](#)

' (virgule) / . (point):

Dans un nombre décimal, la virgule est considérée d'emblée comme séparateur entre la partie entière et la partie décimale, conformément à la norme d'écriture des nombres adoptée en France. Toutefois l'usage du point est toléré. Dans ce cas, lorsque le point est détecté comme séparateur décimal à un endroit du programme ou d'une consigne, l'affichage des données numériques se fera en utilisant aussi le point comme séparateur décimal. Cependant, si une consigne qui contient des nombres à virgule se réfère à une procédure de l'éditeur qui, elle, utilise le point dans l'expression d'un nombre, l'affichage ultérieur de résultats numériques se fera en utilisant également le point comme séparateur décimal. D'une manière générale, c'est le programme contenu dans l'éditeur qui fixe prioritairement le choix du séparateur décimal selon que dans sa rédaction, il se réfère à l'un des deux ou aux deux indifféremment.

Exemple: DONNE "nombre 5,3 // éq: SOIT "nombre 5,3 ( ou SOIT "nombre 5.3 )

### [Début du catalogue.](#)

+ - \* / : opérateur respectivement:

+ : additif.

- : soustractif.

\* : multiplicatif.

/ : de division.

Voir [DEC](#) , [DIFFERENCE](#) , [INC](#) , [PRODUIT](#) , [QUOTIENT](#) , [RESTE \(%\)](#) , [SOMME](#) .

A propos du signe - : s'il est d'usage de considérer l'espace comme séparateur d'instructions (ou primitives), le langage LOGO requiert également beaucoup de parenthèses pour délimiter les calculs, notamment pour séparer plusieurs items numériques dans une liste, et obligatoirement lorsque l'un d'eux commence par un signe négatif, qui peut être interprété comme un signe opératoire.

Exemple: SEGMENT [ 5 (-8) ] [ -13 9 ]

[Début du catalogue.](#)

< : Comparateur d'infériorité.

Voir [<=](#) , [>](#) , [>=](#) , [=](#) , [EGALITE?](#) , [DIFFERENTS?](#) .

Exemple:

```
DONNE "n 2 // éq: SOIT "n 2
TESTE (:n < 5 )
SIVRAI ECRISLIGNE LISTE :n [ est plus petit que 5. ]
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

<= : Comparateur d'infériorité au sens large.

Voir [<](#) , [>](#) , [>=](#) , [=](#) , [EGALITE?](#) , [DIFFERENTS?](#) .

Exemple:

```
DONNE "n 2 // éq: SOIT "n 2
TESTE (:n <= 5 )
SIVRAI ECRISLIGNE LISTE :n [ est plus petit que 5. ]
```

[Début du catalogue.](#)

> : Comparateur de supériorité. Voir [<](#) , [<=](#) , [>=](#) , [=](#) , [EGALITE?](#) , [DIFFERENTS?](#).

Exemple:

```
DONNE "n 5 // éq: SOIT "n 5
TESTE (:n > 2 )
SIVRAI ECRISLIGNE LISTE :n [ est plus grand que 2. ]
```

[Début du catalogue.](#)

>= : Comparateur de supériorité au sens large. Voir [<](#) , [<=](#) , [>](#) , [=](#) , [EGALITE?](#) , [DIFFERENTS?](#).

Exemple:

```
DONNE "n 5 // éq: SOIT "n 5
TESTE (:n >= 2 )
SIVRAI ECRISLIGNE LISTE :n [ est plus grand que 2. ]
```

[Début du catalogue.](#)

\$ : Prédicat de la notation hexadécimale (base 16). Voir [HEXA?](#) , [TRANSLATE](#)

Exemple: ECRISLIGNE \$A3F9.

[Début du catalogue.](#)

≡ : Comparateur d'égalité.

Voir [<](#) , [>](#), [EGALITE?](#), [DIFFERENTS?](#).

Exemple:

SOIT "n 8 // éq: DONNE "n 8

TESTE (:n = 8 )

SIVRAI ECRISLIGNE LISTE :n [ est égal à 8. ]

[Début du catalogue.](#)

ABS :

Fonction ABS <arg num> ou ABS [liste\_vecteur] - fournit en résultat la valeur absolue du nombre <arg num> ou de la liste vectorielle [liste\_vecteur] (une liste vectorielle est une liste à une seule dimension qui ne doit contenir que des nombres). Voir [DEC](#) , [INC](#) , [MAX?](#), [MIN?](#) .

Exemple: ECRIS ABS (-4,5678)

rendra la valeur 4,5678 comme résultat.

ECRIS ABS [ 2 (-3) 4]

rendra la valeur 5,385164807 comme résultat.

[Début du catalogue.](#)

ACCELERE :

Procédure ACCELERE : accélère la vitesse de déplacement de la tortue.

N'affecte en aucun cas la rapidité des calculs numériques. Voir [RALENTIS](#), [CAPTEURAVANT?](#), [CAPTEURGAUCHE?](#), [CAPTEURDROIT?](#) , [PASAPAS](#) .

[Début du catalogue.](#)

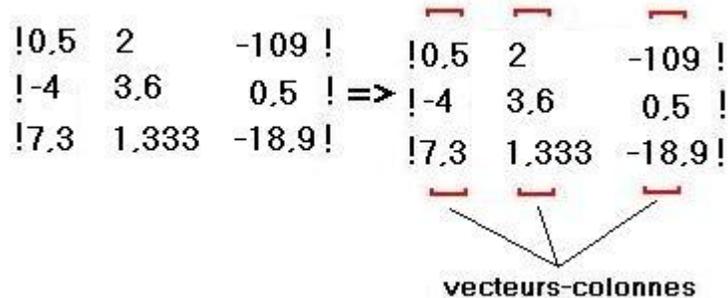
AFFICHE :

Procédure AFFICHE <matrice> ; occasionne l'affichage de <matrice> après un saut de ligne dans l'afficheur des résultats. Les côtés verticaux de la matrice sont alors représentés par des points d'exclamation. Rappel: <matrice> est une liste qui ne contient que des nombres et dont les sous-listes (vecteurs-colonnes) ont les mêmes dimensions. Pour afficher correctement les nombres décimaux à période, la procédure réduit la longueur de la partie décimale en précisant toutefois si le dernier chiffre seul est répété en le faisant suivre par des pointillés (dans ce cas, la partie décimale sera tronquée à cinq chiffres) ou s'il s'agit d'une période longue en la faisant suivre par un tiret. Voir [DETERMINANT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#),

[M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#),  
[V\\_PROPRES](#), [V\\_UNIQUE](#), [PASAPAS](#) , [PRODEXT](#), [PRODINT](#) .

Exemple:

AFFICHE [[0,5 (-4) 7,3][2 3,6 1,333][-109 0,5 (-18,9)]] sera affichée :



n.b. : 12,14077777 sera affiché 12,14077...  
 9,734734734 sera affiché 9,734\_

[Début du catalogue.](#)

ARC :

Procédure ARC [ x1 x2 y1 y2 x3 y3 x4 y4 ] dessine une ligne courbe elliptique avec la couleur de la tortue en cours. L'arc traverse le périmètre de l'ellipse circonscrite par les points [ x1 y1 ] et [ x2 y2 ]. L'arc est dessiné en suivant le périmètre de l'ellipse, dans le sens contraire des aiguilles d'une montre jusqu'au point d'arrivée. Le point de départ est défini par l'intersection de l'ellipse et d'une ligne définie par le centre de l'ellipse et [ x3 y3 ]. Le point d'arrivée est défini par l'intersection de l'ellipse et d'une ligne définie par le centre de l'ellipse et [ x4 y4 ]. Voir [ELLIPSE](#) , [FONDECRAN](#) , [SEGMENT](#) .

Exemple: ARC [ (-50) 40 20 (-60) 0 30 (-10) (-30) ]

[Vers l'index des primitives >>](#)

[Début du catalogue.](#)

ARCCOS :

Fonction ARCCOS <arg num> - calcule la valeur de l'arc cosinus de <arg num>. <arg num> doit être compris entre -1 et 1 (valeurs incluses) . Voir [SINUS](#), [ARCSIN](#), [COSINUS](#), [INTEGRALE](#) , [TAN](#) .

Exemple: MODEDEGRE ECRIS ARCCOS 0,642787609  
 donnera 45 en résultat.

[Début du catalogue.](#)

### ARGCOSH :

Fonction ARGCOSH <arg num> - calcule de la valeur de l'argument cosinus hyperbolique de <arg num>. <arg num> doit être supérieur à 1 (borne incluse). Voir [ARGCOTH](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS **ARGCOSH** 0,642787609      donnera 45 en résultat.

[Début du catalogue.](#)

### ARCCOT :

Fonction ARCCOT<arg num> - calcule la valeur de l'arc cotangente de <arg num>. Voir [SINUS](#) , [ARCCOS](#) , [ARCSIN](#) , [COSINUS](#) , [INTEGRALE](#) , [TAN](#) .

Exemple:

ECRIS **ARCCOT** 0,5      donnera 1,10714871779409 en résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### ARGCOTH :

Fonction ARGCOTH <arg num> - calcule la valeur de l'argument cotangente hyperbolique de <arg num>. <arg num> doit être strictement positif et différent de 1. Voir [ARGCOSH](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS **ARGCOTH** 2      donnera 0,549306144334055 en résultat.

[Début du catalogue.](#)

### ARCCSC :

Fonction ARCCSC <arg num> - calcule la valeur de l'arc cosécante de <arg num>. <arg num> doit être supérieur à 1 (borne incluse). Voir [ARCSEC](#) , [COSEC](#) , [SECANTE](#) .

Exemple:

ECRIS **ARCCSC** 2      donnera 0,523598775598299 en résultat.

[Début du catalogue.](#)

### ARGCSCH :

Fonction ARGCSCH <arg num> - calcule la valeur de l'argument cosécante hyperbolique de <arg num>.

<arg num> doit être différent de 0. Voir [ARGCOSH](#) , [ARGCOTH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS **ARGCSCH** 0,5      donnera 0,972955074527657 en résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**ARCSEC** :

Fonction ARCSEC <arg num> - calcule la valeur de l'arc sécante de <arg num>. <arg num> doit être supérieur à 1 (borne incluse). Voir [ARCCSC](#) , [COSEC](#) , [SECANTE](#).

Exemple:

ECRIS **ARCSEC** 2      donnera 1,0471975511966 en résultat.

[Début du catalogue.](#)

**ARGSECH** :

Fonction ARGSECH <arg num> - calcule la valeur de l'argument sécante hyperbolique de <arg num>. <arg num> doit être compris dans l'intervalle  $[-1..1]-\{0\}$  . Voir [ARGCOSH](#) , [ARGCOTH](#) , [ARGCSCH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS **ARGSECH** 0,5      donnera 1,31695789692482 en résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**ARCSIN** :

Fonction ARCSIN <arg num> - calcule la valeur de l'arc sinus de <arg num>. <arg num> doit être compris sur l'intervalle  $[-1 .. 1]$  . Voir [SINUS](#) , [COSINUS](#) , [ARCCOS](#) , [ARCCOT](#) , [INTEGRALE](#) , [TAN](#) , [ZEROS?](#) .

Exemple:

MODEDEGRE

ECRIS **ARCSIN** 0,258819045      donnera 15 en résultat.

[Début du catalogue.](#)

**ARGSINH** :

Fonction ARGSINH <arg num> - calcule la valeur de l'argument sinus hyperbolique de <arg num>.

Voir [ARGCOSH](#) , [ARGCOth](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGTANH](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

**ECRIS ARGsinh** 0,5      donnera 0,481211825059603 en résultat.

[Début du catalogue.](#)

**ARCTAN** :

Fonction ARCTAN <argnum> - calcule la valeur de l'arc tangente de <argnum>. Voir [SINUS](#) , [COSINUS](#) , [INTEGRALE](#) , [TAN](#) , [ZEROS?](#) .

Exemple: **MODEDEGRE ECRIS ARCTAN** 45      donnera 88,72696998 en résultat.

[Début du catalogue.](#)

**ARGTANH** :

Fonction ARGTANH <arg num> - calcule la valeur de l'argument tangente hyperbolique de <arg num>. <arg num> doit être compris dans l'intervalle ]-1..1[ . Voir [ARGCOSH](#) , [ARGCOth](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGsinh](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

**ECRIS ARGtanh** 0,5      donnera 0,549306144334055 en résultat.

[Début du catalogue.](#)

**ARRONDIR** :

Fonction ARRONDIR <argnum> - calcule la valeur de <argnum> arrondie à l'entier le plus proche. Voir [DEC](#) , [DIV](#) , [ENTIER](#) , [INC](#) , [MAX?](#) , [MIN?](#) .

Exemples: **ECRIS ARRONDIR** 7,75      donnera 8 en résultat, mais  
**ECRIS ARRONDIR** 12,2      donnera 12.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**ASCII** :

Fonction ASCII "arg - fournit en résultat le code ASCII correspondant au mot "arg. Voir [CARACTERE](#) .

Exemple: **ECL ASCII** "A      rend la valeur 65.

[Début du catalogue.](#)

**ATTENDS** :

Procédure ATTENDS <argnum> - suspend le déroulement du programme pendant une durée de <argnum> secondes environ.

Exemple: ATTENDS 2 ... interrompt le programme pendant 2 secondes environ.

[Début du catalogue.](#)

### AUCUNE :

Procédure AUCUNE - rend inertes les seize tortues gérées par Logoplus. Cette primitive correspond aux consignes suivantes : CT OBEIS []. Voir PASAPAS , TOUTES .

Exemple: AUCUNE

[Début du catalogue.](#)

### AVANCE :

Procédure AVANCE <argnum> - fait avancer la tortue de <argnum> pas à l'écran. Abréviation: AV . Voir RECULE, TOURNEDROITE, TOURNEGAUCHE, DEGRE , LIEU , LIMITES? , PASAPAS , RADIAN , MODEDEGRE , MODERADIAN .

Exemple: AVANCE 50 fera avancer la tortue de 50 pas à l'écran.

[Début du catalogue.](#)

### AVANCELUTIN :

Procédure AVANCELUTIN <lutin> <arg num> - fait avancer <lutin> de <arg num> pas à l'écran. Voir RECULELUTIN, TOURNELUTINDROITE, TOURNELUTINGAUCHE, ENDORS, REVEILLE, CACHELUTIN, MONTRELUTIN, FIXELUTINX, FIXELUTINY, COLORIELUTIN, COULEURLUTIN?, VITESSE, POSITIONLUTIN?, LUTINX?, LUTINY?, LUTIN VX, LUTIN VY, CAPLUTIN, CAPLUTIN?, VELOCITE?, COLLISIONS? , DEGRE , RADIAN , MODEDEGRE , MODERADIAN .

Exemples: AVANCELUTIN "lutin1 50

fera avancer "lutin1" de 50 pas à l'écran par rapport à sa position actuelle.

AVANCELUTIN [ "lutin1 "lutin2 ] 50

fera avancer "lutin1 et "lutin2 de 50 pas à l'écran par rapport à leur position actuelle respective. "lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### BAISSECRAYON :

Procédure BAISSSECRAYON - autorise la tortue à laisser une trace à l'écran. Voir [LEVECRAYON](#) , [PASAPAS](#) .

Abréviation: [BC](#) .

Exemple: [BAISSECRAYON](#)

[Début du catalogue.](#)

[BLANCHE](#), [VERTE](#), [JAUNE](#), [ROUGE](#), [BLEUE](#), [NOIRE](#), [TILLEUL](#), [EAU](#), [FUCHSIA](#), [SARCELLE](#), [GRISE](#), [OLIVE](#), [MARRON](#), [ARGENT](#), [ORANGE](#), [ROSE](#), [VIOLETTE](#):

Primitives de couleur. Voir [COULEURFOND](#), [COULEURTORTUE](#), [COULEUR?](#) , [COLORIE](#) , [FONDECRAN](#).

Exemple: [COULEURTORTUE](#) BLEUE.

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

### BARRES :

Procédure BARRES [[absc1 absc2 ... absc(n)][ord1 ord2 ... ord(n)][couleur1 couleur2 ... couleur(n)]] ; trace dans une fenêtre "Diagramme" le diagramme-bâtons qui correspond aux données indiquées par les listes-vecteurs <absc> et <ord> avec absc1 < absc2 < ... < absc(n) tous entiers relatifs. Les barres sont ensuite coloriées selon les indications contenues dans la liste vectorielle <couleur>. Si cette liste est vide ou absente, toutes les barres seront coloriées en gris. Deux choses à retenir:

- 1) Un diagramme précédemment tracé, s'il existe, aura d'abord été effacé.
- 2) Comme l'écran de la fenêtre-diagrammes occupe une surface de 513 x 513 pixels, tout tracé qui excède ces dimensions est ajusté numériquement. Voir [HISTO](#) .

Exemple:

[BARRES](#) [[10 20 30][[-7,5](#) 19 54,8][ orange verte bleue]]

mais :

[BARRES](#) [[10 20 30][[-7,5](#) 19 54,8]] voit les barres coloriées en gris-argenté.

[Début du catalogue.](#)

### BOOLEEN :

Indicateur de type **BOOLEEN** - associé à la primitive **DEFINIS**, précise le type de donnée déclarée par **DEFINIS**. Voir également [FAUX](#) , [VRAI](#) , [MOT](#), [LISTE](#) , [NOMBRE](#) , [COULEUR](#) .

Exemple: **DEFINIS** "signal **BOOLEEN** . "signal prend par défaut la valeur **VRAI**.  
On peut aussi préciser si cette valeur est un mot, une liste, un nombre ou une couleur.

### **BOUCLE** :

Fonction **BOUCLE** - associée à la primitive **REPETE**, fournit en résultat l'indice de la boucle en cours de travail. Voir [PASAPAS](#) , [REPETE](#) .

Exemple: **REPETE** 10 [ **ECRISLIGNE** **BOUCLE** ]

[Début du catalogue.](#)

### **CABLEUSB?** :

**K8061/VM140**> Fonction **CABLEUSB?** :carte - vérifie si la carte **K8061/VM140** est bien connectée à un port USB alimenté de l'ordinateur. Voir [CONNEXIONS](#), [CPUOK?](#) , [DECONNECTE](#).

Exemple:

```
CARTEUSB "VM140
DONNE "carte 0
SI CABLEUSB? [
                /* Placer votre code ici. */
                ]
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### **CACHELUTIN** :

Procédure **CACHELUTIN** <lutin> - occasionne la disparition de <lutin> sur l'écran. Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTINXV](#), [LUTINVY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: **CACHELUTIN** "lutin1  
ou:  
**CACHELUTIN** [ "lutin1 "lutin2 ]

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

### CACHESPOTS :

Procédure **CACHESPOTS** - occasionne la fermeture de la fenêtre des capteurs optiques après une activité robotique. Voir [MONTRESPOTS](#) , [SPOTS](#) .

Exemple :

```
CACHESPOTS
```

[Début du catalogue.](#)

### CACHETORTUE :

Procédure CACHETORTUE - permet de rendre la tortue invisible à l'écran. Abréviation: **CT**  
Voir [MONTRETORTUE](#) , [PASAPAS](#) .

Exemple: CACHETORTUE

[Début du catalogue.](#)

### CANAL? :

K8055/VM110> CANAL? :canal - lit l'état d'un canal d'entrée analogique (1 ou 2) et fournit en résultat une valeur comprise entre 0 et 255. (0<= :carte <=7) Voir [FIXECANAL](#) , [CANAUX?](#) .

Exemple:

```
CARTEUSB "VM110  
DONNE "étatcanal CANAL? 1
```

K8061/VM140> CANAL? :carte :canal lit l'état d'un canal d'entrée analogique (De 1 à 8) et fournit en résultat une valeur comprise entre 0 et 255. (0<= :carte <=7). Voir [FIXECANAL](#) , [CANAUX?](#) .

Exemple:

```
CARTEUSB "VM140  
DONNE "carte 0  
DONNE "étatcanal CANAL? :carte 5
```

[Début du catalogue.](#)

### CANAL\_MAX :

K8055/VM110> Procédure CANAL\_MAX :canal - fixe le canal de sortie analogique (1 ou 2) sur le maximum (255). Voir [CANAL\\_MIN](#) , [CANAUX\\_MIN](#) , [CANAUX\\_MAX](#) .

Exemple:

```
CARTEUSB "VM110  
DONNE "carte 0
```

## CANAL\_MAX 1

K8061/VM140> Procédure CANAL\_MAX :carte :canal - fixe le canal de sortie analogique (de 1 à 8) sur le maximum (255). (0<= :carte <=7). Voir [CANAL\\_MIN](#) , [CANAUX\\_MIN](#) ,[CANAUX\\_MAX](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte 0
CANAL_MAX :carte 6
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## CANAL\_MIN :

K8055/VM110> Procédure CANAL\_MIN :canal - fixe le canal de sortie analogique (1 ou 2) sur le minimum (0). Voir [CANAL?](#) , [FIXECANAL](#) , [CANAUX?](#) .

Exemple:

```
CARTEUSB "VM110
CANAL_MIN 1
```

K8061/VM140> Procédure CANAL\_MIN :carte :canal - fixe le canal de sortie analogique (de 1 à 8) sur le minimum (0). (0<= :carte <=7). Voir [CANAL?](#) , [FIXECANAL](#) , [CANAUX?](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte 0
CANAL_MIN :carte 6
```

[Début du catalogue.](#)

## CANAUX? :

K8055/VM110> Fonction CANAUX? - lit l'état des deux canaux d'entrée analogique (1 et 2) et retourne une liste de deux valeurs numériques [ donnée1 donnée2 ] (comprises entre 0 et 255) associées relativement au canal 1 et au canal 2. Voir [FIXECANAL](#) , [CANAL?](#)

Exemple:

```
CARTEUSB "VM110
CANAUX? 1
```

K8061/VM140> Fonction CANAUX? : carte - lit l'état des deux canaux d'entrée analogique et retourne une liste de huit valeurs numériques [ d1 d2 d3 d4 d5 d6 d7 d8 ] (comprises entre 0 et 255) associées relativement aux canaux analogiques de 1 à 8. (0<= :carte <=7). Voir [FIXECANAL](#) , [CANAL?](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte 0
CANAUX? :carte
```

[Début du catalogue.](#)

### CANAUX\_MAX :

K8055/VM110> Procédure CANAUX\_MAX - fixe les canaux de sortie analogiques (1 et 2) sur le maximum (255). Voir [CANAL\\_MIN](#) , [CANAUX\\_MIN](#) , [CANAL\\_MAX](#) .

Exemple:

```
CARTEUSB "VM110  
CANAUX_MAX
```

K8061/VM140> Procédure CANAUX\_MAX :carte - fixe tous les canaux de sortie analogiques (de 1 à 8) sur le maximum (255). (0<= :carte <=7). Voir [CANAL\\_MIN](#) , [CANAUX\\_MIN](#) , [CANAL\\_MAX](#) .

Exemple:

```
CARTEUSB "VM140  
DONNE "carte 0 DONNE "carte 0  
CANAUX_MAX :carte :canal
```

[Début du catalogue.](#)

### CANAUX\_MIN :

K8055/VM110> Procédure CANAUX\_MIN - fixe les canaux de sortie analogiques (1 et 2) sur le minimum (0). Voir [CANAL\\_MAX](#) , [CANAUX\\_MAX](#) .

Exemple:

```
CARTEUSB "VM110  
CANAUX_MIN
```

K8061/VM140> Procédure CANAUX\_MIN :carte - fixe tous les canaux de sorties analogiques (de 1 à 8) sur le minimum (0). (0<= :carte <=7). Voir [CANAL\\_MAX](#) , [CANAUX\\_MAX](#) .

Exemple:

```
CARTEUSB "VM140  
DONNE "carte 0 DONNE "carte 0  
CANAUX_MIN :carte :canal
```

[Début du catalogue.](#)

### CAP :

Fonction CAP - fournit en résultat le cap (son orientation) de la tortue. Voir [FIXECAP](#) , [SENSHORAIRE](#) , [SENSTRIGO](#) , [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [PASAPAS](#) .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### CAPLUTIN :

Procédure **CAPLUTIN** <lutin> <arg num> - fixe le cap (\*) de <lutin> à <arg num> (radians ou degrés).

Abréviation: **CL** . Voir [MODEDEGRE](#), [MODERADIAN](#), [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

(\*) Si l'argument utilisé est supérieur à 360° (ou 2π) , il sera recadré au modulo à 360° (ou 2 π) près.

Exemple: **CAPLUTIN** "lutin1 50

ou:

**CAPLUTIN** [ "lutin1 "lutin2 ] 50 .

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

**CAPLUTIN?** :

Fonction **CAPLUTIN?** <lutin> - retourne le cap (radian ou degré) de <lutin> en résultat . Voir [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: **CAPLUTIN?** "lutin.

"lutin doit cependant être défini dans l'éditeur de lutin.

[Début du catalogue.](#)

**CAPTEURAVANT?** :

Fonction booléenne **CAPTEURAVANT?** - fournit en résultat la valeur booléenne [VRAI](#) si la tortue touche devant elle un objet graphique de couleur différente de celle du fond. Cette fonction a plus d'efficacité lorsque le mode [ACCELERE](#) est désactivé. Voir [CAPTEURDROIT?](#), [CAPTEURGAUCHE?](#) [TRAIT](#) .

Exemple:

```
TESTE CAPTEURAVANT?  
SIVRAI [  
    RECALE 10  
    TOURNEDROITE 90  
]  
SIFAUX AVANCE 10
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### CAPTEURDROIT? :

Fonction booléenne CAPTEURDROIT? - fournit en résultat la valeur booléenne [VRAI](#) si la tortue touche à sa droite un objet graphique de couleur différente de celle du fond. Cette fonction a plus d'efficacité lorsque le mode [ACCELERE](#) est désactivé. Voir [CAPTEURAVANT?](#) , [CAPTEURGAUCHE?](#) [TRAIT](#) .

Exemple:

```
TESTE CAPTEURDROIT?  
SIVRAI TOURNEGAUCHE 90  
SIFAUX AVANCE 10
```

[Début du catalogue.](#)

### CAPTEURGAUCHE? :

Fonction booléenne CAPTEURGAUCHE? - fournit en résultat la valeur booléenne [VRAI](#) si la tortue touche à sa gauche un objet graphique de couleur différente de celle du fond. Cette fonction a plus d'efficacité lorsque le mode [ACCELERE](#) est désactivé. Voir [CAPTEURAVANT?](#) , [CAPTEURDROIT?](#) [TRAIT](#) .

Exemple:

```
TESTE CAPTEURGAUCHE?  
SIVRAI TOURNEDROITE 90  
SIFAUX AVANCE 10
```

[Début du catalogue.](#)

### CARACTERE :

Fonction CARACTERE <argnum> - fournit en résultat le caractère correspondant au code ASCII de <argnum>. Abréviation: [CAR](#) . Voir [ASCII](#) , [PASAPAS](#) .

Exemple: [ECRIS](#) [CARACTERE](#) 65            rend le caractère "A".

[Début du catalogue.](#)

### CARDINAL :

Fonction CARDINAL [arg] - agit aussi bien sur une liste d'items que sur un mot. Si [arg] est une liste, CARDINAL retournera en résultat le nombre d'items de la liste [arg]. Si "arg est un mot,

CARDINAL retournera en résultat le nombre de lettres du mot. Voir [LISTE](#) , [LISTE?](#) , [LISTEVARS](#) , [ITEM](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [REPLACE](#) , [SAISIS](#) , [SAUFDERNIER](#) , [SAUFPREMIER](#) , [TRIECAR](#) , [TRIEITEMS](#) , [✖ || ...](#) . Abréviation: **CARD** .

Exemple: **ECRIS CARDINAL** "logo rend la valeur 4.

[Début du catalogue.](#)

### CARTEPOINTS :

Procédure CARTEPOINTS [ $\langle$ coultrace $\rangle$   $\langle$ coulfond $\rangle$ ][ $\langle$ angle $\rangle$   $\langle$ long $\rangle$   $\langle$ points $\rangle$ ][ $\langle$ x $\rangle$   $\langle$ y $\rangle$ ] - dessine une carte à points centrée sur les coordonnées figurant dans la liste [ $\langle$ x $\rangle$   $\langle$ y $\rangle$ ]. La liste [ $\langle$ angle $\rangle$   $\langle$ long $\rangle$   $\langle$ points $\rangle$ ] contient l'angle (\*) de la carte, ainsi que sa longueur et le nombre de points à dessiner. La largeur de la carte vaut la moitié de sa longueur. La liste [ $\langle$ coultrace $\rangle$   $\langle$ coulfond $\rangle$ ] indique la couleur du tracé et la couleur intérieure de la carte.

(\*) Si l'argument utilisé est supérieur à  $360^\circ$  (ou  $2\pi$ ), il sera recadré au modulo positif à  $360^\circ$  (ou  $2\pi$ ) près.

Exemple: **CARTEPOINTS [ NOIRE EAU ][ 90 35 6 ][ 50 100 ]**

tracera une carte à 6 points centrée sur  $x=50$  et  $y=100$  sous une inclinaison de  $90^\circ$  avec une longueur de 35 pas. Les bords sont noirs et le fond de la carte est bleu clair.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### CARTEUSB :

Procédure CARTEUSB :carte - bascule les primitives LOGO relatives aux cartes USB soit sur la carte K8055 (VM 110), soit sur la carte K8061 (VM140). L'argument :carte est un mot sous l'une des quatre formes suivantes : "K8055", "VM110", "K8061" ou "VM140" pour rendre la primitive effective. Voir [CONNEXIONS](#), [DECONNECTE](#) .

[Début du catalogue.](#)

### CHAMPVECT :

Procédure CHAMPVECT

[[vecteurX\_queue][vecteurY\_queue][vecteurZ\_queue][vecteurX\_tête][vecteurY\_tête]

[vecteurZ\_tête][couleurs]] ; affiche un champ de vecteurs sur l'écran de la fenêtre du grapheur.

L'argument principal de cette procédure est une liste qui contient elle-même quatre sous-listes qui doivent avoir toutes le même cardinal, puisqu'elles représentent les composantes de chaque vecteur sur les trois axes. Voir [INITGRAPH](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [GRAPHEUR](#), [PLOT](#), [PLOTPOL](#), [SEGMENT3D](#) , [SURFACE](#), [SYSLOT](#), [SYSLOTPOL](#) .

Exemple:

INITGRAPH FAUX MODEDEGRE

CHAMPVECT

[[0 (-10) 20 15][ -30 0 50 (-5)][0 0 0 0] [0 (-10) 20 15][ -30 0 50 (-5)][15 (-8) 30 (-40)]  
[ROUGE NOIRE VERTE ORANGE]]

GRAPHEUR VRAI

[Début du catalogue.](#)

CHANGEREP :

Procédure CHANGEREP [chemin] - change le répertoire en cours pour le chemin spécifié par la liste [chemin]. Si [chemin] contient une lettre d'unité, l'unité en cours est également changée. Si [chemin] n'est pas trouvé ou n'existe pas, une boîte de dialogue le signalera. Voir [CREEREP](#) , [SUPPREP](#) , [SAUVE](#) , [SUPPRIME](#) , [RAPPELLE](#) .

Exemple: CHANGEREP [C:\Education\LogoPlus\Travaux d'élèves]

[Début du catalogue.](#)

CHOIX :

Fonction CHOIX :NomVar ; fournit au hasard l'un des items de :NomVar en résultat s'il s'agit d'une liste. Lorsque l'argument :NomVar est un mot ou une couleur nominale, le résultat retourné sera une lettre. Dans les autres cas (nombre ou booléen), le résultat retourné sera un chiffre (dans le cas d'un booléen, VRAI=0 ; FAUX=1). Les délimiteurs de liste et les prédicats de mot (") ne sont pas comptabilisés lors de la sélection d'un item ou d'un caractère.

Voir [DERNIER](#) , [INSERE](#) , [ITEM](#) , [LISTE](#) , [LISLISTE](#) , [LISTEVARS](#) , [MEMBRE?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PHRASE](#) , [PREMIER](#) , [REMPLECE](#) , [SAISIS](#) , [SAUFPREMIER](#) , [SAUFDERNIER](#) , [SEPRE](#) , [TRIECAR](#) , [TRIEITEMS](#) , [\\* \\* | | ...](#) .

Exemple:

EFFTXT

ECL CHOIX [1 2 3 4 5 6] // pourra rendre 5 au choix comme résultat.

ECL CHOIX "abracadabra // pourra rendre "d au choix comme résultat.

ECL CHOIX 1234567890 // pourra rendre 7 au choix comme résultat.

ECL CHOIX \$a10bc // pourra rendre "b au choix comme résultat.

ECL CHOIX VERTE // pourra rendre "R au choix comme résultat.

DONNE "liste LISTEVARS 0 50 51

ECL CHOIX :liste // pourra rendre 28 au choix comme résultat.

DONNE "mot "abracadabrarbadacarba

ECL CHOIX :mot // pourra rendre "a au choix comme résultat.

DONNE "nombre 1 / 27

ECL CHOIX :nombre // pourra rendre 3 au choix comme résultat.

DONNE "couleur ROUGE

ECL CHOIX :couleur // pourra rendre "U au choix comme résultat.

ECRIS CHOIX? [un deux trois] // pourra rendre "deux au choix comme résultat.

[Début du catalogue.](#)

**CHOSE :**

Fonction CHOSE "arg - fournit en résultat la valeur de la variable "arg.

Voir [DONNE / SOIT](#) , [PASAPAS](#) .

Exemple: **DONNE** "MonNombre 5 // éq : **SOIT** "MonNombre 5

**ECRIS CHOSE** "MonNombre            rendra la valeur 5 .

[Début du catalogue.](#)

**CLIC? :**

Fonction booléenne CLIC? - fournit en résultat la valeur booléenne [VRAI](#) si le bouton gauche de la souris est enfoncé, [FAUX](#) sinon.

Exemple: **ECRISLIGNE** CLIC?

rendra **VRAI** si on appuie sur le bouton gauche de la souris au moment de son appel.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**COLLISIONS? :**

Fonction COLLISIONS? - fournit en résultat la liste des couples de lutins qui collisionnent.

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#),

[ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#),

[COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#),

[LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#) , [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) ,

[MODERADIAN](#) .

**Hypothèse:** supposons que votre programme LOGO fasse évaluer six lutins nommés lutin1, lutin2, lutin3, lutin4, lutin5 et lutin6 sur l'écran de la tortue. Ces six lutins doivent avoir bien sûr été définis dans la [grille](#) de définition des lutins. Supposons également que lors du déroulement de votre programme, lutin3 et lutin6 se collisionnent entre eux, lutin2 et lutin5 également, ainsi que lutin1 et lutin4, tout cela au moment où l'utilisateur introduit dans la ligne de commande ou que le programme

rencontre la primitive [COLLISIONS?](#) . Cette primitive créera alors la liste [ [lutin3 lutin 6] [lutin2 lutin5] [lutin1 lutin4] ], qui indique bien dans cet exemple que lutin3 et lutin6 se collisionnent entre eux, lutin2 et lutin5 également ainsi que lutin1 et lutin4.

Exemple:

DONNE "listedeschocs [COLLISIONS?](#)

Rappel : ne jamais oublier de définir les lutins dont votre programme aura besoin dans l'éditeur de lutin.

[Début du catalogue.](#)

COLORIE :

Procédure COLORIE <lieu> <couleur> - colorie avec <couleur> la zone indiquée par <lieu>. <lieu> est un argument de type LIEU - [ :x :y ] - et <couleur> prend alors l'une des valeurs suivantes: VERTE, ROUGE, BLEUE, JAUNE, BLANCHE, NOIRE, FUCHSIA, SARCELLE, TILLEUL, ROSE, ORANGE, MARRON, EAU, GRISE, OLIVE, ARGENT, VIOLETTE. Voir [FONDECRAN](#) , [POINT](#) , [PIXELS?](#) .

Exemple: COLORIE [50 20] JAUNE

(coloriera en jaune la zone fermée autour du point [50 20].)

Voir [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#)

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

COLORIELUTIN :

Procédure COLORIELUTIN <lutin> <couleur> - fixe la couleur de <lutin> à <couleur> . <couleur> peut prendre l'une des valeurs suivantes: VERTE, ROUGE, BLEUE, JAUNE, BLANCHE, NOIRE, FUCHSIA, SARCELLE, TILLEUL, ROSE, ORANGE, MARRON, EAU, GRISE, OLIVE, ARGENT, VIOLETTE. Voir [AVANCELUTIN](#) , [RECULELUTIN](#) , [TOURNELUTINDROITE](#) , [TOURNELUTINGAUCHE](#) , [ENDORS](#) , [REVEILLE](#) , [CACHELUTIN](#) , [MONTRELUTIN](#) , [FIXELUTINX](#) , [FIXELUTINY](#) , [COULEURLUTIN?](#) , [VITESSE](#) , [POSITIONLUTIN?](#) , [LUTINX?](#) , [LUTINY?](#) , [LUTIN VX](#) , [LUTIN VY](#) , [CAPLUTIN](#) , [CAPLUTIN?](#) , [VELOCITE?](#) , [COLLISIONS?](#) .

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

Exemple: [COLORIELUTIN](#) "lutin1 BLEUE

ou:

[COLORIELUTIN](#) [ "lutin1 "lutin2 ] ORANGE.

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### COMPAS :

Fonction COMPAS - ouvre la fenêtre du compas de la tortue. Cela permet de se représenter plus facilement le cap ou l'orientation que va prendre la tortue. La fermeture de la fenêtre renvoie selon le cas l'écriture des primitives Tournedroite, Tournegauche ou Fixecap suivi de la valeur numérique souhaitée. Voir [TOURNEDROITE](#), [TOURNEGAUCHE](#), [FIXECAP](#), [MODEDEGRE](#), [MODERADIAN](#), [SENSHORAIRE](#), [SENSTRIGO](#).

Exemple: `FIXECAP COMPAS`

[Début du catalogue.](#)

### COMPLEXE :

Fonction COMPLEXE <arg\_num\_part\_réelle> <arg\_num\_part\_imag>, où <arg\_num\_part\_réelle> et <arg\_num\_part\_imag> sont des valeurs numériques - construit une liste de deux items [<arg\_num\_part\_réelle> <arg\_num\_part\_imag>] formée à partir des deux arguments numériques de la fonction. Voir [CONJUG](#), [CPLXADD](#), [CPLXPROD](#), [IMAGE](#), [REEL](#).

Exemple:

`SOIT "compl COMPLEXE 5 (-2) affectera à :compl la liste [5 (-2)]`  
qui correspond au nombre imaginaire 5-2i.

Remarque: Dans l'exemple, la primitive COMPLEXE est équivalente au code : `PHRASE 5 (-2)`

[Début du catalogue.](#)

### CONJUG :

Fonction CONJUG <liste>, où <liste> est une liste vectorielle de deux items [<arg\_num\_part\_réelle> <arg\_num\_part\_imag>] - retourne comme résultat le nombre imaginaire conjugué de <liste>, i.e [<arg\_num\_part\_réelle> (-<arg\_num\_part\_imag>)]. Voir [COMPLEXE](#), [CPLXADD](#), [CPLXPROD](#), [IMAGE](#), [REEL](#).

Exemple:

`SOIT "conjugué CONJUG [4,1 7]`

affecte la liste [4,1 -7] à la donnée :conjugué, ce qui correspond au nombre imaginaire 4,1-7i.

Remarque: Dans l'exemple, la primitive **CONJUG** est équivalente au code :

**PHRASE PREMIER [4,1 7] ((DERNIER [4,1 7]) \* (-1))**

[Début du catalogue.](#)

### CONNECTE :

K8055/VM110 > Fonction **CONNECTE** :adresse\_carte - ouvre le lien de communication avec la carte K8055 et retourne **VERI** si elle est présente, **FAUX** dans le cas contraire. :adresse\_carte est une valeur numérique comprise entre 0 et 3 inclus. Voir [DECONNECTE](#).

Exemple: **CONNECTE** 0

[Début du catalogue.](#)

### CONNEXIONS :

K8061/VM140 > Fonction **CONNEXIONS** - ouvre le lien de communication avec la carte K8061/VM140 et retourne la liste des adresses des cartes (0 <= a <= 7), numéros correspondant aux cavaliers (A1, A2, A3) de chaque carte. Voir [DECONNECTE](#).

Exemple:

**DONNE** " adresses **CONNEXIONS** // éq : **SOIT** "adresses **CONNEXIONS**  
**SI NON (VIDE? :adresses) [**

**DONNE** "carte **PREM** :adresses  
/\* Code **LOGO** à effectuer... \*/

**]**

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### CONTENU :

Fonction **CONTENU** - rappelle le contenu de l'éditeur dans l'afficheur des résultats.

Exemple: **ECRISLIGNE** **CONTENU**

[Début du catalogue.](#)

### COPIE :

Procédure **COPIE** <chose1> <chose2> - copie le contenu de <chose2> dans <chose1>. Voir [PASAPAS](#).

Exemple: **COPIE** :a :b

/\* :a et :b doivent auparavant avoir été définis par la primitive DONNE. \*/

[Début du catalogue.](#)

**COSEC** :

Fonction COSEC <arg num> - calcule la valeur de la cosécante de <arg num>.

Voir [ARCCSC](#) , [ARCSEC](#) , [COTAN](#) , [SECANTE](#).

Exemple:

ECRIS COSEC 0,5                      donnera 0,999961923064171 en résultat.

[Début du catalogue.](#)

**COSH** :

Fonction COSH <arg num> - calcule la valeur du cosinus hyperbolique de <arg num>.

Voir [ARGCOSH](#) , [ARGCOth](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS COSH 2                      donnera 1,12762596520638 en résultat.

[Début du catalogue.](#)

**COSINUS** :

Fonction COSINUS <argnum> - calcule la valeur du cosinus du nombre <argnum> (\*). Voir [SINUS](#) , [TAN](#) , [ARCTAN](#) , [ARCCOT](#) , [DEGRE](#) , [INTEGRALE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [ZEROS?](#) . Abréviation: COS

(\* Si l'argument utilisé est supérieur à 360° (ou 2π) , il sera recadré au modulo à 360° (ou 2 π) près.

Exemple: ECRIS COSINUS 30                      rendra 0,8660254.

[Début du catalogue.](#)

**COTAN** :

Fonction COTAN <arg num> - calcule la valeur de la cotangente de <arg num>. <arg num> doit

cependant être différent de 0. Voir [ARCCSC](#) , [ARCSEC](#) , [ARCCSC](#) , [COSEC](#) , [COSINUS](#) , [SECANTE](#) , [SINUS](#) , [TAN](#).

Exemple:

ECRIS COTAN 2                      donnera -0,457657554360286 en résultat.

[Début du catalogue.](#)

**COTH** :

Fonction COTH <arg num> - calcule la valeur de la cotangente hyperbolique de <arg num>.

<arg num> doit cependant être différent de 0. Voir [ARGCOSH](#) , [ARGCOTH](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [COTH](#) , [CSCH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS COTH 2

donnera 1,03731472072755 en résultat.

[Début du catalogue.](#)

COULEUR? :

Fonction COULEUR? <couleur> - fournit en résultat la valeur booléenne [VRAI](#) si <couleur> est identique à l'une des valeurs suivantes: [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#) .

Exemple: ECRISLIGNE COULEUR? TILLEUL rendra [VRAI](#) en résultat.

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

COULEURFOND :

Procédure COULEURFOND <couleur> - change la couleur de l'écran de la tortue. <couleur> peut prendre les valeurs [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#) .

Abréviation: [FCFG](#) . Voir [FONDECRAN](#) , [PASAPAS](#) .

Exemple: COULEURFOND ROUGE

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

COULEURFOND? :

Fonction COULEURFOND? - fournit en résultat la couleur de l'écran de la tortue. La réponse peut alors prendre l'une des valeurs suivantes: [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#) . Abréviation : [CF](#) . Voir [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#) , [PASAPAS](#) .

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

### COULEURLETTRE :

Procédure COULEURLETTRE <couleur> - change la couleur des lettres affichées sur l'écran de la tortue. Elle est fixée à la couleur noire à l'origine d'une session LOGO. Voir [STYLELETTRE](#) , [TAILLELETTRE](#) , [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#)

Exemple: COULEURLETTRE BLEUE

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

### COULEURLUTIN? :

Fonction COULEURLUTIN? <lutin> - fournit en résultat la couleur de <lutin>. Voir [AVANCELUTIN](#) , [RECULELUTIN](#) , [TOURNELUTINDROITE](#) , [TOURNELUTINGAUCHE](#) , [ENDORS](#) , [REVEILLE](#) , [CACHELUTIN](#) , [MONTRELUTIN](#) , [FIXELUTINX](#) , [FIXELUTINY](#) , [COLORIELUTIN](#) , [VITESSE](#) , [POSITIONLUTIN?](#) , [LUTINX?](#) , [LUTINY?](#) , [LUTINXV](#) , [LUTINVY](#) , [CAPLUTIN](#) , [CAPLUTIN?](#) , [VELOCITE?](#) , [COLLISIONS?](#) .

Exemple: COULEURLUTIN? "lutin

ou:

COULEURLUTIN? [ "lutin1 "lutin2 ]

La réponse prend alors l'une des valeurs suivantes: VERTE, ROUGE, BLEUE, JAUNE, BLANCHE, NOIRE, FUCHSIA, SARCELLE, TILLEUL, ROSE, ORANGE, MARRON, EAU, GRISE, OLIVE, ARGENT, VIOLETTE.

Rappel : "lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### COULEURPOINT? :

Fonction COULEURPOINT? <arg>, où <arg> est une liste de type LIEU - fournit en résultat la couleur de l'écran de la tortue correspondant au point de coordonnées <arg>. Voir [LIEU](#).

Exemple: COULEURPOINT? [5 6]

rendra comme résultat l'une des valeurs suivantes:

BLANCHE, VERTE, JAUNE, ROUGE, BLEUE, NOIRE, TILLEUL, EAU, FUCHSIA, SARCELLE, GRISE, OLIVE, MARRON, ARGENT, ORANGE, ROSE, VIOLETTE

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

### COULEURTORTUE :

Procédure COULEURTORTUE <couleur> - change la couleur de la tortue. <couleur> peut prendre les valeurs BLANCHE, VERTE, JAUNE, ROUGE, BLEUE, NOIRE, TILLEUL, EAU, FUCHSIA, SARCELLE, GRISE, OLIVE, MARRON, ARGENT, ORANGE, ROSE, VIOLETTE . Abréviation: FCC . Voir [COULEURFOND](#) , [FONDECRAN](#) , [PASAPAS](#) .

Exemple: COULEURTORTUE BLEUE

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

### COULEURTORTUE? :

Fonction COULEURTORTUE? - fournit en résultat la couleur de la tortue. La réponse peut alors prendre l'une des valeurs suivantes: BLANCHE, VERTE, JAUNE, ROUGE, BLEUE, NOIRE, TILLEUL, EAU, FUCHSIA, SARCELLE, GRISE, OLIVE, MARRON, ARGENT, ORANGE, ROSE, VIOLETTE. Abréviation : CC . Voir [COULEURFOND?](#) , [PASAPAS](#) .

Exemple: ECL COULEURTORTUE?

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### CPLXADD :

Fonction CPLXADD <liste1> <liste2>, où <liste1> et <liste2> sont deux listes vectorielles de deux items chacune de la forme [<arg\_num\_part\_réelle> <arg\_num\_part\_imag>] - retourne en résultat une liste qui correspond à la somme complexe de <liste1> et <liste2>. Voir [COMPLEXE](#) ,

[CONJUG](#) , [CPLXPROD](#) , [IMAGE](#) , [REEL](#) .

Exemple:

**ECL CPLXADD [2 (-1)] [4 3]** verra s'afficher la liste [6 2] ((2+4) et (-1+3))  
qui correspond au nombre imaginaire 6+2i.

[Début du catalogue.](#)

CPLXPROD :

Fonction CPLXPROD <liste1> <liste2>, où <liste1> et <liste2> sont deux listes vectorielles de deux items chacune de la forme [arg\_num\_part\_réelle] [arg\_num\_part\_imag] - retourne en résultat une liste qui correspond au produit complexe de <liste1> et <liste2>. Voir [COMPLEXE](#) , [CONJUG](#) , [CPLXADD](#) , [IMAGE](#) , [REEL](#) .

Exemple:

**ECL CPLXPROD [4 7] [-3 2]** verra s'afficher la liste [16 32] ((12+4) et (24+8))  
qui correspond au nombre imaginaire 16+32i.

[Début du catalogue.](#)

CPUOK? :

[K8061/VM140](#)> Fonction CPUOK? :carte - détecte la présence de la carte K8061/VM140 et rend vrai si la carte est correctement alimentée en énergie. faux sinon. (0<= :carte <= 7). Voir [CONNEXIONS](#) , [DECONNECTE](#) , [FERMECARTE](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte 0
SI CPUOK? :carte [
/* Placer votre code ici. */
]
```

[Début du catalogue.](#)

CREEREP :

Procédure CREEREP [chemin] - crée un sous-répertoire avec le chemin d'accès spécifié par la liste [chemin]. Le dernier élément du chemin ne peut être le nom d'un fichier existant. Si [chemin] ne peut pas être créé, une boîte de dialogue apparaîtra. Voir [CHANGEREP](#) , [SUPPREP](#) , [RAPPELLE](#) , [SAUVE](#) , [SUPPRIME](#)

Exemple: **CREEREP [C:\Education\LogoPlus\Mes programmes]**

[Début du catalogue.](#)

CSCH :

Fonction CSCH <arg num> - calcule la valeur de la cosécante hyperbolique de <arg num>.

Voir [ARGCOSH](#) , [ARGCOth](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [SECH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS CSCH 2

donnera 0,275720564771783 en résultat.

[Début du catalogue.](#)

DATE :

Fonction DATE - fournit en résultat la liste contenant le nom du jour, sa date, le mois et l'année en cours. Voir [TEMPS](#) , [PASAPAS](#) .

Exemple:

ECRISLIGNE DATE

affichera [ mercredi 7 juillet 2004 ]

[Début du catalogue.](#)

DEC :

Procédure de décrémentation DEC :nomvar ou DEC [:nomvar argnum ] - retranche 1 à :nomvar dans le premier cas (ég : [DONNE](#) "nomvar :nomvar - 1) et la valeur numérique argnum dans le second cas (ég : [DONNE](#) "nomvar :nomvar - argnum) .

:nomvar et argnum peuvent aussi bien être entiers que décimaux. La variable à décrémentation ainsi que la variable de décrémentation doivent cependant avoir été définies. Voir [INC](#) , [PASAPAS](#) .

Exemple: [SOIT](#) "MonNombre 8 // ég: [DONNE](#) "MonNombre 8  
[DEC](#) :Monnombre // :Monnombre <- :Monnombre -1  
[DEC](#) [:Monnombre 5] // :Monnombre <- :Monnombre -5  
[DONNE](#) "delta 2 // ég: [SOIT](#) "delta 2  
[DEC](#) [:Monnombre :delta] // :Monnombre <- :Monnombre -2

La donnée :Monnombre contient finalement la valeur 0.

[Début du catalogue.](#)

DECONNECTE :

Procédure DECONNECTE - ferme le lien avec la carte K8055. Elle s'effectue automatiquement à la fin d'un programme LOGO si elle n'y figure pas. Voir [CONNECTE](#) , [CONNEXIONS](#) , [FERMECARTE](#) .

[Début du catalogue.](#)

## DEFINIS :

Procédure DEFINIS "NomVar <type> - définit le type d'une variable. <type> peut alors prendre les valeurs [NOMBRE](#), [MOT](#), [LISTE](#), [VRAI](#), [FAUX](#), [COULEUR](#). Cette primitive est utile pour supprimer certains avertissements du compilateur lors de l'évaluation du type d'une expression. Voir [DONNE / SOIT](#), [PASAPAS](#). Abréviation: DEF

Exemple: DEFINIS "MaVariable NOMBRE définira :MaVariable en tant que valeur numérique.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## DEGRE :

Fonction DEGRE <argnum> - fournit en résultat la valeur en degré d'un nombre exprimé en radians. Voir [RADIAN](#), [SENSHORAIRE](#), [SENSTRIGO](#), [MODEDEGRE](#), [MODERADIAN](#).

Exemple: ECRIS DEGRE 3,141592654 rendra la valeur 180 comme résultat.

[Début du catalogue.](#)

## DERNIER :

Fonction DERNIER [arg] - fournit en résultat le dernier item de la liste [arg]. Si un mot est passé à cette fonction en argument, une liste vide sera retournée en résultat.

Voir [LISTE](#), [MEMBRE?](#), [INSERE](#), [ITEM](#), [LISTEVARS](#), [OCCURRENCES](#), [PASAPAS](#), [PERMUTATIONS](#), [CHOIX](#), [PREMIER](#), [REMPLECE](#), [SAISIS](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [\\*\\* || ...](#).

Exemple: ECRIS DERNIER [un deux trois] rendra trois comme résultat.

Abréviation: DER

[Début du catalogue.](#)

## DERNIERELETTRE :

Fonction DERNIERELETTRE "arg - fournit en résultat le dernier caractère du mot "arg. Voir [MOT](#), [DERNIERELETTRE](#), [PASAPAS](#), [PERMUTATIONS](#), [PREMIERELETTRE](#), [SAUFPREMIERELETTRE](#), [SAUFDERNIERELETTRE](#), [SEPRE](#), [CHOIX](#), [\\*\\* || ...](#). Abrév: DERL

Exemple: ECRIS DERNIERELETTRE "tables rendra "s comme résultat.

## [Début du catalogue.](#)

### DEROULE :

Procédure [DEROULE](#) - permet à la tortue de sortir de la zone visible de l'écran. Il faudra alors manœuvrer les curseurs situés dans la fenêtre de travail pour observer le graphique tracé par la tortue. Voir [ENROULE](#) , [PASAPAS](#) , [XCOORD](#) , [YCOORD](#) .

## [Début du catalogue.](#)

### DESSINETEXTE :

Procédure [DESSINETEXTE](#) <situation> <phrase> - écrit <phrase> sur l'écran de la tortue au point de coordonnées indiqué par les deux premiers items de la liste <situation> avec la couleur actuelle de la tortue. <situation> est une liste de type [SITUATION](#) ([:x :y :angle]) et <phrase> peut être soit une liste, soit un mot, soit un nombre. Le troisième item de la liste <situation> (:angle) indique l'inclinaison du texte sur l'écran de la tortue. Utilisez auparavant [STYLELETTRE](#) , [COLORIELETTRE](#) ou [TAILLELETTRE](#) pour que cet item soit actif. Voir aussi [FONDECRAN](#) .

Exemple: [STYLELETTRE](#) "Mistral" [DESSINETEXTE](#) [50 20 270] [ Bonjour, les enfants ! ]

aura pour résultat l'écriture horizontale de :

Bonjour, les enfants ! à partir du point (50, 20).

## [Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### DETERMINANT :

Fonction [DETERMINANT](#) <liste\_matrice> - où <liste\_matrice> est une liste matricielle carrée ; calcule le déterminant de <liste\_matrice> . Rappel: les sous-listes de <liste> représentent les vecteurs-colonnes. Un argument incorrect transmis à [DETERMINANT](#) entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [V\\_PROPRES](#), [V\\_UNIQUE](#), [PRODEXT](#), [PRODINT](#)

Exemple:

[ECL](#) [DETERMINANT](#) [[2 (-2) 3 ][2 1 5][ -1 4 (-3)]]

produira -21 comme résultat.

## [Début du catalogue.](#)

### DETRUIS :

Fonction DETRUIS <liste> ou DETRUIS <mot> - fournit en résultat le contenu de <liste> ou de <mot> sans leurs attributs de typage. Voir [PASAPAS](#) .

Exemples: **DETRUIS** [:x]      donnera :x en résultat  
**DETRUIS** ":y      rendra :y en résultat.

## [Début du catalogue.](#)

### DIFFERENCE :

Fonction DIFFERENCE <arg num> où <arg num> est une liste de valeurs ou de variables numériques - fournit le résultat de la différence généralisée des termes inclus dans <arg num>. Abréviation: DIFF . Voir [ [+ - \\* /](#) ], [DEC](#) , [INC](#) , [LISTEVARS](#) , [MAX?](#) , [MIN?](#) , [PASAPAS](#) , [PRODUIT](#) , [QUOTIENT](#) , [RESTE \(%\)](#) , [SOMME](#) , [TRIEITEMS](#) .

#### Exemple:

**DONNE** "v -4 **ECRIS DIFFERENCE** [ 1 5 :v 3 ]

rendra -3 comme résultat.

( 1 - 5 - (-4) -3)

## [Début du catalogue.](#)

### DIFFERENTS? :

Fonction booléenne DIFFERENTS? <arg1> <arg2> - fournit la valeur booléenne [VRAI](#) en résultat si les arguments <arg1> et <arg2> sont différents, [FAUX](#) sinon. <arg1> et <arg2> peuvent aussi bien être ensemble des arguments numériques que des listes ou des mots. Abréviation: DIFF? . Voir [EGALITE?](#) , [PRODUIT](#) , [QUOTIENT](#) , [RESTE \(%\)](#) , [SOMME](#) .

Exemple: **ECRIS DIFFERENTS?** 5 2      rendra **VRAI** comme résultat.

## [Début du catalogue.](#)

## [Vers l'index des primitives >>](#)

### DIV :

Opérateur infixé <arg1> DIV <arg2> - calcule la partie entière de la division <arg1> / <arg2>. Voir [MAX?](#) , [MIN?](#) , [MOD \(%\)](#) , [QUOTIENT](#) , [SOMME](#) .

Exemple: **ECRIS** 10 **DIV** 4      rendra 2 comme résultat.

## [Début du catalogue.](#)

## DIVISEURS :

Fonction DIVISEURS <arg num> ; construit la liste des diviseurs de <arg num>.  
Voir [MOD](#), [%](#), [PERMUTATIONS](#), [QUOTIENT](#), [SOMME](#) .

Exemple:

ECL DIVISEURS 12

rendra [1 2 3 4 6 12] comme résultat (qui est une liste vectorielle).

[Début du catalogue.](#)

## DONNE / SOIT :

Instruction d'affectation DONNE "nomvar <arg> ou DONNE "nomvar" <arg> - attribue à la variable nomvar la valeur <arg>. <arg> peut aussi bien être un mot, une liste ou une valeur numérique.

Remarque: on peut omettre la deuxième partie des guillemets dans la déclaration du nom de la variable. Equivalent à SOIT . Voir [LISTEVARS](#) , [PASAPAS](#) .

Exemples: DONNE "MonNombre" 3 // éq : SOIT "MonNombre" 3

ou:

DONNE "MonNombre 3 (on peut omettre la seconde partie des [guillemets](#).)  
// éq : SOIT "MonNombre 3

et :MonNombre prend la valeur 3 dans les deux cas.

[Début du catalogue.](#)

## e, E :

Caractère e ou E (dans les nombres): placé dans un nombre entier ou réel, signifie " est multiplié par 10 à la puissance ". Voir [EXP](#), [LN](#) , [LOG](#), [NOMBRE](#) , [PUISSANCE](#) , [RACINE2](#) , [RACINE3](#) .

Exemple 1: ECRIS 2\*7,5e2 , ECRIS 2\*7,5E2 ou ECRIS 2\*7,5E+2 donneront le même résultat.

Exemple 2: ECRIS 4\*3,14e-5 et:  
ECRIS 4\*3,14E-5 donneront également un résultat identique.

[Début du catalogue.](#)

\_ :

Séparateur entre la partie entière et la partie décimale d'un nombre. Pour être le plus proche possible de l'écriture normale d'un nombre, l'usage de la virgule a été conservé. Exemple: DONNE "nombre 123,456. Voir [E](#), [e](#)

[Début du catalogue.](#)

## ECART\_TYPE :

Fonction ECART\_TYPE <liste\_vecteur> - où <liste\_vecteur> est une liste de nombres à une seule dimension (liste vectorielle) ; calcule l'écart-type suivant les valeurs de <liste> qui ne doit contenir que des nombres (liste vectorielle). Un argument incorrect transmis à ECART\_TYPE entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [DIVISEURS](#) , [MOD](#) , [%](#) , [PERMUTATIONS](#) , [QUOTIENT](#) , [SOMME](#)

Exemple:

```
ECL ECART_TYPE [1,5 (-27) 3,8 42 (-16) 2]
```

rendra 23,5077646746772 comme résultat.

[Début du catalogue.](#)

## ECRIS :

Procédure ECRIS <arg> ou ECRIS :nomarg - écrit la valeur <arg> ou le contenu de la variable :nomarg. Abréviation: EC . Voir [ECRISLIGNE](#) , [PASAPAS](#) , [xx || ...](#) .

Exemple: ECRIS "Bonjour! réalisera l'écriture du mot Bonjour! dans l'afficheur.

[Début du catalogue.](#)

## ECRISLIGNE :

Procédure ECRISLIGNE <arg> ou ECRISLIGNE :nomarg - écrit la valeur <arg> ou le contenu de la variable :nomarg en forçant auparavant le passage à la ligne. Abréviation: ECL . Voir [ECRIS](#) , [PASAPAS](#) , [xx || ...](#) .

Exemple: ECRIS "Bonjour ECRISLIGNE [ à tous ! ]

réalisera l'écriture du mot Bonjour et à tous! sur deux lignes séparées dans l'afficheur.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## EFFACEECRAN :

Procédure EFFACEECRAN - réalise l'effacement complet de l'écran de la tortue et réinitialise les paramètres physiques (position en (x,y) et cap) de toutes les tortues. Voir [LIEU](#) , [LIMITES?](#) , [NETTOIE](#) , [PASAPAS](#) .

Abréviation: EFF

[Début du catalogue.](#)

### EFFACELIGNE:

Procédure **EFFACELIGNE** - réalise l'effacement complet de la ligne des commandes.

Abréviation: **EFFL**

[Début du catalogue.](#)

### EFFACETEXTE :

Procédure **EFFACETEXTE** - réalise l'effacement complet de l'afficheur des résultats. Voir [PASAPAS](#) . Abréviation: **EFFTXT**

[Début du catalogue.](#)

### EGALITE? :

Fonction booléenne **EGALITE?** <arg1> <arg2> - fournit la valeur booléenne [VRAI](#) en résultat si <arg1> est égal à <arg2> quelque soit leur type, [FAUX](#) sinon. Voir [=](#) , [DEC](#) , [DIFFERENTS?](#) , [INC](#) .

Exemple: **ECRIS EGALITE?** 5 5      donnera **VRAI** comme résultat.

[Début du catalogue.](#)

### ELLIPSE :

Procédure **ELLIPSE** [ Xe Ye Ra Rp angle ] ; trace une ellipse dont les paramètres figurent dans la liste passée en argument. Les deux premiers items (Xe et Ye) correspondent aux coordonnées cartésiennes du centre de l'ellipse, Ra et Rp, aux valeurs des rayons d'apogée Ra (demi-grand axe) et de périégée Rp (demi-petit axe) respectivement avec l'excentricité  $e=Ra/Rp$ . Le dernier item de la liste, angle, détermine l'inclinaison de l'ellipse. Voir [ARC](#) , [FONDECRAN](#) , [SEGMENT](#) .

Exemple:

**ELLIPSE** [0 0 20 50 45] tracera une ellipse d'un demi-grand axe de 50 pixels et d'un demi-petit axe de 20 pixels de rayon inclinée de 45° sur l'origine de l'écran.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### ENDORS :

Procédure **ENDORS** <lutin> - occasionne l'arrêt de <lutin> sur l'écran. <lutin> reste toutefois visible.

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTINXV](#), [LUTINVY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemples: **ENDORS** "lutin1

ou:

**ENDORS** [ "lutin1 "lutin2 ]

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

**ENROULE** :

Procédure **ENROULE** - interdit à la tortue de sortir de la zone visible de son écran. Elle rentrera de ce fait par l'autre côté de cette zone. Voir [DEROULE](#) , [PASAPAS](#) , [XCOORD](#) , [YCOORD](#) .

Abréviation: **ENR**

[Début du catalogue.](#)

**ENTIER** :

Fonction **ENTIER** <argnum> - fournit la partie entière de <argnum> en résultat.

Voir [ARRONDIR](#), [DEC](#) , [INC](#) , [MAX?](#), [MIN?](#) , [PASAPAS](#) .

Exemple: **ECRIS ENTIER PI** donnera 3 en résultat.

[Début du catalogue.](#)

**ET** :

Connecteur logique **ET** - <arg1> **ET** <arg2> - fournit en résultat les valeurs booléennes suivantes selon les valeurs respectives de <arg1> et <arg2> (également booléennes) - [VRAI](#) **ET** [VRAI](#) -> [VRAI](#), [FAUX](#) **ET** [VRAI](#) -> [FAUX](#), [VRAI](#) **ET** [FAUX](#) -> [FAUX](#), [FAUX](#) **ET** [FAUX](#) -> [FAUX](#). Voir [OU\(\)](#) , [LESDEUX?](#) , [UNDESDEUX?](#) . Alias : **&**

Exemple: **DONNE** "x 3 // éq : **SOIT** "x 3  
**TESTE** ((:x > 0) **ET** (:x < 5)) rendra **VRAI** en résultat.  
// éq : **TESTE** ((:x > 0) **&** (:x < 5))

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**EXECDLL** :

Fonction EXECDLL [chemin\_dll fonction type\_fonc] [type\_param1 val\_param1 type\_param2 val\_param2 type\_param3 val\_param3] - fournit le résultat de la fonction incluse dans la dll. Les deux listes doivent être décrites de manière explicite, c'est à dire, ne pas être représentées par un nom de variable. Par contre, les valeurs de paramètre peut être invoquées nominalement et faire l'objet de calculs. Cette liste de paramètres peut bien sûr être vide ([]) et elle ne peut contenir au plus que trois paramètres. Seuls les types du langage C: boolean, double, float, int, longint, char, void peuvent être invoqués dans la liste des paramètres. Par sécurité, toute erreur dans la structure des deux listes entraînera un arrêt immédiat du programme LOGO en cours, de même, qu'un chemin d'accès incorrect sur la dll. Voir aussi [OUVREDLL](#), [FERMEDLL](#).

### Exemples:

```
DONNE "résultat EXECDLL [[ c:/Program files/Education/LogoPlus/bibdyn.dll] stdsad double]
[longint 3 double 2,5]
```

ou:

```
DONNE "n 3 DONNE "résultat EXECDLL [bibdyn.dll stdsad double] [longint :n double 2,5]
(remarquer la virgule dans 2,5).
```

[Début du catalogue.](#)

### EXP :

Fonction EXP <arg num> - calcule le résultat de la fonction exponentielle élevée à la puissance de <arg num>. Voir [INTEGRALE](#), [LN](#), [LOG](#), [PUISSANCE](#), [RACINE2](#), [RACINE3](#), [ZEROS?](#).

### Exemple:

```
ECRIS EXP 3 rendra 20,0855369231877 comme résultat.
```

```
ECRIS EXP (-3) rendra 0,049780683678639 comme résultat.
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### FAIRE :

Primitive FAIRE <prim> ou FAIRE [ prim1 prim2 ...] - réalise l'action des primitives qui la suivent. Remplace la primitive EXECUTE des version LOGO anglo-saxonnes.

Exemple: FAIRE carré 5 (On suppose que la procédure carré est déjà définie dans l'éditeur)

[Début du catalogue.](#)

## **FAUX** :

Valeur booléenne **FAUX** (comparaison lors d'un test). A utiliser de préférence avec les fonctions booléennes [DIFFERENTS?](#), [EGALITE?](#), [LISTE?](#), [NEGATIVE?](#), [NOMBRE?](#), [PROCEDURE?](#), [LESDEUX?](#), [UNDESDEUX?](#), [VIDE?](#), [VISIBLE?](#), [ZERO?](#). Voir [BOOLEEN](#), [VRAI](#).

n.b : Les valeurs booléennes **VRAI** et **FAUX** peuvent être utilisées telles quelles dans une expression conditionnelle, c'est à dire sans avoir recours aux signes de comparaison.

Exemple:

```
SI VRAI ECL [ Cette consigne sera toujours exécutée. ]
```

```
SI FAUX ECL [ Cette consigne ne sera jamais exécutée. ]
```

[Début du catalogue.](#)

## **FERMECANAL** :

K8055/VM110> Procédure FERMECANAL :canal - éteint le canal de sortie numérique. (:canal compris entre 1 et 8). Voir [CONNECTE](#), [FERMECANAUX](#), [FIXECANAUX\\_L](#), [OUVRECANAL](#).

Exemple:

```
CARTEUSB "VM110
```

```
FERMECANAL 1
```

K8061/VM140> Procédure FERMECANAL :carte :canal éteint le canal de sortie numérique. (:canal compris entre 1 et 8). (0<= :carte <=7). Voir [CONNEXIONS](#), [FERMECANAUX](#), [FERMECARTE](#), [FIXECANAUX\\_L](#), [OUVRECANAL](#).

Exemple:

```
CARTEUSB "VM140
```

```
DONNE "carte CONNEXIONS
```

```
FERMECANAL :carte 1
```

[Début du catalogue.](#)

## **FERMECANAUX** :

K8055/VM110> Procédure FERMECANAUX - éteint tous les canaux de sortie numériques. Voir [CONNECTE](#), [FIXECANAUX\\_L](#), [FERMECANAL](#).

Exemple:

```
CARTEUSB "VM110
```

```
FERMECANAUX
```

K8061/VM140> Procédure FERMECANAUX :carte éteint tous les canaux de sortie numériques. (0<= :carte <=7). Voir [CONNEXIONS](#), [FERMECARTE](#), [FIXECANAUX\\_L](#), [FERMECANAL](#).

Exemple:

```
CARTEUSB "VM140
```

**DONNE** "carte **CONNEXIONS**

**FERMECANAUX** :carte /\* On éteint tous les canaux de sortie logique. \*/

[Début du catalogue.](#)

## **FERMECARTE** :

**K8061/VM140**> Procédure **FERMECARTE** :carte - déconnecte la carte K8061 de rang :carte (0 <= :carte <=7). Voir [CONNEXIONS](#) , [FERMECARTE](#) , [FIXECANAL\\_L](#) , [FIXECANAUX\\_L](#) , [FERMECANAUX](#) , [OUVRECANAL](#) .

Exemple: **FERMECARTE** 0 .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## **FERMEDLL** :

Fonction **FERMEDLL** <liste chemin dll> - ferme et libère la place en mémoire occupée par la bibliothèque dynamique et retourne VRAI ou FAUX selon le déroulement de la fermeture. Voir [OUVREDLL](#), [EXECDLL](#) .

Exemple: **FERMEDLL** [ c:/Program files/Education/LogoPlus/bibdyn.dll ]

[Début du catalogue.](#)

## **FIN** :

Primitive **FIN**: déclenche l'arrêt de la procédure en cours (mais pas obligatoirement du programme LOGO). Il doit toujours être associé au mot POUR. Ces mots-clés ne doivent jamais apparaître dans la ligne de commande (un message d'erreur apparaîtra alors). Voir [POUR](#) , [REPONDRE](#) , [RETOURNE](#) , [STOP](#) .

Exemple:

**POUR** tracer

**EFFACECRAN** carré 50 **COLORIE** [ 5 10 ] ROSE

**ATTENDS** 2 **LEVECRAYON** **AVANCE** 70 **ATTENDS** 2

**BAISSECRAYON** rectangle 80 30 **COLORIE** [ 5 130 ] JAUNE

**FIN**

[Début du catalogue.](#)

## **FIXECANAL** :

K8055/VM110> Procédure **FIXECANAL** :canal :donnée - ajuste le canal de sortie analogique (1 ou 2) selon la donnée indiquée (comprise entre 0 et 255). Voir [CONNECTE](#) , [CANAL?](#) , [CANAUX?](#) .

Exemple:

```
CARTEUSB "VM110
CONNECTE 0
FIXECANAL 1 255
```

K8061/VM140> Procédure **FIXECANAL** :carte :canal :donnée - ajuste le canal de sortie analogique (de 1 à 8) selon la donnée indiquée (comprise entre 0 et 255). (0<= :carte <=7). Voir [CONNEXIONS](#) , [CANAL?](#) , [CANAUX?](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte CONNEXIONS
DONNE "canal_sortie 1 /* On allume d'abord le canal analogique n°1. */
REPETE 8 [
    REPETE 255 [
        FIXECANAL :carte :canal_sortie BOUCLE
        ATTENDS 0,002 // on attend 2 millisecondes.
    ]
    INC "canal_sortie
]
```

[Début du catalogue.](#)

**FIXECANAL\_L :**

K8061/VM140> Fonction **FIXECANAL\_L** :carte - fournit une valeur comprise entre 0 et 255 sur le port digital de la carte K8061 de rang :carte (0 <= :carte <=7). Voir [CONNEXIONS](#) , [FIXECANAUX\\_L](#) , [FERMECANAL](#) , [FERMECANAUX](#) , [FERMECARTE](#) , [OUVRECANAL](#) , [OUVRECANAUUX](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte 0
DONNE "valeurportL FIXECANAL_L :carte
```

[Début du catalogue.](#)

**FIXECANAUX\_A :**

K8055/VM110> Procédure **FIXECANAUX\_A** :a1 :a2 - ajuste les deux canaux de sortie analogiques (1 et 2) selon les données. Voir [CONNECTE](#) , [CANAL?](#) , [FIXECANAL](#) , [CANAUX?](#) , [CANAL\\_MIN](#) , [CANAL\\_MAX](#) , [CANAUX\\_MIN](#) , [CANAUX\\_MAX](#) .

Exemple:

```
CARTEUSB "VM110
CONNECTE 0
```

## FIXECANAUX\_A 128 255

K8061/VM140> Procédure FIXECANAUX\_A :carte :listecanaux - ajuste tous les canaux de sorties analogiques (de 1 à 8) selon les valeurs entre 0 et 255) contenues dans :liste avec :carte (0<= :carte <=7). Voir [CONNEXIONS](#) , [CANAL?](#) , [FIXECANAL](#) , [CANAUX?](#) , [CANAL\\_MIN](#) , [CANAL\\_MAX](#) , [CANAUX\\_MIN](#) , [CANAUX\\_MAX](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte CONNEXIONS
FIXECANAUX_A :carte 128
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## FIXECANAUX\_L :

K8055/VM110> Procédure FIXECANAUX\_L :valeur - ajuste les sorties numériques selon la valeur. (valeur comprise entre 0 (00000000) et 255 (11111111)). Voir [CONNECTE](#) , [OUVRECANAL](#) , [OUVRECANAL](#) , [FERMECANAL](#) , [FERMECANAL](#) , [OUVERT?](#) , [OUVERTS?](#) .

Exemple:

```
carteusb "VM110 // Rappel : Logoplus comprend également les primitives en minuscules.
connecte 0
soit "sél translate 8 (-2)
fixecanaux_l :sél
```

K8061/VM140> Procédure FIXECANAUX\_L :carte :valeur - ajuste les sorties numériques selon la valeur. (valeur comprise entre 0 (00000000) et 255 (11111111)). (0<= :carte <=7). Voir [CONNEXIONS](#) , [OUVRECANAL](#) , [OUVRECANAL](#) , [FERMECANAL](#) , [FERMECANAL](#) , [FERMECANAL](#) , [FERMECANAL](#) , [OUVERT?](#) , [OUVERTS?](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte CONNEXIONS
FIXECANAUX_L :carte 128
```

[Début du catalogue.](#)

## FIXECAP :

Procédure FIXECAP <argnum> - donne au cap (\*) (orientation) de la tortue une valeur de <argnum> exprimée en degrés. Voir [CAP](#) , [COMPAS](#) , [DEGRE](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [NETTOIE](#) , [PASAPAS](#) , [RADIAN](#) , [SENSHORAIRE](#) , [SENSTRIGO](#) .

(\*) Si l'argument utilisé est supérieur à 360° (ou 2 $\pi$ ) , il sera recadré au modulo positif à 360° (ou 2  $\pi$ ) près.

Exemple: [FIXECAP](#) 90

...et la tortue pointe son nez vers la droite de l'écran en mode SENSORAIRE.

[Début du catalogue.](#)

**FIXEHASARD** :

Procédure [FIXEHASARD](#) <borne1> <borne2> - où <borne1> et <borne2> sont des arguments numériques ou des variables représentant des valeurs numériques. [FIXEHASARD](#) détermine l'intervalle des nombres générés par la fonction [HASARD](#) . Voir [HASARD](#) , [PASAPAS](#) , [RANDDEC](#) , [RANDOM](#) .

Exemple: [FIXEHASARD](#) 10 20

réalise la génération d'un nombre compris entre 10 et 20 inclus lors du prochain appel de [HASARD](#).

[Début du catalogue.](#)

**FIXELUTINX** :

Procédure [FIXELUTINX](#) <lutin> <arg num> - déplace <lutin> à l'abscisse <arg num> sur l'écran. Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINXY](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTINXV](#), [LUTINVY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) , [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#) .

Exemple: [FIXELUTINX](#) "lutin1 50

ou:

[FIXELUTINX](#) [ "lutin1 "lutin2 ] 50

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**FIXELUTINXY** :

Procédure [FIXELUTINXY](#) <lutin> <arg numX> <arg numY> - déplace <lutin> à l'abscisse <arg numX> et à l'ordonnée <arg numY> sur l'écran. Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#),

[LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#), [MODERADIAN](#).

Exemple: **FIXELUTINXY** "lutin1 50 10

ou:

**FIXELUTINXY** [ "lutin1 "lutin2 ] 50 10

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

### FIXELUTINY :

Procédure **FIXELUTINY** <lutin> <arg num> - déplace <lutin> à l'ordonnée <arg num> sur l'écran.

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINXY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#), [MODERADIAN](#).

Exemple: **FIXELUTINY** "lutin1 50

ou:

**FIXELUTINY** [ "lutin1 "lutin2 ] 50

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### FIXEMODULATION :

[K8061/VM140](#)> Fonction **FIXEMODULATION** :carte :a - ajuste la sortie PWM de la carte (0 <= :carte <=7) au pourcentage de la valeur :a (0 <= :a <= 100). Abréviation : **FIXEPWM**. Voir [CONNEXIONS](#), [DECONNECTE](#), [MODULATION?](#)

Exemple:

**CARTEUSB** "VM140

**DONNE** "carte **CONNEXIONS**

**DONNE** "largeurimpulsion **FIXEMODULATION** :carte 25

// fixe à 25% la valeur de la largeur de la modulation sur le port PWM.

[Début du catalogue.](#)

### FIXEREBOND :

Procédure **FIXEREBOND** :compteur :t - ajuste le temps d'élimination du rebondissement selon le compteur d'impulsions, avec :compteur = 1 ou 2 et :t compris entre 0 et 5000 millisecondes. Voir [INIT\\_COMPTEUR](#) , [TOPS?](#) .

Exemple:

```
CARTEUSB "VM110
CONNECTE 0
FIXEREBOND 1 2500
```

[Début du catalogue.](#)

**FIXEX:**

Procédure **FIXEX** <argnum> - fixe à la coordonnée horizontale (abscisse) de la tortue une valeur de <argnum> exprimée en pas sur l'écran. Voir [FIXEY](#) , [FIXEXY](#) , [LIEU](#) , [LIMITES?](#) , [NETTOIE](#) , [PASAPAS](#) .

Exemple: **FIXEX** 50

[Début du catalogue.](#)

**FIXEXY :**

Procédure **FIXEXY** <arg numX> <arg numY> - fixe aux coordonnées de la tortue une valeur de <arg numX> en abscisse et une valeur de <arg numY> en ordonnée, valeurs exprimées en pas sur l'écran. **FIXEXY** est équivalente à **LEVECRAYON VA** [<arg numX> <arg numY>] sauf qu'elle ne modifie pas son cap. Voir [FIXEX](#) , [FIXEY](#) , [LIEU](#) , [LIMITES?](#) , [NETTOIE](#) , [PASAPAS](#) .

Exemple: **FIXEXY** 50 20

[Début du catalogue.](#)

**FIXEY :**

Procédure **FIXEY** <argnum> - fixe à la coordonnée verticale (ordonnée) de la tortue une valeur de <argnum> exprimée en pas sur l'écran. Voir [FIXEX](#) , [FIXEXY](#) , [LIEU](#) , [LIMITES?](#) , [NETTOIE](#) , [PASAPAS](#) .

Exemple: **FIXEY** 50

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**FONDECRAN :**

Primitive **FONDECRAN** [ chemin ] où <chemin> est l'emplacement physique d'un fichier image. Si

<chemin> est valide, FONDECRAN fera apparaître l'image voulue en arrière-fond sur l'écran de la tortue. L'image doit avoir été enregistrée dans l'un des trois formats suivants : bitmap (.bmp) , Jpeg (.jpg) ou curseur (.curs). Un chemin non valable rendra cette primitive inopérante ou effacera l'image d'arrière-fond sélectionnée auparavant dans le cours du programme en cours d'exécution. L'image sélectionnée dans le programme sera automatiquement effacée à la fin de son exécution ou laissera à nouveau apparaître l'image d'arrière-plan sélectionnée par l'option "Importer un fond d'écran." du menu flottant de la fenêtre de travail. Voir [ARC](#) , [ELLIPSE](#) , [GRAPHE](#) , [GRAPH2D](#) , [GRAPH3D](#) , [INITGRAPH](#) , [SEGMENT](#) , [SURFACE](#) , [ZMAX](#) , [ZMAX?](#) , [ZPOINT](#) , [ZPOINT?](#) , [ZSEGMENT](#) .

#### Exemple:

FONDECRAN [ C:\Program Files\Education\Logoplus\Défis en images\GdDéfi n°3.JPG ]  
fera apparaître l'image correspondante.

Pour effacer cette image, on pourra donner n'importe quel argument.

Ainsi:

FONDECRAN 0 ou FONDECRAN " ou encore FONDECRAN []

[Début du catalogue.](#)

#### FORME :

Primitive [FORME](#) <arg num> ou <arg num> est un nombre compris entre 1 et 3. Elle fixe l'aspect de (des) la tortue(s). 1 correspond à la silhouette d'une tortue, 2 au triangle isocèle et 3 au vecteur. Voir [OBEIS](#) , [PASAPAS](#) , [TORTUES?](#)

#### Exemple :

[FORME](#) 2

[Début du catalogue.](#)

#### FVECT :

Fonction FVECT [nom\_fonction <listevars1> <listevars2> ... <listevarsn> ] où <listevars1> <listevars2> ... <listevarsn> contiennent le même nombre <I> d'items numériques ; calcule la liste des valeurs prises par la fonction <nom\_fonction> lorsque chacun des n paramètres prend successivement pour valeur les <I> valeurs dans sa liste de variables respective. La fonction invoquée ne doit pas être récursive. Voir [INTEGRALE](#) , [ZEROS?](#) .

#### Exemple :

[EFFTXT](#)

DONNE "i1 [0 1 2]

DONNE "i2 [3 \$4 5]

DONNE "i3 [6 7 8]

DONNE "i4 [ fxyz [0 1 2] :i2 [6 7 8]]

```
DONNE "k FVECT :i4
ECL :k
DONNE "k FVECT [fxyz [0 1 2][3 4 5] [6 7 8]]
```

```
POUR fxyz :x :y :z
RENDS :x + :y + :z
FIN
```

Affectera à la variable k la liste (ici on peut même dire vecteur) [9 12 15].

[Début du catalogue.](#)

## GRAPHE :

Fonction GRAPHE <coordonnées> <listes\_rangs> <paramètres> ; calcule la longueur totale d'un graphe et en réalise son tracé. <coordonnées> est une liste matricielle des coordonnées rectangulaires des noeuds du graphe en 2D ou en 3D et <listes\_rangs>, la liste des listes des rangs des noeuds à connecter contenues dans <coordonnées>. Chaque nouvelle liste de rangs représente donc une nouvelle série de noeuds à connecter (le graphe défini n'est pas obligatoirement connexe).

<paramètres> est une liste de cinq termes : <orienté> <affiche\_noeuds> <affiche\_rang\_noeuds> <couleur> et <nom réel des noeuds> , qui définissent les propriétés du graphe (<orienté> = VRAI = 1), l'affichage des noeuds sous forme de points (<affiche\_noeuds> = VRAI = 1), l'affichage du nom des noeuds (<affiche\_nom> = VRAI = 1), la couleur du graphe <couleur> (nombre ou nom de couleur) et la liste des noms réels des noeuds du graphe. Les deux derniers paramètres de cette liste sont facultatifs: en cas d'ommission, la couleur noire est sélectionnée et le rang des points à relier fait office de nom des noeuds. Voir [CHAMPVECT](#) , [GRAPH3D](#) , [INITGRAPH](#), [PLOT](#), [PLOTPOL](#), [SEGMENT3D](#), [SURFACE](#) , [SYSPLOT](#), [SYSPLOTPOL](#), [ZPOINT](#), [ZSEGMENT](#) .

## Exemple:

```
INITGRAPH FAUX
```

```
ECRISLIGNE GRAPHE [[-50 (-50) (-50)][-50 50 (-50)][50 (-50) (-50)][50 50 (-50)][-50 (-50) 50]
[-50 50 50][50 (-50) 50][50 50 50]] [[1 2 3 4 1 5 ][ 7 8 5 6 ][ 1 5 6 2 ]] [FAUX VRAI VRAI NOIRE
[Sommet1 Sommet2 Sommet3 Sommet4 Sommet5 Sommet6 Sommet7 Sommet8]]
```

```
// ou
```

```
GRAPHE [[-50 (-50) (-50)][-50 50 (-50)][50 (-50) (-50)][50 50 (-50)][-50 (-50) 50] [-50 50
50][50 (-50) 50][50 50 50] ] [[1 2 3 4 1 5 ][ 7 8 5 6 ][ 1 5 6 2 ]] [FAUX VRAI VRAI NOIRE
[Sommet1 Sommet2 Sommet3 Sommet4 Sommet5 Sommet6 Sommet7 Sommet8]]
```

```
GRAPHEUR VRAI
```

[Début du catalogue.](#)

## GRAPH2D :

Fonction `GRAPH2D` [ binf bsup npoints profondeur nom\_fonction discret ] - affiche le graphique de la fonction numérique désignée par `nom_fonction` dans l'intervalle [binf bsup] et retourne éventuellement la longueur de la conique tracée (même si le paramètre discret est fixé à **VRAI**). Le bloc `nom_fonction` doit être défini dans l'éditeur et ne doit comporter qu'un seul argument dans sa définition. Le paramètre booléen `discret` indique si la fonction sera tracée comme une succession de points ou de manière continue. Il est possible d'afficher le tracé de plusieurs fonctions tant que `INIGRAPH` n'a pas été invoqué. Bien que le tracé d'une fonction à une seule variable se situe sur un plan, la fenêtre du grapheur donne la possibilité de faire pivoter ce tracé selon tous les angles de vue au moyen de la souris. Voir [CHAMPVECT](#), [GRAPH3D](#), [INITGRAPH](#), [PLOT](#), [PLOTPOL](#), [SEGMENT3D](#), [SURFACE](#), [SYSLOT](#), [SYSLOTPOL](#), [ZPOINT](#), [ZSEGMENT](#).

$$Y = f(x) \quad x \in \{ \text{binf}, \text{binf} + \delta, \text{binf} + 2\delta, \text{binf} + 3\delta, \dots, \text{bsup} \}$$

$$Y = \{ f(\text{binf}), f(\text{binf} + \delta), f(\text{binf} + 2\delta), f(\text{binf} + 3\delta), \dots, f(\text{bsup}) \}$$

$$\delta = \frac{\text{bsup} - \text{binf}}{\text{npoints}}$$

Exemple:

```
INITGRAPH FAUX MODERADIAN
GRAPHEUR FAUX
GRAPH2D [ -10 0 500 0 fx1 VRAI ] // ou ECL GRAPH2D [ -10 0 500 0 fx1 VRAI ]
GRAPHEUR VRAI

POUR fx1 : x
REND 10 * SIN :x
FIN
```

[Début du catalogue.](#)

GRAPH3D :

Fonction `GRAPH3D` [ binf bsup npoints nom\_fonction1 nom\_fonction2 discret ] - affiche le graphique des fonctions numériques désignées par `nom_fonction1` `nom_fonction2` dans l'intervalle [binf bsup] et retourne éventuellement la longueur de la conique tracée (même si le paramètre discret est fixé à **VRAI**). Les blocs `nom_fonction1` et `nom_fonction2` doivent être définis dans l'éditeur et ne doivent comporter qu'un seul argument dans leur définition. Le paramètre booléen `discret` indique si les fonctions seront tracées comme une succession de points ou de manière continue. Il est possible d'afficher le tracé de plusieurs fonctions tant que `INIGRAPH` n'a pas été invoqué. Bien que le tracé d'une fonction paramétrique puisse se situer sur un plan, la fenêtre du grapheur donne la possibilité de faire pivoter ce tracé selon tous les angles de vue au moyen de la

souris. Cette primitive a été conçue pour tracer facilement les fonctions paramétriques. Voir [CHAMPVECT](#), [GRAPHE](#), [GRAPH2D](#), [INITGRAPH](#), [PLOT](#), [PLOTPOL](#), [SEGMENT3D](#), [SURFACE](#), [SYSLOT](#), [SYSLOTPOL](#), [ZPOINT](#), [ZSEGMENT](#).

$$\begin{cases} X = x(m) \\ Y = y(m) \end{cases} \quad m \in \{ \text{binf}, \text{binf} + \delta, \text{binf} + 2\delta, \text{binf} + 3\delta, \dots, \text{bsup} \}$$

$$X = \{x(\text{binf}), x(\text{binf} + \delta), x(\text{binf} + 2\delta), x(\text{binf} + 3\delta), \dots, x(\text{bsup})\}$$

$$Y = \{y(\text{binf}), y(\text{binf} + \delta), y(\text{binf} + 2\delta), y(\text{binf} + 3\delta), \dots, y(\text{bsup})\}$$

$$\delta = \frac{\text{bsup} - \text{binf}}{\text{npoints}}$$

Exemple:

```
INITGRAPH FAUX MODERADIAN
```

```
GRAPHEUR FAUX
```

```
GRAPH3D [ 5 15 500 fx fy VRAI ] // ou DONNE "longueur GRAPH3D [ 5 15 500 0 fx fy VRAI ]
```

```
GRAPHEUR VRAI
```

```
POUR fx : x
```

```
REND 10 * SIN :x
```

```
FIN
```

```
POUR fy : y
```

```
REND 10 * COS :y
```

```
FIN
```

[Début du catalogue.](#)

GRAPHEUR :

Procédure GRAPHEUR <argbool> - ouvre si nécessaire la fenêtre du grapheur lorsque <argbool> est [VRAI](#), qu'elle soit vide de tracé ou non. Si <argbool> est [FAUX](#), la fenêtre du grapheur sera au besoin refermée. On peut remplacer <argbool> par 0 (FAUX) ou 1 (VRAI). Voir [CHAMPVECT](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [INITGRAPH](#), [PLOT](#), [PLOTPOL](#), [SEGMENT3D](#), [SURFACE](#), [SYSLOT](#), [SYSLOTPOL](#).

Exemple 1:

```
GRAPHEUR VRAI /* = GRAPHEUR 1 */
```

```
GRAPHEUR FAUX /* = GRAPHEUR 0 */
```

## Exemple 2:

```
INITGRAPH FAUX MODERADIAN
GRAPHEUR FAUX
GRAPH2D [ -10 0 500 0 fx1 VRAI ]
GRAPHEUR VRAI
```

```
POUR fx1 : x
REND 10 * SIN :x
FIN
```

[Vers l'index des primitives >>](#)

[Début du catalogue.](#)

## HASARD :

Fonction HASARD - fournit initialement en résultat une valeur numérique entière comprise entre 0 et 9 inclus. Les bornes de l'ensemble des choix peuvent cependant être modifiées avec la primitive FIXEHASARD. Voir [FIXEHASARD](#) , [PASAPAS](#) , [RANDDEC](#) , [RANDOM](#) .

## Exemple:

```
FIXEHASARD 1 6
ECRIS HASARD // éq: ECRIS RANDOM 1 6
// effectue le tirage aléatoire d'un nombre compris entre 1 et 6.
```

[Début du catalogue.](#)

## HEXA? :

Fonction HEXA <arg> - fournit en résultat la valeur booléenne VRAI si <arg> correspond à une quantité numérique hexadécimale. FAUX sinon. Voir [TRANSLATE](#) .

Exemple: ECRIS HEXA? \$A5F rendra VRAI comme résultat.

[Début du catalogue.](#)

## HISTO :

- 3) Procédure HISTO [[borne1 borne2 ... borne(n)]] [[absc1 absc2 ... absc(n)]] [[ord1 ord2 ... ord(n)]] [[couleur1 couleur2... couleur(n-1)]] <arg booléen> ; trace dans la fenêtre "Diagramme" l'histogramme qui correspond aux données indiquées par les listes-vecteurs. Les items des listes <bornes> et <absc> doivent cependant être rangées par ordre croissant. Les barres sont ensuite coloriées selon les indications contenues dans la liste vectorielle <couleurs>. Si cette liste est vide ou absente, les barres seront coloriées en gris. La liste vectorielle [borne] doit avoir un item de plus que la liste vectorielle [couleur] si elle est indiquée et les listes

vectérielles [absc] et [ord] doivent avoir le même nombre d'items. <arg booléen> précise si les points (abscx ordx) doivent être reliés (VRAI/1) ou pas (FAUX/0). Deux choses à retenir :

- 1) Un diagramme précédemment tracé, s'il existe, aura d'abord été effacé.
- 2) Comme l'écran de la fenêtre-diagrammes occupe une surface de 513 x 513 pixels, tout tracé qui excède ces dimensions est ajusté numériquement. Voir [BARRES](#) .

Exemple:

HISTO [[-8 10 15 20 40 55][[-7,5 13 16 19 41 43,4 50,8]][24,3 (-18,7) 34]] [ rouge marron rose verte jaune]]

mais :

HISTO [[-8 10 15 20 40 55][[-7,5 13 16 19 41 43,4 50,8]][24,3 (-18,7) 34]]]

verra les colonnes de l'histogramme coloriées en gris-argenté.

[Début du catalogue.](#)

IMAGE :

Fonction IMAGE <liste>, où <liste> est une liste vectorielle de deux items [<arg\_num\_part\_réelle> <arg\_num\_part\_imag>] - retourne comme résultat le nombre réel <arg\_num\_part\_imag>. Voir [COMPLEXE](#) , [CONJUG](#) , [CPLXADD](#) , [CPLXPROD](#) , [REEL](#) .

Exemple:

ECL IMAGE [1 (-5)] (qui correspond au nombre complexe 1-5i)  
verra s'afficher le nombre -5.

Remarque: Dans l'exemple, la primitive IMAGE est équivalente au code : DERNIER [1 (-5)]

[Début du catalogue.](#)

IMPRIMEEcran :

Procédure IMPRIMEEcran - réalise l'impression de l'écran de la tortue. Voir [IMPRIMETEXTE](#) .

Abréviation: IM .

Exemple: IMPRIMEEcran

[Début du catalogue.](#)

IMPRIMETEXTE :

Procédure IMPRIMETEXTE - réalise l'impression du contenu de l'afficheur. Abréviation: [IMTXT](#) .  
Voir [IMPRIMEECRAN](#) .

Exemple: [IMPRIMETEXTE](#)

[Début du catalogue.](#)

**INC** :

Procédure d'incrémentation INC :nomvar ou INC [:nomvar argnum ] - ajoute 1 à :nomvar dans le premier cas (ég : [DONNE](#) "nomvar :nomvar + 1) et la valeur numérique argnum dans le second cas (ég : [DONNE](#) "nomvar :nomvar + argnum) .

:nomvar et argnum peuvent aussi bien être entiers que décimaux. La variable à incrémenter ainsi que la variable d'incrémentation doivent cependant avoir été définies. Voir [DEC](#) , [MAX?](#) , [MIN?](#) , [PASAPAS](#) .

Exemple: [SOIT](#) "MonNombre 3 // ég: [DONNE](#) "MonNombre 3  
[INC](#) :Monnombre // :Monnombre <- :Monnombre +1  
[INC](#) [:Monnombre 5] // :Monnombre <- :Monnombre +5  
[DONNE](#) "delta 2 // ég: [SOIT](#) "delta 2  
[INC](#) [:Monnombre :delta] // :Monnombre <- :Monnombre +2

La donnée :Monnombre contient finalement la valeur 11.

[Début du catalogue.](#)

**INIT\_COMPTEUR** :

Procédure INIT\_COMPTEUR :compteur - réinitialise le compteur d'impulsions 16 bit (:compteur = 1 ou 2) à zéro. Voir [FIXEREBOND](#) , [TOPS?](#) .

Exemple:

```
CARTEUSB "VM110
INIT_COMPTEUR 1
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**INITGRAPH** :

Procédure INITGRAPH <arg bool> - initialise la liste des coordonnées graphiques du grapheur de fonction et ouvre la fenêtre d'affichage du tracé si <arg bool> = VRAI. Dans le cas où <arg bool> est FAUX, cette fenêtre sera refermée. Voir [CHAMPVECT](#) , [GRAPHE](#) , [GRAPH2D](#) , [GRAPH3D](#) , [GRAPHEUR](#) , [PLOT](#) , [PLOTPOL](#) , [SEGMENT3D](#) , [SURFACE](#) , [SYSPLOT](#) , [SYSPLOTPOL](#) .

### Exemple:

```
INITGRAPH FAUX MODERADIAN
GRAPHEUR FAUX
GRAPH2D [ -10 0 500 0 fx1 VRAI ]
GRAPHEUR VRAI
```

```
POUR fx1 : x
REND 10 * SIN :x
FIN
```

[Début du catalogue.](#)

### INSERE :

Procédure INSERE [liste] [item] [pos] - insère l'argument [item] à la position indiquée dans la liste [pos]. [pos] ne doit contenir que des arguments numériques directement utilisables ( nombres ou variables), sans quoi LOGO déclenchera une erreur en exécution. La liste [pos] ne sera calculée que si elle figure dans l'appel de la procédure. Si [pos] indique une position nulle ou l'unité, [item] sera placé au début de la liste (ou de la sous-liste). Si [pos] indique une position plus grande que le cardinal de [liste], [item] sera placé à la fin de la liste (ou de la sous-liste). Cette primitive remplace avantageusement les primitives MPREMIER/MDERNIER des anciennes versions LOGO puisqu'elle fonctionne de la même manière dans une sous-liste et pour n'importe quelle position.

Voir [DERNIER](#) , [ITEM](#) , [LISTE](#) , [LISTEVARS](#) , [MAX?](#) , [MIN?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PREMIER](#) , [REPLACE](#) , [SAISIS](#) , [TRIEITEMS](#) .

Exemple: `ECL INSERE [ un deux [ 1 2 3 ] quatre ] [un] [3 1]`

insérera [un] au début de la sous-liste [1 2 3]  
pour obtenir la liste: [ un deux [ [un] 1 2 3 ] quatre ].

`INSERE [ a b c d e ] "f [6]`                      affichera [ a b c d e f ]

et:

`ECL INSERE [ b c d e f ] "a [1]`                      affichera [a b c d e f].

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### INTEGRALE :

Fonction INTEGRALE [ binf bsup nintervalles nom\_fonction ] - calcule la valeur de l'intégrale de la fonction désignée par nom\_fonction dans l'intervalle [binf bsup]. nom\_fonction ne doit comporter qu'un seul argument et nintervalles doit être supérieur ou égal à 1. La méthode d'intégration utilisée est celle de Simpson. Voir [ABS](#) , [ARCCOS](#) , [ARCSIN](#) , [ARCTAN](#) , [COSINUS](#) , [DEGRE](#) , [EXP](#) , [FVECT](#) , [HEXA?](#) , [LN](#) , [LOG](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [NON](#) , [OPPOSE](#) , [RACINE2](#) , [RACINE3](#) , [RADIAN](#) , [SENSHORAIRE](#) , [SENSTRIGO](#) , [SINUS](#) , [TAN](#) , [TRANSLATE](#) , [ZEROS?](#) .

Exemple:

```
MODEDEGRE EFFTXT
ECL INTEGRALE [ 1 3,14159 5000 fx ]
```

```
POUR fx : x
RENDS 1 / :x
FIN
```

[Début du catalogue.](#)

ITEM :

Fonction ITEM <arg> <argnum> - fournit en résultat le ou les <argnum-ième(s)> item(s) de la liste (ou du mot) <arg>. Dans le cas où <arg> est une liste, cette fonction ne travaille que sur les éléments situés sur le premier niveau de cette liste, contrairement à la primitive SAISIS et doivent être immédiatement utilisables (soit sous forme de nombres, soit sous forme de variables)  
. Voir [DERNIER](#) , [INSERE](#) , [LISTE](#) , [LISTEVARS](#) , [MEMBRE?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PREMIER](#) , [REMPLECE](#) , [SAISIS](#) , [SAUFPREMIER](#) , [SAUFDERNIER](#) , [TRIECAR](#) , [TRIEITEMS](#) , [xx](#) [||](#) [\\*\\*\\*\\*](#) , [ /e ou /E ].

Exemples:

```
ECRIS ITEM [un deux trois quatre] 3   rendra le mot "trois comme résultat.
ECRIS ITEM [un [deux trois] quatre] 2   rendra la liste [deux trois] en résultat.
ECRIS ITEM [un deux trois quatre] [2 4 1]   rendra la liste [ deux quatre un] en résultat.
ECRIS ITEM [un deux trois quatre cinq six] [1 [3 5 6] 2 ]   rendra la liste [un [trois cinq six] deux ]
                                                                comme résultat.
ECRIS ITEM "abcdefghijklmnop [12 8 16]   rendra le mot "lhp en résultat.
ECRIS ITEM "abcdefghijklmnop [3 [ 5 7 ] 9 12 15]   rendra la liste [c [e g] i l o] comme résultat.
```

[Début du catalogue.](#)

JOUE :

Procédure JOUE [arg] ou JOUE "arg - déclenche l'émission de la mélodie codée dans [arg] ou "arg. L'argument peut être soit un mot, soit une liste. Cet argument contient pour chaque note en premier son nom: La (A), Si (B), Do(C), Ré(D), Mi(E), Fa(F), Sol(G) suivi ou non d'une altération (# ou b) puis de son octave (de 1 à 6) et finalement de sa durée: quadruple-croche(QC), double-croche (DC), Triolet (T), croche(C), croche-pointée (CP), noire(N), noire-pointée(NP), blanche(B), blanche-pointée(BP) et ronde(R). La note Ré dièse noire-pointée située à la 3ième octave sera codée par Ré#3NP ou D#3NP. Un quart de soupir sera noté -5, un demi-soupir sera codé -4, un soupir: -3, une demi-pause: -2 et une pause: -1 . Voir [TEMPO](#) .

Exemples:

JOUE "Do3CDo3CDoCRé3CMi3NRé3NDo3CMi3CRé3CRé3CDo3N

ou:

JOUE [ Do3C Do3C DoC Ré3C Mi3N Ré3N Do3C Mi3C Ré3C Ré3C DO3N ]

ou:

JOUE "C3CC3CC3CD3CE3ND3NC3CE3CD3CD3CC3N

ou:

JOUE [ C3C C3C C3C D3C E3N D3N C3C E3C D3C D3C C3N ]

font entendre la même mélodie.

[Début du catalogue.](#)

LESDEUX? :

Fonction logique LESDEUX? - LESDEUX? <arg1> <arg2> - fournit en résultat les valeurs booléennes suivantes selon les valeurs respectives de <arg1> et <arg2> (également booléennes) - LESDEUX? [VRAI](#) VRAI -> VRAI, LESDEUX? [FAUX](#) VRAI ->FAUX, LESDEUX? VRAI FAUX -> FAUX, LESDEUX? FAUX FAUX -> FAUX. La fonction LESDEUX? a la même table de vérité que le connecteur ET. Voir [OU\(!\)](#) , [ET\(&\)](#) , [UNDESDEUX?](#) .

Exemple: [DONNE](#) "x 3 // éq : [SOIT](#) "x 3

[ECL](#) [LESDEUX?](#) (:x > 0) (:x < 5) rendra VRAI en résultat.

[Début du catalogue.](#)

LEVECRAYON :

Procédure LEVECRAYON - n'autorise pas la tortue à laisser une trace à l'écran. Voir [BAISSECRAYON](#) , [PASAPAS](#) . Abréviation: LC

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

LIEU :

Fonction [LIEU](#) - fournit en résultat la liste des coordonnées actuelles [[XCOORD](#) [YCOORD](#)] de la tortue. Voir [LIEU](#), [SITUATION](#), [FIXEX](#), [FIXEXY](#), [FIXEY](#), [FIXECAP](#), [LIMITES?](#), [NETTOIE](#), [PASAPAS](#), [RAMENE](#), [VA](#), [VERS](#).

Exemple:

```
DONNE "loc LIEU
AV 50 TD 45 AV 50
FIXEXY PREM :loc DER :loc
```

[Début du catalogue.](#)

**LIMITES?** :

Fonction [LIMITES?](#) - fournit en résultat la liste des valeurs absolues des bornes de l'écran de la tortue. Voir [AVANCE](#), [FIXEX](#), [FIXEY](#), [FIXEXY](#), [LIEU](#), [NETTOIE](#), [PASAPAS](#), [RAMENE](#), [RECOULE](#), [SITUATION](#).

Exemple: [ECRIS LIMITES?](#)

affichera [ 269 300 ] pour les dimensions en abscisse  $\pm 269$  et en ordonnée  $\pm 300$  de l'écran.

[Début du catalogue.](#)

**LISCAR** :

Fonction [LISCAR](#) - suspend le déroulement du programme tant qu'un caractère n'aura pas été saisi au clavier et retourne ce caractère en résultat précédé du symbole d'un mot (") quand l'utilisateur ne l'a pas placée. Voir [LISLISTE](#), [LISNOMBRE](#), [TRIECAR](#), [TRIEITEMS](#).

Exemple: [DONNE](#) "carac [LISCAR](#)

[Début du catalogue.](#)

**LISLISTE** :

Fonction [LISLISTE](#) - suspend le déroulement du programme tant qu'une phrase n'aura pas été saisie au clavier dans l'éditeur après avoir tapé sur la touche Entrée et fournit en résultat cette phrase sous forme de liste entre crochets ([]) lorsque cette liste comporte plusieurs items ou sous forme d'un mot (") s'il n'y a qu'un seul item. Voir [LISCAR](#), [LISNOMBRE](#), [TRIEITEMS](#), [\\*\\* || ""](#), [[/e ou /E](#)].

Abréviation: LL

Exemple: [DONNE](#) "nom [LISLISTE](#)

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### **LISMAT :**

Fonction LISMAT - suspend le déroulement du programme et ouvre la fenêtre d'édition de matrice puis attend que cette fenêtre soit refermée avec une matrice valide. Voir [AFFICHE](#), [DETERMINANT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [V\\_PROPRES](#), [V\\_UNIQUE](#), [PRODEXT](#), [PRODINT](#)

Exemple: `SOIT "matrice LISMAT`

[Début du catalogue.](#)

### **LISNOMBRE :**

Fonction LISNOMBRE - suspend le déroulement du programme tant qu'une chaîne de caractères correspondant à un nombre n'aura pas été saisie au clavier dans l'éditeur après avoir tapé sur la touche Entrée.

Exemple: `DONNE "valeur LISNOMBRE`

Voir [LISCAR](#) , [LISLISTE](#) .

[Début du catalogue.](#)

### **LISPORT:**

Fonction LISPORT <argnum> (fonction de lecture du port série (RS 232)) - selon la valeur de <argnum> comprise entre 1 et 4, fournit l'état de CTS (<argnum>=1), DSR (<argnum>=2), RI (<argnum>=3) ou DCD (<argnum>=4) du port correspondant à OUVREPORT . Voir [OUVREPORT](#), [PORT](#), [PORTS?](#)

Exemple: `OUVREPORT 1 SOIT "dsr LISPORT 2`

affectera à :dsr l'état de DSR du port ouvert.

[Début du catalogue.](#)

### **LISTE :**

Fonction LISTE :liste1 :liste2 - réalise la fusion des deux listes. Voir [LISTE?](#), [MEMBRE?](#) , [PHRASE](#) , [INSERE](#) , [ITEM](#), [CHOIX](#) , [DERNIER](#), [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#), [REPLACE](#) , [SAISIS](#) , [SAUFPREMIER](#), [SAUFDERNIER](#) , [TRIEITEMS](#) , `xx || ...` , [ /e ou /E ] .

Exemple: LISTE [marguerite] [violette pétunia]

construit la liste [marguerite [ violette pétunia ]].

[ violette pétunia ] est le second item de la liste formée.

Attention :

Ne pas confondre :

SOIT "maliste LISTE [ item1] [item2] qui construira la liste [[item1][item2]]

et :

SOIT "maphrase PHRASE [ item1] [item2] qui construira la liste [item1 item2]

Remarque : Lorsque l'un des arguments :liste1 et :liste2 (ou les deux en même temps) n'est (ne sont) pas de type liste, il(s) est (sont) transformé(s) en liste soit au moment où LOGO apprend le texte, soit au cours du déroulement du programme.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

LISTE? :

Fonction LISTE? :NomVar - fournit la valeur booléenne [VRAI](#) en résultat si la variable NomVar correspond à une liste. [FAUX](#) sinon. Voir [CHOIX](#) , [DERNIER](#) , [LISTE](#) , [ITEM](#) , [MEMBRE](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#) , [REPLACE](#) , [SAISIS](#) , [SAUFPREMIER](#) , [SAUFDERNIER](#) , [TRIEITEMS](#).

Exemple: ECRIS LISTE? [un deux trois] rendra [VRAI](#) comme résultat.

[Début du catalogue.](#)

LISTEVARS :

Fonction LISTEVARS binf bsup nvaleurs - fournit la liste des nvaleurs comprises entre binf et bsup (bornes comprises). Voir [CHAMPVECT](#) , [CHOIX](#) , [DERNIER](#) , [FVECT](#) , [ITEM](#) , [INSERE](#) , [LISLISTE](#) , [MEMBRE?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PHRASE](#) , [PREMIER](#) , [REPLACE](#) , [SAUFDERNIER](#) , [SAUFPREMIER](#) , [SEPRE](#) , [TRIEITEMS](#) .

Exemple:

SOIT "maliste LISTEVARS 2 12 6

affectera [ 2 4 6 8 10 12 ] à :maliste.

[Début du catalogue.](#)

LN :

Fonction LN <argnum> - calcule la valeur du Log Népérien d'un nombre <argnum>. <argnum> doit être strictement plus grand que 0. Voir [INTEGRALE](#), [LOG](#), [PUISSANCE](#), [RACINE2](#), [RACINE3](#), [ZEROS?](#).

Exemple: ECRIS LN 2,78182818      rendra 1 comme résultat.

[Début du catalogue.](#)

LOG :

Fonction LOG <argnum> - calcule la valeur du Log décimal d'un nombre <argnum>. <argnum> doit être strictement plus grand que 0. Voir [INTEGRALE](#), [LN](#), [PUISSANCE](#), [RACINE2](#), [RACINE3](#), [ZEROS?](#).

Exemple: ECRIS LOG 10      rendra 1 comme résultat.

[Début du catalogue.](#)

LUTINVERS :

Procédure LUTINVERS <lutin> [arg] - dirige <lutin> vers le lieu indiqué par [arg] sur l'écran, [arg] doit être une liste de type LIEU. "lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin. Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTINVY](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#).

Exemple:

LUTINVERS "lutin1 [ 10 -30]

ou:

LUTINVERS [ "lutin1 "lutin2 ] [10 -30]

LUTIN VX :

Procédure LUTIN VX <lutin> <arg num> - fixe l'abscisse de <lutin> à <arg num> sur l'écran. Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTINVY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#).

Exemple: LUTIN VX "lutin1 50

ou:

LUTIN VX [ "lutin1 "lutin2 ] 50

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**LUTIN VY :**

Procédure LUTIN VY <lutin> <arg num> - fixe l'ordonnée de <lutin> à <arg num> sur l'écran.

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTIN VX](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: LUTIN VY "lutin1 50

ou:

LUTIN VY [ "lutin1 "lutin2 ] 50

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

**LUTIN X?** :

Fonction LUTIN X? <lutin> - fournit l'abscisse de <lutin> en résultat .

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINY?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: LUTIN X? "lutin

"lutin doit cependant être défini dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**LUTIN Y?** :

Fonction LUTINY? <lutin> - fournit l'ordonnée de <lutin> en résultat . Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: **LUTINY?** "lutin

"lutin doit cependant être défini dans l'éditeur de lutin.

[Début du catalogue.](#)

**MAX?** :

Fonction MAX? <arg1> <arg2> - fournit deux types de résultat selon que <arg2> soit vrai (=1) ou faux (=0). <arg1> peut être soit une liste, soit un nombre, soit un mot. Si <arg2> est vrai (=1), le résultat sera la valeur numérique maximale contenue dans <arg1>. Si <arg2> est faux (=0), le résultat rendu sera la position de la valeur maximale dans <arg1>. Cette position peut aussi bien être représentée par un nombre que par une liste de nombres si cette valeur maximale se situe dans une sous-liste de <arg1>. Les items non numériques de <arg1> ne sont pas pris en compte.

Voir [FVECT](#) , [DERNIER](#) , [INSERE](#), [LISTEVARS](#) , [MIN?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#) , [REMPLECE](#), [SAUFDERNIER](#) , [SAUFPREMIER](#) , [SAISIS](#) , [TRIEITEMS](#) .

Exemple: **ECRIS MAX?** [1 2 4 [3 5 6 ] 0 ] **VRAI** rendra 6 en résultat, mais

**ECRIS MAX?** [1 2 4 [3 5 6 ] 0 ] **FAUX** rendra [4 3] comme résultat.

[Début du catalogue.](#)

**MEMBRE?** :

Fonction MEMBRE? [sous liste] [Liste principale] ou MEMBRE? "monmot [Liste principale]: retourne la valeur booléenne [VRAI](#) en résultat si la liste [sous liste] ou le mot "monmot est contenu dans [Liste principale]. [FAUX](#) sinon. Voir [DERNIER](#), [LISTE](#) , [LISTEVARS](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#), [SAUFPREMIER](#), [SAUFDERNIER](#) , [TRIECAR](#) , [TRIEITEMS](#) , [\\*\\* ||](#) ,  
.....  
\_\_\_\_\_.

n.b. : MEMBRE? prend en compte la valeur des données nominales contenues dans les listes avant de les analyser.

Exemple 1: **ECRIS MEMBRE?** [un] [un deux trois] rendra VRAI comme résultat,

**ECRIS MEMBRE?** "deux [un deux trois] rendra VRAI également.

Exemple 2: **SOIT** "x 5 **SOIT** "y 6  
**SOIT** "sousliste [ 4 5 :y ]  
**ECL MEMBRE?** [ 4 :x 6 ] [1 2 3 :sousliste 7 8 ] rendra VRAI comme résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### MIN? :

Fonction **MIN?** <arg1> <arg2> - fournit deux types de résultat selon que <arg2> soit vrai (=1) ou faux (=0). <arg1> peut être soit une liste, soit un nombre, soit un mot. Si <arg2> est vrai (=1), le résultat sera la valeur numérique minimale contenue dans <arg1>. Si <arg2> est faux (=0), le résultat rendu sera la position de la valeur minimale dans <arg1>. Cette position peut aussi bien être représentée par un nombre que par une liste de nombres si cette valeur minimale se situe dans une sous-liste de <arg1>. Les items non numériques de <arg1> ne sont pas pris en compte.

Voir [FVECT](#) , [DERNIER](#) , [INSERE](#) , [LISTEVARS](#) , [MAX?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#) , [REPLACE](#) , [SAUFDERNIER](#) , [SAUFPREMIER](#) , [SAISIS](#) , [TRIEITEMS](#).

Exemple: **ECRIS MIN?** [1 \$A 4 [3 5 0 ] 6 ] VRAI rendra 0 en résultat, mais  
**ECRIS MIN?** [1 \$A 4 [3 5 0 ] 6 ] FAUX rendra [4 3] comme résultat.

[Début du catalogue.](#)

### MOD :

Opérateur infixé <arg1> MOD <arg2> - calcule le reste de la division <arg1> / <arg2>. Abrév. **%** . Voir [DIV](#) , [DIVISEURS](#) , [QUOTIENT](#) , [SOMME](#) .

Exemple: **ECRIS** 10 **MOD** 3 rendra 1 comme résultat.

[Début du catalogue.](#)

### MODEDEGRE :

Procédure **MODEDEGRE** - mode initial de calcul trigonométrique, place LOGO en mode degré. Tous les arguments passés aux fonctions [SINUS](#) , [COSINUS](#) , [TAN](#) seront exprimés en degrés pour donner un résultat correct.

Voir [MODERADIAN](#) , [DEGRE](#) , [PASAPAS](#) , [RADIAN](#) .

Exemple :

**MODEDEGRE**

[Début du catalogue.](#)

## MODULATION? :

K8061/VM140 Fonction MODULATION? :carte - fournit une valeur (comprise entre 0 et 100) de la largeur de l'impulsion du port PWM de la carte K8061 de rang :carte (0 <= :carte <=7). Voir [CONNEXIONS](#) , [DECONNECTE](#) , [FIXEMODULATION](#) .

Exemple:

```
DONNE "carte 0  
DONNE "p% MODULATION? :carte
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## MODERADIAN :

Procédure MODERADIAN - place LOGO en mode calcul radian. Tous les arguments passés aux fonctions [SINUS](#), [COSINUS](#), [TAN](#) seront exprimés en radians pour donner un résultat correct. Voir [MODEDEGRE](#) , [DEGRE](#), [PASAPAS](#) , [RADIAN](#) .

Exemple :

```
MODERADIAN
```

[Début du catalogue.](#)

## MONTRELUTIN :

Procédure MONTRELUTIN <lutin> - occasionne l'apparition de <lutin> sur l'écran.

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: MONTRELUTIN "lutin1

ou:

```
MONTRELUTIN [ "lutin1 "lutin2 ]
```

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

## MONTRESPOTS :

Procédure MONTRESPOTS - réalise l'ouverture de la fenêtre des capteurs optiques en vue d'une activité robotique. Voir [CACHESPOTS](#) , [SPOTS](#) .

Exemple :

## MONTRESPOTS

[Début du catalogue.](#)

### MONTRETORTUE :

Procédure MONTRETORTUE - permet de rendre la tortue visible à l'écran. Voir [CACHETORTUE](#) , [PASAPAS](#) .

Abréviation: [MT](#) .

Exemple: [MONTRETORTUE](#)

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### MOT :

Fonction MOT :mot1 :mot2 - réalise la fusion des deux mots. Voir [MOT?](#) , [DERNIERELETTRE](#) , [PASAPAS](#) , [PREMIERELETTRE](#) , [SAUFPREMIERELETTRE](#) , [SAUFDNIERELETTRE](#) , [SEPARE](#) , [TRIECAR](#) , [TRIEITEMS](#) , [xx || ...](#) .

Exemple: MOT "table "tte construit le mot "tablette.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### MOT? :

Fonction MOT? :NomVar - fournit la valeur booléenne [VRAI](#) en résultat si la variable NomVar correspond à un mot. [FAUX](#) sinon. Voir [MOT](#) , [CHOIX](#) , [DERNIERELETTRE](#) , [PASAPAS](#) , [PREMIERELETTRE](#) , [SAUFPREMIERELETTRE](#) , [SAUFDNIERELETTRE](#) , [TRIECAR](#) .

Exemple: [ECRIS](#) MOT? "juillet rendra [VRAI](#) comme résultat.

[Début du catalogue.](#)

### M\_ALEA :

Fonction M\_ALEA <n\_ligne> <k\_colonnes> <arg\_bool> ; construit une liste matricielle de n lignes et de k colonnes remplie de valeurs aléatoires en relation avec [FIXEHASARD](#). Si <arg\_bool> = [VRAI](#) (=1), des nombres entiers seront générés. Dans le cas contraire (<arg\_bool> = [FAUX](#) (=0)), ce sont des nombres décimaux qui seront cette fois générés. A retenir: les sous-listes de la liste matricielle produite représentent les vecteurs-colonnes. Voir [AFFICHE](#) , [DETERMINANT](#) , [LISMAT](#) , [M\\_DIAG](#) , [M\\_DIFF](#) , [M\\_IDENT](#) , [M\\_INVERSE](#) , [M\\_NORME](#) , [M\\_NULLE](#) , [M\\_SOMME](#) , [M\\_SUPPR](#) , [M\\_TRACE](#) ,

[M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [TRIEITEMS](#), [V\\_PROPRES](#),  
[V\\_UNIQUE](#) .

Exemple:

```
FIXEHASARD -3,5 4,2  
SOIT "matrice M_ALEA 4 3 FAUX
```

aura ce résultat de donner à la variable :matrice  
la valeur  $[[2,7 (-2) 3 (-3)][2 1,6 (-0,4) 3,1][-1,1 3,5 (-3) 1]]$  parmi d'autres.

[Début du catalogue.](#)

M\_DIAG :

Fonction M\_DIAG <liste> - où <liste> est une liste matricielle carrée d'ordre 2 ou 3 ; calcule et construit la liste matricielle de même ordre avec la diagonale principale remplie des valeurs propres de <liste>. Rappel: les sous-listes de :liste représentent les vecteurs-colonnes. Un argument incorrect transmis à M\_DIAG entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "matrice  $[[2 (-2) 3 ][2 1 (-0,5)][-1 4 (-3)]]$   
AFFICHE M_DIAG :matrice
```

rendra

```
 $[[ 0,29139389762078 0 0][0 2,28785390534934 0][0 0 -2,57924780297012 ]]$ 
```

comme résultat.

[Début du catalogue.](#)

M\_DIFF :

Fonction M\_DIFF <liste1> <liste2> - où :liste1 et :liste2 représentent des listes matricielles de mêmes dimensions ; calcule et construit la matrice <liste1> - <liste2>. Rappel: les sous-listes de la liste produite représentent les vecteurs-colonnes. Un argument incorrect transmis à M\_DIFF entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#),

[M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "liste1 [[7 2][ -3 1]] SOIT "liste2 [[7 6][ -1 9]]
AFFICHE M_DIFF :liste1 :liste2 FAUX
```

aura pour résultat l'affichage de  $[[0 \ (-4)][ -2 \ (-8)]]$  .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**M\_IDENT** :

Fonction `M_IDENT <n_lignes>` ; construit une liste matricielle carrée de n lignes et de n colonnes remplie de 1 sur sa diagonale principale. Rappel: les sous-listes de la liste produite représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#) , [M\\_ALEA](#) , [M\\_DIAG](#), [M\\_DIFF](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "matrice M_IDENT 3 3
```

aura pour résultat de donner à :matrice la valeur  $[[1 \ 0 \ 0][0 \ 1 \ 0][0 \ 0 \ 1]]$

[Début du catalogue.](#)

**M\_INVERSE** :

Fonction `M_INVERSE <liste_matrice>` - où <liste\_matrice> représente une liste matricielle carrée ; construit la matrice inverse de <liste\_matrice> . A retenir: les sous-listes de la liste produite représentent les vecteurs-colonnes. Un argument incorrect transmis à `M_INVERSE` entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#) , [M\\_ALEA](#) , [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
AFFICHE M_INVERSE [[10 (-100) 14,5][ -30 24,8 57][ 18 (-20) 31,01]]
```

aura pour résultat

```
[[ -0,0109518822306995 -0,0161262267635472 0,0347628899666989 ]  
[ -0,0112229588820802 -0,000281678311664947 0,00576551330393629 ] [ -  
0,000881176958691159 0,0091789266530329 0,0157877538432166 ]]
```

[Début du catalogue.](#)

### M\_NORME :

Fonction M\_NORME <liste> - où <liste> représente une liste matricielle ou vectorielle ; calcule la norme de <liste> . Rappel: les sous-listes de la liste produite représentent les vecteurs-colonnes. Un argument incorrect transmis à M\_NORME entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
AFFICHE M_NORME [[24,8 -30 57]][14,5 10 (-100)][-20 31,01 18]]
```

aura 96,0472284868231 pour résultat.

[Début du catalogue.](#)

### M\_NULLE :

FONCTION M\_NULLE <n\_lignes> <k\_colonnes> ; construit une liste matricielle de n lignes et de k colonnes remplie de 0. A retenir: les sous-listes de la liste matricielle produite représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "matrice M_NULLE 3 3 aura pour résultat de donner à :matrice  
la valeur [[0 0 0]][0 0 0][0 0 0] .
```

[Début du catalogue.](#)

### M\_SOMME :

Fonction M\_SOMME <liste1> <liste2> - où :liste1 et :liste2 représentent des listes matricielles de mêmes dimensions ; calcule et construit la matrice <liste1> + <liste2>. Rappel: les sous-listes de :liste1 et :liste2 représentent les vecteurs-colonnes. Un argument incorrect transmis à M\_SOMME entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#),

[DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#),  
[M\\_NULLE](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#),  
[PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "liste1 [[7 2][-3 1]]
SOIT "liste2 [[7 6][-1 9]]
AFFICHE M_SOMME :liste1 :liste2
```

aura pour résultat l'affichage de `[[14 8][-4 10]]` .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### M\_SUPPR :

Fonction `M_SUPPR <liste_matrice> [L/C [ rng1 rng2 ...]]` ; construit une sous-matrice de `<liste_matrice>` , supprimant les listes (L) ou colonnes (C) qui correspondent aux rangs `rng1, rng2 ...` . Les indices de rang redondants seront supprimés et les indices plus grands que le nombre de lignes (respectivement colonnes) de `<liste_matrice>` seront ignorés. A retenir: les sous-listes de `:liste_matrice` représentent les vecteurs-colonnes. Un argument incorrect transmis à `M_TRANSPOSE` entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "matrice [[1 2 3 4][5 6 7 8][9 10 11 12] [13 14 15 16]]
AFFICHE M_SUPPR :matrice [ L [2 4]]
```

occasionnera l'affichage de `[[1 3][5 7][9 11][13 15]]`

[Début du catalogue.](#)

### M\_TRACE :

Fonction `M_TRACE <liste_matrice>` ; calcule la somme des termes de la diagonale principale de `<liste_matrice>` n'est répété. Les valeurs redondantes seront alors remplacées par des valeurs aléatoires situées dans l'intervalle. Rappel: les sous-listes de `<liste_matrice>` représentent les

vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PROEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#).

Exemple:

```
SOIT "matrice [[5 7 (-3)][-8 2 1][14 (-6) 3]]
ECL M_TRACE :matrice
```

produira l'affichage de 10 (5+2+3) comme résultat.

[Début du catalogue.](#)

**M\_TRANSPOSE** :

Fonction `M_TRANSPOSE <liste_matrice>` - où `<liste_matrice>` représente une liste matricielle carrée ; construit la matrice transposée de `<liste_matrice>`. Rappel: les sous-listes de `:liste_matrice` représentent les vecteurs-colonnes. Un argument incorrect transmis à `M_TRANSPOSE` entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PROEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#).

Exemple:

```
AFFICHE M_TRANSPOSE [[2,84 35 75][14 10 (-100,5)][(-20,01) 31 16]]
```

aura `[[ 2,84 35 75][14 10 -100,5][-20,01 31 16]]` pour résultat.

[Début du catalogue.](#)

**M\_TRIINF** :

Fonction `M_TRIINF <liste_matrice>` - où `<liste_matrice>` représente une liste matricielle carrée ; construit une matrice formée des éléments du triangle inférieur de `<liste_matrice>`. Rappel: les sous-listes de `:liste_matrice` représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRISUP](#), [PROEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#).

Exemple:

```
SOIT "matrice [[1 7 8 ][ 2 3 9][ 4 5 6]]
AFFICHE M_TRIINF :matrice
```

affichera la matrice-triangle inférieure  $\begin{bmatrix} 1 & 0 & 0 \\ 2 & 3 & 0 \\ 4 & 5 & 6 \end{bmatrix}$ .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### M\_TRISUP :

Fonction `M_TRISUP` <liste\_matrice> - où <liste\_matrice> représente une liste matricielle carrée ; construit une matrice formée des éléments du triangle supérieur de <liste\_matrice>. Rappel: les sous-listes de :liste\_matrice représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [LISMAT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [PROEXT](#), [PRODINT](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
SOIT "matrice  $\begin{bmatrix} 1 & 2 & 3 \\ 7 & 4 & 5 \\ 8 & 9 & 6 \end{bmatrix}$ "  
AFFICHE M_TRISUP :matrice FAUX
```

affichera la matrice-triangle supérieure  $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 4 & 5 \\ 0 & 0 & 6 \end{bmatrix}$ .

[Début du catalogue.](#)

### NEGATIVE? :

Fonction `NEGATIVE?` <argnum> - fournit la valeur booléenne [VRAI](#) en résultat si <argnum> est inférieur à zéro. [FAUX](#) sinon.

Exemple: `ECRIS NEGATIVE? (-5)` rendra [VRAI](#) comme résultat.

[Début du catalogue.](#)

### NETTOIE :

Procédure `NETTOIE` - réalise l'effacement complet de l'écran de la tortue sans modifier la position des tortues actives. Voir [EFFACEECRAN](#), [LIEU](#), [LIMITES?](#), [PASAPAS](#) .

Exemple: `NETTOIE`

[Début du catalogue.](#)

### NOMBRE :

Fonction `NOMBRE` <arg> où <arg> est un mot - fournit en résultat la valeur numérique de <arg>. Voir [NOMBRE?](#), [MAX?](#), [MIN?](#), [PASAPAS](#), [PERMUTATIONS](#), [e](#), [E](#) .

Exemple: **DONNE** "numéro "1234 // éq : **SOIT** "numéro "1234  
**ECRIS NOMBRE** :numéro aura pour résultat l'affichage de 1234.

[Début du catalogue.](#)

### NOMBRE? :

Fonction **NOMBRE?** <arg> - fournit la valeur booléenne **VRAI** en résultat si la variable <arg> correspond à une valeur numérique. **FAUX** sinon. Voir [NOMBRE](#) , [PASAPAS](#) , [PERMUTATIONS](#) .

Exemple: **ECRIS NOMBRE?** 1234 rendra **VRAI** comme résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### NON :

Fonction booléenne **NON** <arg booléen> - retourne la valeur booléenne **VRAI** en résultat si <arg booléen> correspond à **FAUX** et inversement. Voir [OPPOSE](#) , [PASAPAS](#) .

Exemple: **ECRIS NON** VRAI rendra **FAUX** comme résultat.

[Début du catalogue.](#)

### OBEIS :

Primitive **OBEIS** <arg num> ou **OBEIS** [ arg1 arg2 ... ] où <arg num>, arg1 arg2 ... sont des valeurs numériques comprises entre 1 et 16 - réalise l'activation de ces tortues aux consignes LOGO suivantes: AVANCE, RECULE, BAISSSECRAYON, LEVECRAYON, CACHETORTUE, COULEURTORTUE, MONTRETORTUE, FIXECAP, FIXEX, FIXEXY, FIXEY, TOURNEDROITE, TOURNEGAUCHE, RAMENE, VA, VERS. Voir [FORME](#) , [PASAPAS](#) , [TORTUES?](#) .

Exemples:

**OBEIS** 5

ou:

**OBEIS** [ 1 2 3 ]

[Début du catalogue.](#)

### OCCURRENCES :

Fonction **OCCURRENCES** <liste> <item> <complète> ou **OCCURRENCES** <mot principal> <mot> <casse prise en compte> ; recherche le nombre d'occurrences de <item> dans <liste>. L'argument booléen <complète> indique si cette recherche doit se faire seulement sur le premier niveau de <liste> (i.e là où les item immédiatement accessibles) s'il est **FAUX** ou si elle doit être faite également à l'intérieur des sous-listes de <liste>. Si l'argument principal est un mot, le paramètre <casse prise en compte> = **VRAI** indiquera à la fonction **OCCURRENCES** si elle doit tenir compte de la différence

majuscules/minuscules dans le processus de comptabilisation d'un mot. Abrév: OCCURS . [CHOIX](#) , [DERNIER](#) , [INSERE](#) , [ITEM](#) , [LISTE](#) , [LISTEVARS](#) , [MEMBRE?](#) , [PERMUTATIONS](#) , [PREMIER](#) , [REMPLECE](#) , [SAISIS](#) , [SAUFPREMIER](#) , [SAUFDERNIER](#) , [TRIECAR](#) , [TRIEITEMS](#) , [\\*\\* || ...](#) .

Exemples :

ECL OCCURRENCES [ a [ b c ] d [ e [ b c ] f ] g h [ b c ] i ] [ b c ] VRAI rendra 4 en résultat.

ECL OCCURRENCES [ a [ b c ] d [ e [ b c ] f ] g h [ b c ] i ] [ b c ] FAUX rendra 2 en résultat.

ECL OCCURRENCES "abcdeBCfghBCi "bc FAUX rendra 3 en résultat.

ECL OCCURRENCES "abcdeBCfghBCi "BC VRAI rendra 2 en résultat.

ECL OCCURRENCES 3,14159265358979 9 FAUX rendra 3 en résultat.

[Début du catalogue.](#)

OPPOSE :

Fonction OPPOSE <arg booléen> ou OPPOSE <argnum> ( OPPOSE <arg booléen> est équivalente à la fonction booléenne NON (Voir [NON](#))). OPPOSE <argnum> - fournit en résultat sa valeur positive si <argnum> est négatif, sa valeur négative si <argnum> est positif. Voir [PASAPAS](#) .

Exemple:

ECRIS OPPOSE (-5)

rendra 5 comme résultat.

ECRIS OPPOSE ( FAUX )

rendra VRAI comme résultat.

[Début du catalogue.](#)

ORIGINE :

Procédure [ORIGINE](#) - place la tortue au centre de l'écran sans l'effacer. Equivaut à [FIXEX 0](#) [FIXEY 0](#) .

Voir [EFFACEECRAN](#) , [FIXEX](#) , [FIXEXY](#) , [FIXEY](#) , [LIEU](#) , [LIMITES?](#) , [NETTOIE](#) , [PASAPAS](#) .

[Début du catalogue.](#)

OU:

Connecteur logique OU: <arg1> OU <arg2> - fournit en résultat les valeurs booléennes suivantes selon les valeurs booléennes respectives de <arg1> et <arg2> - [VRAI](#) OU VRAI -> VRAI, [FAUX](#) OU VRAI -> VRAI, VRAI OU FAUX -> VRAI, FAUX OU FAUX -> FAUX. Voir [ET\(&\)](#) , [LESDEUX?](#) , [UNDESDEUX?](#) . Alias : | (<CTRL droit>+6)

Exemple: [DONNE](#) "x 6 // éq: [SOIT](#) "x 6

[TESTE](#) (:x > 0) [OU](#) (:x < 5) rendra [VRAI](#) en résultat (car :x > 0).

// éq: [TESTE](#) (:x > 0) | (:x < 5)

[Vers l'index des primitives >>](#)

**OUVERT?** :

K8055/VM110> Fonction **OUVERT?** :canal - lit l'état du canal d'entrée logique. Rends VRAI si le canal est sous tension, FAUX si il est au niveau bas. (la valeur de l'argument :canal doit être compris entre 1 et 5). Voir [CONNECTE](#) , [OUVERTS?](#) .

Exemple:

```
CARTEUSB "VM110
DONNE "test OUVERT? 0
```

K8061/VM140> Fonction **OUVERT?** :carte :canal - lit l'état du canal d'entrée logique. Rends VRAI si le canal est sous tension, FAUX si il est au niveau bas. (la valeur de l'argument :canal doit être compris entre 1 et 8). (0<= :carte <=7). Voir [CONNEXIONS](#) , [FERMECARTE](#) , [OUVERTS?](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte CONNEXIONS
DONNE "test OUVERT? 0 5
```

**OUVERTS?** :

K8055/VM110> Fonction **OUVERTS?** - lit l'état des cinq canaux d'entrée et retourne une liste binaire de 5 nombres (0 : niveau bas ou 1 : niveau haut) positionnés dans la liste selon l'index du canal associé. Voir [CONNECTE](#) , [OUVERT?](#) .

Exemple:

```
CARTEUSB "VM110
CONNECTE 0 // On connecte la carte . Les jumps sk5 et sk6 sont sensés être présents.
TANTQUE 5 < 6 [ /* C'est une boucle sans fin. */
    EFFACETEXTE ECRISLIGNE OUVERTS?
    /* On examine l'état des 5 canaux d'entrée logique et on retourne une liste de 5 bits
    correspondant à chaque canal : [ c5 c4 c3 c2 c1 ]
    Il faut agir sur les interrupteurs pour modifier la liste. */
    ATTENDS 1
]
```

K8061/VM140> Fonction **OUVERTS?** :carte - lit l'état des huit canaux d'entrée et retourne une liste binaire de 8 nombres (0 : niveau bas ou 1 : niveau haut) positionnés dans la liste selon l'index du canal associé. (0<= :carte <=7). Voir [CONNEXIONS](#) , [FERMECARTE](#) , [OUVERT?](#) .

Exemple:

```
CARTEUSB "K8061
```

```

DONNE "adresses CONNEXIONS /* On connecte la carte K8061. */
SI NON (VIDE? :adresses ) [
  DONNE "carte PREM :adresses
  TANTQUE 5 < 6 [ /* C'est une boucle sans fin. Utiliser le bouton "Stop" pour
                  l'interrompre. */
    EFFACETEXTE ECRISLIGNE OUVERTS? :carte
    /* On examine l'état des 8 canaux d'entrée logique et on retourne une liste de 8 bits
       correspondant à chaque canal : [ c7 c6 c5 c4 c3 c2 c1 ] Il faut agir sur les
       interrupteurs pour modifier la liste. */
    ATTENDS 1
      ]
    ]

```

[Début du catalogue.](#)

**OUVERTS\_A?** :

K8061/VM140> Fonction OUVERTS\_A? :carte - fournit l'état des huit canaux analogiques de la carte K8061 de rang :carte (0 <= :carte <=7) sous forme d'une liste de huit valeurs numériques. Voir CONNEXIONS , FIXECANAL\_L , FIXECANAUX\_L , FERMECANAL , FERMECANAUX , FERMECARTE , OUVRECANAL , OUVRECANAUX .

Exemple: DONNE "canauxa OUVERTS\_A? 0

[Début du catalogue.](#)

**OUVRECANAL** :

K8055/VM110> Procédure OUVRECANAL :valeur - ouvre un ou plusieurs canaux de sortie numérique. (:valeur doit être compris entre 1 et 8 ). Voir CONNECTE , FERMECANAL , FIXECANAUX\_A .

Exemple:

```

CARTEUSB "VM110
OUVRECANAL 1

```

K8061/VM140> Procédure OUVRECANAL :carte :valeur - ouvre un ou plusieurs canaux de sortie numérique. (:valeur doit être compris entre 1 et 8 ). (0<= :carte <=7). Voir CONNEXIONS , FERMECANAL , FERMECARTE , FIXECANAUX\_A .

Exemple:

```

CARTEUSB "VM140
DONNE "carte CONNEXIONS
OUVRECANAL :carte 3

```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**OUVRECANAUX** :

K8055/VM110> Procédure [OUVRECANAUX](#) - ouvre les huit canaux de sortie. Voir [FERMECANAUX](#) .

Exemple:

```
CARTEUSB "VM110
DONNE "carte CONNEXIONS
```

K8061/VM140> Procédure [OUVRECANAUX](#) :carte - ouvre les huit canaux de sortie. (0<= :carte <=7). Voir [CONNEXIONS](#) , [FERMECANAL](#) , [FERMECARTE](#) , [FIXECANAUX\\_A](#) , [FERMECANAUX](#) .

Exemple:

```
CARTEUSB "VM140
DONNE "carte CONNEXIONS
OUVRECANAUX :carte
```

[Début du catalogue.](#)

OUVREDLL :

Fonction [OUVREDLL](#) <chemin accès dll> - alloue une place en mémoire pour la bibliothèque dynamique et retourne VRAI ou FAUX en résultat. Par sécurité, un chemin d'accès incorrect à la dll entraînera un arrêt immédiat du programme LOGO en cours. Voir [EXECDLL](#), [FERMEDLL](#) .

Exemple: [OUVREDLL](#) [ c:/Program files/Education/LogoPlus/bibdyn.dll ]

[Début du catalogue.](#)

OUVREPORT:

Procédure [OUVREPORT](#) <argnum> - Primitive de gestion du port série (RS 232) - ouvre le port précisé par <argnum> en vue d'une transmission de données: Si <argnum> vaut 1, COM1 sera ouvert. <argnum> doit être compris entre 1 et 4. Voir aussi [PORT](#), [LISPORT](#) , [PORTS?](#) .

Exemple: [OUVREPORT](#) 1

[Début du catalogue.](#)

PARTAGE :

Procédure [PARTAGE](#) ["nvar1 "nvar2 ...] ou [PARTAGE](#) "nvar - définit les variables contenues dans la liste ou suivant la primitive [PARTAGE](#) **dans la procédure initiale** (i.e la procédure qui appelle les autres procédures de votre programme LOGO) comme valables dans les procédures appelées, rendant ainsi leur usage possible dans la suite du programme, cela tant que votre procédure initiale (**appelante**) n'est pas terminée. Voir [PASAPAS](#) .

Exemple: [PARTAGE](#) [ "var1 "var2 "var3 ]      ou:      [PARTAGE](#) "var4

[Début du catalogue.](#)

## PASAPAS :

Procédure PASAPAS <arg\_booléen> ; autorise ou non le mode pas à pas du déroulement du programme selon que <arg\_booléen> vaut VRAI (1) ou FAUX (0). Cette primitive interrompt le déroulement du programme seulement sur quelques autres primitives choisies parce qu'elles ne diffèrent pas déjà le déroulement du programme et qu'elles n'impliquent pas des processus robotisés qui nécessitent impérativement un déroulement fluide du programme. Ces autres primitives sont les suivantes: [ACCELERE](#) [AFFICHE](#) [AUCUNE](#) [AVANCE](#) [BAISSECRAYON](#) [BOUCLE](#) [CACHETORTUE](#) [CAP](#) [CARACTERE](#) [CARDINAL](#) [CHOIX](#) [CHOSE](#) [COPIE](#) [COULEURFOND](#) [COULEURFOND?](#) [COULEURTORTUE](#) [COULEURTORTUE?](#) [DATE](#) [DEC](#) [DEFINIS](#) [DERNIER](#) [DERNIERELETTRE](#) [DEROULE](#) [DETRUIS](#) [DIFFERENCE](#) [DONNE](#) [ECRIS](#) [ECRISLIGNE](#) [EFFACEECRAN](#) [EFFACELIGNE](#) [EFFACETEXTE](#) [ENROULE](#) [ENTIER](#) [FIXECAP](#) [FIXEHASARD](#) [FIXEX](#) [FIXEXY](#) [FIXEY](#) [FORME](#) [HASARD](#) [INC](#) [INSERE](#) [ITEM](#) [LEVECRAYON](#) [LIEU](#) [LIMITES?](#) [LISTE](#) [LISTE?](#) [LISTEVARS](#) [MAX?](#) [MEMBRE?](#) [MIN?](#) [MODEDEGRE](#) [MODERADIAN](#) [MONTRETORTUE](#) [MOT](#) [MOT?](#) [NEGATIVE?](#) [NETTOIE](#) [NOMBRE](#) [NOMBRE?](#) [OBEIS](#) [OPPOSE](#) [ORIGINE](#) [PARTAGE](#) [PERMUTE](#) [PHRASE](#) [PI](#) [PREMIER](#) [PREMIERELETTRE](#) [PRODUIT](#) [QUOTIENT](#) [RALENTIS](#) [RAMENE](#) [RANDDEC](#) [RANDOM](#) [RECULE](#) [REPLACE](#) [REPETE](#) [REPONDRE](#) [RESTE](#) [RVB](#) [SAISIS](#) [SAUFDERNIER](#) [SAUFDERNIERELETTRE](#) [SAUFPREMIER](#) [SAUFPREMIERELETTRE](#) [SENTHORAIRE](#) [SENSTRIGO](#) [SILENCE](#) [SIFAUX \(SINON\)](#) [SIVRAI](#) [SOMME](#) [SOURISX](#) [SOURISY](#) [STOP](#) [SUSPENDRE](#) [TESTE](#) [TOURNEDROITE](#) [TOURNEGAUCHE](#) [TOUTES](#) [TRACE](#) [TRAIT](#) [TRANSFORME](#) [VA VERS](#) [VIDE?](#) [VISIBLE?](#) [XCOORD](#) [YCOORD](#). L'interception de l'une de ces primitives occasionnera l'affichage de la primitive interceptée, la ligne du programme LOGO où elle se situe et la liste des variables rencontrées dans la fenêtre prévue à cet effet. Ce mode ne fonctionne qu'à l'intérieur du programme en cours et disparaît lorsqu'il est définitivement terminé. Toute connexion extérieure à une carte de communication (USB ou RS232) annule le mode "pas à pas". Abrév. : **PAP** . n.b : L'écriture PAS A PAS (avec deux espaces) fonctionne aussi.

Exemple :

**PASAPAS VRAI**

[Début du catalogue.](#)

## PERMUTATIONS :

Fonction PERMUTATIONS <argument - fournit en résultat une liste des permutations selon le type d'argument qui lui a été transmis. Elle accepte 3 types d'argument :

Premier cas: si <argument> est une liste PERMUTATIONS <Maliste> construira et retournera en résultat la liste de toutes les permutations circulaires des items de <Maliste>. Voir [DERNIER](#), [DERNIERELETTRE](#), [DIVISEURS](#), [ITEM](#), [LISTE?](#), [LISTEVARS](#), [MAX?](#), [MEMBRE?](#), [MIN?](#), [OCCURRENCES](#), [PHRASE](#), [PREMIER](#), [PREMIERELETTRE](#), [REPLACE](#), [SAISIS](#),

[SAUFPREMIER](#), [SAUFPREMIERELETTRE](#), [SAUFDERNIER](#), [SAUFDERNIERELETTRE](#), [TRIEITEMS](#), [xx || ...](#), [ /e ou /E ] .

Exemple 1:

**ECL PERMUTATIONS** [a b c ] retournera la liste :  
[[a b c][a c b][b a c][b c a][c a b][c b a]] .

Deuxième cas: si <argument> est un mot, PERMUTATIONS <Monmot> construira et retournera en résultat la liste des mots formés de toutes les permutations des caractères de ce mot.

Exemple 2:

**ECL PERMUTATIONS** "abc retournera la liste:  
[abc acb bac bca cab cba] .

Troisième cas: si <argument> est un nombre, PERMUTATIONS <Monnombre> construira et retournera en résultat la liste des permutations des chiffres de <Monnombre>.

Exemple 3:

**PERMUTATIONS** 123 retournera la liste :  
[123 132 231 213 312 321] .

**Attention** : le nombre d'items à permuter doit être strictement inférieur à 8.

[Début du catalogue.](#)

**PERMUTE** :

Procédure PERMUTE <donnée1> <donnée2> - échange les contenus des données <donnée1> et <donnée2>.

Voir [DONNE / SOIT](#), [DEFINIS](#), [PARTAGE](#), [PASAPAS](#) .

Exemple:

**DONNE** "donnée1 5 **DONNE** "donnée2 6  
**PERMUTE** :donnée1 :donnée2 . :donnée1  
contient à présent la valeur 6 et :donnée2 contient la valeur 5 .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**PHRASE** :

Fonction PHRASE :liste1 :liste2 - réalise la fusion des deux listes en supprimant les attributs des listes internes à la nouvelle liste formée. Voir [CHOIX](#), [DERNIER](#), [ITEM](#), [LISTE](#), [LISTE?](#),

[LISTEVARS](#), [MEMBRE?](#), [PASAPAS](#), [PERMUTATIONS](#), [PREMIER](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [TRIEITEMS](#), [xx || ...](#) .

Abréviation: PH

Exemple: PHRASE [marguerite] [ violette pétunia ]

construit la liste [marguerite violette pétunia ].  
[ violette ] devient le second item de la liste formée.

Attention :

Ne pas confondre :

SOIT "mphrase PHRASE [ item1] [item2] qui construira la liste [item1 item2]

et :

SOIT "maliste LISTE [ item1] [item2] qui construira la liste [[item1][item2]]

Remarque : Lorsque l'un des arguments :liste1 et :liste2 (ou les deux en même temps) n'est (ne sont) pas de type liste, il(s) est (sont) transformé(s) en liste soit au moment où LOGO apprend le texte, soit au cours du déroulement du programme.

[Début du catalogue.](#)

PI :

Fonction PI - fournit en résultat la valeur numérique 3,14159265... de la constante PI.  
Voir [DEC](#) , [DEGRE](#) , [INC](#) , [INTEGRALE](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [PASAPAS](#) , [RADIAN](#) , [ZEROS?](#) .

Exemple: ECRISLIGNE PI

[Début du catalogue.](#)

PIXELS? :

Fonction PIXELS? <arg> (où <arg> est une liste de type LIEU) - fournit en résultat la liste des points contigus à <arg> et qui sont de la même couleur. Le cardinal de cette liste revient à intégrer graphiquement la zone tout autour du point <arg> et donne une évaluation en pixels de l'aire de cette zone. Pour éviter des temps de recherche trop longs (cas d'une surface très grande), le cardinal maximum de cette liste de pixels a été fixé à 3000 pixels. Voir [COLORIE](#), [FONDECRAN](#), [POINT](#).

Exemple: ECRISLIGNE PIXELS? [ 50 (-10)]

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## PLEINECRAN :

Procédure PLEINECRAN - ouvre une fenêtre qui affiche la totalité de l'écran de la tortue.

Exemple: PLEINECRAN

n. b. : Cette primitive recopie l'intégralité de l'écran graphique dans une zone-mémoire tampon (bitmap), ce qui prend du temps et ralentit de ce fait la vitesse de tracé des tortues ainsi que le déroulement du programme LOGO.

Il est préférable de n'ouvrir la fenêtre PLEINECRAN qu'à la fin d'un programme LOGO. Elle est de plus insensible aux clics de la souris.

### [Début du catalogue.](#)

## PLOT :

Procédure PLOT [X Y Z nom] - trace un point de coordonnées (X,Y,Z) sur l'écran du grapheur de fonctions. L'item <nom> est facultatif. Voir [CHAMPVECT](#) , [INITGRAPH](#) , [GRAPHE](#) , [GRAPH2D](#) , [GRAPH3D](#) , [GRAPHEUR](#) , [PLOTPOL](#) , [SEGMENT3D](#) , [SURFACE](#) , [SYSLOT](#) , [SYSLOTPOL](#) .

Exemple 1:

```
DONNE "x 50 DONNE "y -25 DONNE "z 42
```

```
PLOT [:x :y :z point_n°1]
```

aura pour résultat de faire apparaître un point à l'emplacement (50,-25,42) .

Exemple 2:

```
DONNE "liste1 [ 0 100 100 ] DONNE "liste2 [ 100 (-100) (-100) ]
```

```
INITGRAPH FAUX
```

```
GRAPHEUR FAUX
```

```
PLOT [ 100 120 90 ]
```

```
PLOT [ 100 (-90) 90 ]
```

```
PLOT :liste1
```

```
PLOT [ 80 90 (-90) ]
```

```
PLOT :liste2
```

```
GRAPHEUR VRAI
```

### [Début du catalogue.](#)

## PLOTPOL :

Procédure PLOTPOL [Rho Theta Phi nom] ; trace un point de coordonnées (Rho Theta Phi) sur

l'écran du grapheur de fonctions. L'item <nom> est facultatif. Voir [CHAMPVECT](#), [INITGRAPH](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [GRAPHEUR](#), [PLOT](#), [SEGMENT3D](#), [SURFACE](#), [SYSPLOT](#), [SYSPLOTPOL](#).

Exemple :

DONNE "Rho 15

DONNE "Theta 45

DONNE "Phi 120

PLOTPOL [:Rho :Theta :Phi Point\_n°1]

GRAPHEUR VRAI

aura pour résultat de faire apparaître un point aux coordonnées polaires (15,45,120), ce qui correspond approximativement à (-5,303 9,18 10,6) en coordonnées rectangulaires.

[Début du catalogue.](#)

POINT :

Procédure POINT <lieu> - trace un point de la couleur en cours au point de coordonnées fixées par l'argument <lieu> sur l'écran de la tortue. <lieu> est sous forme d'une liste de type LIEU.

Voir [LIEU](#), [FIXEX](#), [FIXEXY](#), [FIXEY](#), [PASAPAS](#), [PIXELS?](#).

Exemple: SOIT "x 50 SOIT "y -25 POINT [:x :y]

aura pour résultat de faire apparaître un point de couleur aux coordonnées x = 50 et y = -25.

[Début du catalogue.](#)

POLYSOLVE :

Fonction POLYSOLVE [a(n) a(n-1) ... a(0)] où les a() sont les coefficients d'un polynome en x rangés par ordre décroissant des puissances de la variable, a(0) étant le terme constant et n <= 4 ; fournit la liste des racines du polynome indiqué par ses coefficients. Les solutions imaginaires sont placées dans une sous-liste [ part\_réelle part\_imag ]. S'il n'existe aucune solution, une liste vide sera retournée. Pour une puissance supérieure à 4, préférer la primitive ZEROS? pour une recherche dichotomique des racines. Voir [ZEROS?](#).

Exemple:

ECL POLYSOLVE [6 3 (-2)] pour calculer les solutions du polynome  $6x^2+3x-2=0$ .

[Début du catalogue.](#)

PORT:

Procédure PORT [DTR RTS TXD] (Primitive de gestion du port série (RS 232)) - affecte aux sorties DTR RTS et DTR la valeur 1 (état haut) pour une durée déterminée (en millisecondes),

correspondant au port ouvert par la primitive OUVREPORT . Toute requête d'interruption d'un programme LOGO sera considérée comme un arrêt définitif de ce programme si cette requête survient pendant l'activité du port. Voir [OUVREPORT](#), [LISPORT](#), [PORTS?](#) .

Exemple: OUVREPORT 1 PORT [ 2000 500 1000 ]

affectera DTR à 1 pendant 2 secondes, RTS à 1 pendant  $\frac{1}{2}$  seconde et TXD à 1 pendant 1 seconde.

[Début du catalogue.](#)

### PORTS? :

Fonction PORTS? (Primitive de gestion du port série (RS 232)) - fournit en résultat la liste des numéros de port accessibles à LOGO. 1 correspond à COM1, 2 à COM2, 3 à COM3 et 4 à COM4. Voir [OUVREPORT](#), [LISPORT](#), [PORT](#).

Exemple: ECRISLIGNE PORTS?

[Début du catalogue.](#)

### POSITIONLUTIN? :

Fonction POSITIONLUTIN? <lutin> - fournit en résultat la liste composée des valeurs d'abscisse et d'ordonnée de <lutin>. Voir [MODEDEGRE](#), [MODERADIAN](#), [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [LUTINX?](#), [LUTINY?](#), [LUTINXV](#), [LUTINYV](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemple: POSITIONLUTIN? "lutin

"lutin doit cependant être défini dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### POUR :

Primitive POUR <nomproc> <arg1 arg2 ...> - définit le nom d'une procédure. Il doit toujours être associé au mot FIN . Ces mots-clés ne doivent jamais figurer sur la ligne de commande (un message d'erreur apparaîtra alors). Voir [FIN](#) , [REPONDRE](#) , [RETOURNE](#) , [STOP](#) .

Exemple:

POUR tracer

EFFACEECRAN carré 50 COLORIE [ 5 10 ] ROSE

ATTENDS 2 LEVECRAYON AVANCE 70 ATTENDS 2  
BAISSECRAYON rectangle 80 30 COLORIE [ 5 130 ] JAUNE  
FIN

[Début du catalogue.](#)

### PREMIER :

Fonction PREMIER [arg] - fournit en résultat le premier item de la liste [arg]. Si un mot est passé à cette fonction en argument, une liste vide sera retournée en résultat. Voir [CHOIX](#) , [DERNIER](#) , [INSERE](#) , [ITEM](#) , [LISTE](#) , [LISTEVARS](#) , [MEMBRE?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [REPLACE](#) , [SAISIS](#) , [SAUFPREMIER](#) , [SAUFDERNIER](#) , [TRIECAR](#) , [TRIEITEMS](#) , [xx || ...](#) .

Abréviation: **PREM**

Exemple: **ECRIS PREMIER** [un deux trois]                      rendra: un comme résultat.

[Début du catalogue.](#)

### PREMIERELETTRE :

Fonction PREMIERELETTRE "arg - fournit en résultat le premier caractère du mot "arg . Voir [CHOIX](#) , [DERNIERELETTRE](#) , [MOT](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [SAUFPREMIERELETTRE](#) , [SAUFDERNIERELETTRE](#) , [SEPRE](#) , [TRIECAR](#) , [TRIEITEMS](#) , [xx || ...](#) .  
Abrév: **PREML**

Exemple: **ECRIS PREMIERELETTRE** "tables                      rendra "t" comme résultat.

[Début du catalogue.](#)

### PROCEDURES? :

Procédure **PROCEDURES?** - réalise l'affichage de toutes les procédures déclarées dans le programme.

Exemple: **PROCEDURES?**

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### PRODEXT :

Fonction PRODEXT <liste1> <liste2> - où :liste1 est une matrice carrée et :liste2 est soit une matrice carrée de mêmes dimensions que <liste1> ou une liste vectorielle dont le nombre de lignes est identique au nombre de colonnes de <liste1> (PRODEXT signifie produit extérieur) . Rappel: les sous-listes de :liste1 et :liste2 représentent les vecteurs-colonnes. Voir [AFFICHE](#) , [DETERMINANT](#) ,

[M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#),  
[M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODINT](#), [V\\_PROPRES](#),  
[V\\_UNIQUE](#).

Exemple 1:

```
SOIT "m1 [[10 20 (-30) ][7 9 2][8 (-3) 15]]
SOIT "m2 [[(-10) 9 2][7 20 (-30) ][(-8) (-3) 15]]
AFFICHE PRODEXT :m1 :m2
```

occasionnera l'affichage de la liste matricielle  
[[280 580 -1030][ -23 237 -226][ -221 -33 331]]

Exemple 2:

```
SOIT "M [[-2 2 ][0 2][ -1 4] [-1 1]]
SOIT "e [4 5 1 15]
AFFICHE PRODEXT :M :e
```

occasionnera l'affichage de la liste vectorielle [-24 37].

### Début du catalogue.

#### PRODINT :

Fonction PRODINT <liste\_matrice> <arg\_num> ; construit une matrice construite en multipliant chaque terme de <liste\_matrice> par <arg\_num> (PRODINT signifie produit intérieur). Rappel: les sous-listes de :liste\_matrice représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#),  
[M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#),  
[M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [V\\_PROPRES](#),  
[V\\_UNIQUE](#).

Exemple:

```
SOIT "matrice [[1 2][3 (-4)]]
AFFICHE PRODINT :matrice (-5)
```

produira l'affichage de [[5 (-10)][(-15) 20]]

### Début du catalogue.

#### PRODUIT :

Fonction **PRODUIT** <arg num> où <arg num> est une liste de valeurs ou de variables numériques - calcule le produit généralisé des termes inclus dans <arg num>. Abréviation: PROD .  
Voir [ [+ - \\* /](#) ], [DEC](#) , [DIFFERENCE](#) , [DIV](#) , [FVECT](#) , [INC](#) , [MAX?](#) , [MIN?](#) , [MOD \(%\)](#) , [PASAPAS](#) , [QUOTIENT](#) , [RESTE \(%\)](#) , [RACINE2](#) , [RACINE3](#) , [SOMME](#) , [TRIEITEMS](#) .

Exemple: **DONNE** "v -4 **ECRIS PRODUIT** [ :v 4 2 5 ]

rendra -160 comme résultat.  
( -4 \* 4 \* 2 \* 5 )

[Début du catalogue.](#)

## **PUISSANCE** ( ^ ):

Fonction **PUISSANCE** <argnum1> <argnum2> - calcule le résultat de <argnum1> à la puissance de <argnum2>. Voir [DEC](#) , [EXP](#) , [INC](#) , [INTEGRALE](#) , [LOG](#) , [LN](#) , [MAX?](#) , [MIN?](#) , [RACINE2](#) , [RACINE3](#) , [ZEROS?](#) .

Exemples: **ECRIS PUISSANCE** 5 2

rendra 25 comme résultat.

**ECRIS PUISSANCE** 5 (-2)

rendra 0,04 comme résultat.

n.b : La primitive **PUISSANCE** peut être remplacée par l'opérateur d'élévation à la puissance ^ .

Exemples: **ECRIS** 5 ^ 2 rendra 25 comme résultat.

**ECRIS** 5 ^ (-2) rendra 0,04 comme résultat.

[Début du catalogue.](#)

## **QUOTIENT** :

Fonction **QUOTIENT** <arg num> où <arg num> est une liste de valeurs ou de variables numériques - calcule le quotient généralisé des termes inclus dans <arg num>. Sauf le premier item de la liste, les autres items doivent être différents de zéro. Voir [ [+ - \\* /](#) ], [DEC](#) , [DIFFERENCE](#) , [DIV](#) , [DIVISEURS](#) , [FVECT](#) , [INC](#) , [MAX?](#) , [MIN?](#) , [PASAPAS](#) , [PRODUIT](#) , [RESTE \(%\)](#) , [RACINE2](#) , [RACINE3](#) , [SOMME](#) , [TRIEITEMS](#) , [MOD \(%\)](#) .

Abréviation : **QUOT**

Exemple: **SOIT** "v 2 **ECRIS QUOTIENT** [ 256 4 :v 8 ] rendra 4 comme résultat.

// éq **ECRIS** 256 / 4 / 2 / 8

[Début du catalogue.](#)

## **RACINE2** :

Fonction **RACINE2** <argnum> - calcule la valeur de la racine carrée d'un nombre <argnum>. Un

argument négatif aura pour conséquence l'arrêt du programme LOGO. Voir [INTEGRALE](#) , [RACINE3](#) , [PUISSANCE](#) , [DIV](#) , [FVECT](#) , [MOD](#) , [ZEROS?](#) . Abréviation : **RAC2**

Exemple: **ECRIS RACINE2 16** rendra 4 comme résultat.

[Début du catalogue.](#)

### **RACINE3** :

Fonction RACINE3 <argnum> - calcule la valeur de la racine cubique d'un nombre <argnum>. Si l'argument est négatif, le résultat sera négatif mais n'entraînera pas l'arrêt du programme LOGO. Voir [DIV](#) , [FVECT](#) , [INTEGRALE](#) , [MOD](#) , [PUISSANCE](#) , [RACINE2](#) , [ZEROS?](#) . Abréviation : **RAC3**

Exemple: **ECRIS RACINE3 27** rendra 3 comme résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### **RADIAN** :

Fonction RADIAN <argnum> - calcule la valeur en radians d'un nombre exprimé en degrés. Voir [DEGRE](#) , [SENSHORAIRE](#) , [SENSTRIGO](#) , [MODEDEGRE](#) , [MODERADIAN](#) .

Exemple: **ECRIS RADIAN 180** rendra la valeur 3,14159... comme résultat.

[Début du catalogue.](#)

### **RALENTIS** :

Procédure **RALENTIS** - ralentit la vitesse de déplacement de la tortue. C'est le mode initial du déplacement de la tortue. N'affecte en aucun cas la rapidité des calculs numériques. Voir [ACCELERE](#) , [PASAPAS](#) .

[Début du catalogue.](#)

### **RAMENE** :

Procédure RAMENE [arg] , où [arg] est une liste de type [SITUATION](#) comportant trois valeurs numériques définissant le lieu et le cap de la tortue - a pour effet de ramener la tortue dans une situation topologique antérieure.

VOIR [LIEU](#) , [LIMITES?](#) , [NETTOIE](#) , [VA](#) , [VERS](#) , [FIXEX](#) , [FIXEXY](#) , [FIXEY](#) , [FIXECAP](#) , [PASAPAS](#) , [SITUATION](#) .

Exemple: **DONNE "AnciennePosition SITUATION**

/\* On supposera que SITUATION équivaut à [-20 56 120]. \*/

...

RAMENE :AnciennePosition

/\* La tortue reprend sa situation antérieure [-20 56 120]. \*/

[Début du catalogue.](#)

### RANDDEC :

Fonction numérique RANDDEC <randinf> <randsup> - fournit en résultat un nombre décimal aléatoire compris entre <randinf> <randsup>. La partie décimale est cependant limitée à 5 chiffres maximum. Les paramètres assignés à RANDDEC ne modifient en rien les bornes de l'ensemble des choix fixés par FIXEHASARD . Voir [HASARD](#) , [FIXEHASARD](#) , [PASAPAS](#) , [RANDOM](#) .

Exemple: SOIT "nombre RANDDEC 5 10.

### RANDOM :

Fonction numérique RANDOM <randinf> <randsup> - fournit en résultat un nombre entier aléatoire compris entre <randinf> <randsup>. Elle ne modifie en rien les bornes de l'ensemble des choix fixés par FIXEHASARD. Voir [HASARD](#) , [FIXEHASARD](#) , [PASAPAS](#) , [RANDDEC](#) .

Exemple: DONNE "nombre RANDOM 5 10

// est équivalent à : FIXEHASARD 5 10 DONNE "nombre HASARD

Rare exception dans la syntaxe francophone de Logoplus, la fonction RANDOM, très répandue dans les langages de programmation, est un emprunt aux versions LOGO anglo-saxonnes.

[Début du catalogue.](#)

### RAPPELLE :

Fonction RAPPELLE "nomfichier ou RAPPELLE [ chemin\"nomfichier ] - rappelle le contenu d'un fichier et ouvre si nécessaire une fenêtre de sélection des fichiers si le fichier demandé ne se trouve pas dans le répertoire courant. Les guillemets sont ignorés dans le chemin d'accès à l'emplacement physique de la mémoire (disque dur, mémoire flash...). Voir [DONNE / SOIT](#) , [SAUVE](#)

Exemples:

DONNE "MaDonnée RAPPELLE "fichier

ou:

DONNE "MaDonnée RAPPELLE [C:\Program Files\Education\Logoplus\"fichier]

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### RECALE :

Procédure RECALE <argnum> - fait reculer la tortue de <argnum> pas à l'écran. Voir [AVANCE](#), , [DEGRE](#), [LIEU](#), [LIMITES?](#), [PASAPAS](#), [MODEDEGRE](#), [MODERADIAN](#), [RADIAN](#), [TOURNEDROITE](#), [TOURNEGAUCHE](#). Abréviation: RE

Exemple: `RECALE 50` fera reculer la tortue de 50 pas à l'écran.

[Début du catalogue.](#)

### RECALELUTIN :

Procédure RECALELUTIN <lutin> <arg num> - fait reculer <lutin> de <arg num> pas à l'écran. Voir [AVANCELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTINXV](#), [LUTINVY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#), [MODERADIAN](#).

Exemple: `RECALELUTIN "lutin1 50`

fera reculer "lutin1" de 50 pas à l'écran par rapport à sa position actuelle;  
et:

`RECALELUTIN [ "lutin1 "lutin2 ] 50`

fera reculer "lutin1 et "lutin2 de 50 pas à l'écran par rapport à leur position actuelle respective. "lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

### REEL :

Fonction REEL <liste>, où <liste> est une liste vectorielle de deux items [<arg\_num\_part\_réelle> <arg\_num\_part\_imag>] - retourne comme résultat le nombre réel <arg\_num\_part\_réelle>. Voir [COMPLEXE](#), [CONJUG](#), [CPLXADD](#), [CPLXPROD](#), [IMAGE](#).

Exemple:

`ECL REEL [1 (-5)]` (qui correspond au nombre complexe 1-5i)  
verra s'afficher le nombre 1.

Remarque: Dans l'exemple, la primitive `REEL` est équivalente au code : `PREMIER [1 (-5)]`

## Début du catalogue.

### REMPPLACE :

Procédure REMPLACE [liste] [item] [pos] - remplace l'argument situé à la position indiquée dans la liste [pos]. [pos] ne doit contenir que des arguments numériques directement utilisables ( nombres ou variables) , sans quoi LOGO déclenchera une erreur en exécution. La liste [pos] ne sera calculée que si elle figure dans l'appel de la procédure. Si l'argument situé par [pos] se réfère à un item inexistant, une erreur sera aussi déclenchée. Unique primitive à nécessiter trois arguments, elle permet de gérer une liste comme s'il s'agissait d'un tableau. Voir [DERNIER](#), [ITEM](#), [LISTE](#), [INSERE](#), [LISTEVARS](#), [MEMBRE?](#), [OCCURRENCES](#), [PASAPAS](#), [PERMUTATIONS](#), [PREMIER](#), [SAISIS](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [TRIEITEMS](#) .

### Exemple:

```
DONNE "MaListe [ un deux [ 1 2 3 ] quatre ]
```

```
REMPPLACE :MaListe [un] [3 1]
```

```
ECL :MaListe
```

occasionnera la transformation du 3ième item (ici: [1 2 3]) pour obtenir la liste [ un deux [ un 2 3 ] quatre ] .

## Début du catalogue.

### [Vers l'index des primitives >>](#)

### REPETE :

Primitive REPETE <argnum> <primitive> ou REPETE <argnum> [prim1 prim2 ... ] - déclenche la répétition d'une primitive ou d'un groupe de primitives <argnum> fois. Si <argnum> n'est pas une valeur entière, c'est la partie entière qui sera prise en compte pour déterminer le nombre de fois à accomplir la primitive (ou la liste de primitives). <argnum> doit être strictement supérieur à zéro pour que LOGO puisse accomplir la boucle REPETE. Voir [BOUCLE](#), [PASAPAS](#) .

Exemples: REPETE 5 AVANCE 10 fera avancer la tortue de 50 pas à l'écran.

et:

```
REPETE (10/3) [ AVANCE 10 TOURNEDROITE 90]
```

Cette ligne sera accomplie 3 fois puisque la partie entière de 10/3 (=3,333) est 3.

Attention, écrire :

```
REPETE (4/5) ECRISLIGNE "Bonjour!"
```

revient à écrire :

```
REPETE 0 ECRISLIGNE "Bonjour! "
```

ce qui signifie que le mot Bonjour! ne sera jamais écrit.

De même, la ligne :

**REPETE** -4 **ECRISLIGNE** "Bonjour! " ne sera jamais exécutée.

[Début du catalogue.](#)

### REPONDRE :

Primitive REPONDRE <expression> - occasionne l'arrêt de la procédure en cours avec l'obligation de rendre la valeur de <expression> comme résultat. La procédure en cours devient alors une fonction.

Voir [POUR](#) , [FIN](#) , [FVECT](#) , [PASAPAS](#) , [RETOURNE](#) , [STOP](#) . Abréviation : **RENDS** .

Exemple:

```
POUR périmètre1 : longueur :largeur  
EFFACETEXTE  
REPONDRE 2 * (:longueur + :largeur)  
FIN
```

[Début du catalogue.](#)

### RESTE :

Fonction RESTE <argnum1> <argnum2> - calcule le reste de la division de <argnum1> par <argnum2>. <argnum2> doit être différent de zéro. Equivaut à écrire <argnum1> % <argnum2> .

Voir [ [+](#) , [-](#) , [\\*](#) , [/](#) ] , [DEC](#) , [DIFFERENCE](#) , [DIV](#) , [FVECT](#) , [INC](#) , [MAX?](#) , [MIN?](#) , [PASAPAS](#) , [PRODUIT](#) , [QUOTIENT](#) , [SOMME](#) , [RACINE2](#) , [RACINE3](#) .

Exemple:

```
ECRIS RESTE 5 2          rendra 1 comme résultat.  
// éq ECRIS 5 % 2 ou ECRIS 5 MOD 2
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### RETOURNE :

Primitive RETOURNE - occasionne l'arrêt de la procédure en cours et poursuit son déroulement à l'endroit où la procédure précédente avait été laissée au moment de son appel. Son usage est indispensable en cas d'appel récursif avec critère d'arrêt. Voir [PASAPAS](#) , [POUR](#) , [FIN](#) , [REPONDRE](#) , [RETOURNE](#) , [STOP](#) .

Exemple de programme récursif :

POUR arbre :longueur

TESTE :longueur < 10 /\* Critère d'arrêt \*/

SIVRAI RETOURNE /\* Le programme peut continuer dans la suite de la procédure appellante. \*/

SIFAUX [

AVANCE :longueur TOURNEGAUCHE :angle /2

arbre :longueur \* 0,7 Tournedroite :angle

arbre :longueur \* 0,7 TOURNEGAUCHE :angle /2

LEVECRAYON REcule :longueur BAISSecRAYON

]

FIN

### Début du catalogue.

### REVEILLE :

Procédure REVEILLE <lutin> - effectue la mise en route de <lutin> sur l'écran si sa vitesse n'est pas nulle. <lutin> reste toutefois visible. Voir [AVANCELUTIN](#), [REculeLUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#), [LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) .

Exemples: REVEILLE "lutin1

ou:

REVEILLE [ "lutin1 "lutin2 ]

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

### Début du catalogue.

### ROTATION :

Fonction ROTATION <arg> <décalage> , où <arg> est une liste ou un mot et <décalage> un entier positif ou négatif - fournit un résultat de même type que <arg> (une liste ou un mot) qui contient les mêmes items (ou caractères), mais décalés d'autant de positions indiquées par <décalage>, dans le sens horaire (si <décalage> positif), dans le sens antihoraire sinon. Si <décalage> est plus grand que le cardinal de <arg>, la rotation sera effectuée au modulo de cardinal de <arg>. Un décalage nul laisse la liste inchangée . Voir [DERNIER](#), [INSERE](#) , [ITEM](#), [LISTE](#) , [LISTEVARS](#) , [OCCURRENCES](#) , [PREMIER](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [SAISIS](#), [REMPPLACE](#), [SCINDE](#), [SWAP](#), [TRIEITEMS](#) , [VICEVERSA](#).

Abrév. ROT .

Exemple: ECL ROTATION [a [b c ] d e ] (-2)

occasionnera l'affichage de [d e a [b c ]].

## Début du catalogue.

### RVB :

Fonction RVB [ Cr Cv Cb] où Cr, Cv et Cb sont respectivement les composantes rouge, verte et bleue des couleurs fondamentales. Cr, Cv et Cb sont des nombres ou des variables qui représentent des nombres compris entre 0 et 255 (\*). RVB fournit en résultat un nombre entier long qui correspond à une couleur dans le code Windows. Voir [COULEURFOND](#) , [COULEURTORTUE](#) , [PASAPAS](#) .

(\*) Si l'un des trois arguments de la liste est supérieur à 255 ou négatif, il sera recadré sur l'intervalle demandé, au modulo positif à 255 près.

Exemple: [COULEURTORTUE](#) RVB [0 255 0]

## Début du catalogue.

### SAISIS :

Fonction SAISIS [liste] [pos] - fournit en résultat l'item contenu dans [liste] situé à la position indiquée dans la liste [pos]. [pos] ne doit contenir que des arguments numériques directement utilisables ( nombres ou variables), sans quoi LOGO déclenchera une erreur en exécution. La liste [pos] ne sera calculée que si elle figure dans l'appel de la procédure. Elle est équivalente à la fonction ITEM mais permet d'accéder à chaque item d'une liste comme s'il s'agissait d'un tableau. Si l'argument situé par [pos] se réfère à un item inexistant, une erreur sera aussi déclenchée. Voir [CHOIX](#) , [DERNIER](#), [INSERE](#) , [ITEM](#) , [LISTE](#), [LISTEVARS](#) , [MEMBRE?](#), [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#), [REPLACE](#) , [SAUFPREMIER](#), [SAUFDERNIER](#) , [TRIEITEMS](#) .

Exemple: SAISIS [ a b c d e ] [4] rendra "d en résultat (équivalent à [ITEM](#) 4 [ a b c d e ] ).  
SAISIS [ 10 20 [1 3 5] 30 40 ] [3 2] rendra 3 en résultat.

**A retenir** : Il est quelquefois possible d'utiliser des noms de données à l'intérieur d'une liste, pour rappeler leur contenu (ou valeur) ensuite. Dans ce cas, il est nécessaire de ne pas séparer par un espace le prédicat du nom de cette donnée ( : ) du reste de ce nom, ce qui aurait à la fois pour conséquence de créer deux items et de rendre le nom de la donnée illisible. De même, ne pas hésiter à utiliser des parenthèses pour séparer des items numériques qui commencent par un signe négatif. Préférer : [SOIT](#) "maliste [ :item1 :item2 :item3 ] à : [SOIT](#) "maliste [ : item1 : item2 : item3 ]

## Début du catalogue.

### SAUFDERNIER :

Fonction SAUFDERNIER [arg] - fournit en résultat la liste [arg] privée de son dernier item. Si un mot est passé à cette fonction en argument, une liste vide sera retournée en résultat. Cette primitive agit également sur les mots. Voir [CHOIX](#) , [DERNIER](#), [INSERE](#) , [ITEM](#), [LISTE](#), [LISTEVARS](#) , [MEMBRE?](#), [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#), [REPLACE](#) ,

[SAISIS](#) , [SAUFPREMIER](#) , [TRIEITEMS](#) , [xx || ...](#) . Abréviation: **SD** .

Exemple: **ECRIS SAUFDERNIER** [un deux trois] rendra un deux comme résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### **SAUFDERNIERELETTRE :**

Fonction SAUFDERNIERELETTRE "arg - fournit en résultat le mot "arg privé de sa dernière lettre. On peut la remplacer éventuellement par la primitive [SAUFDERNIER](#) . Voir [CHOIX](#) , [DERNIERELETTRE](#) , [MOT](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIERELETTRE](#) , [SAUFPREMIERELETTRE](#) , [SEPRE](#) , [TRIECAR](#) , [xx || ...](#) . Abréviation: **SDL** .

Exemple: **ECRIS SAUFDERNIERELETTRE** "tables  
rendra "table comme résultat.

[Début du catalogue.](#)

### **SAUFPREMIER :**

Fonction SAUFPREMIER [arg] - fournit en résultat la liste [arg] privée de son premier item. Si un mot est passé à cette fonction en argument, une liste vide sera retournée en résultat. Cette primitive agit également sur les mots. Voir [CHOIX](#) , [DERNIER](#) , [INSERE](#) , [ITEM](#) , [LISTE](#) , [LISTEVARS](#) , [MEMBRE?](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIER](#) , [REMPLECE](#) , [SAISIS](#) , [SAUFDERNIER](#) , [TRIECAR](#) , [TRIEITEMS](#) , [xx || ...](#) . Abréviation: **SP** .

Exemple: **ECRIS SAUFPREMIER** [ un deux trois] rendra deux trois comme résultat.

[Début du catalogue.](#)

### **SAUFPREMIERELETTRE :**

Fonction SAUFPREMIERELETTRE "arg - fournit en résultat le mot "arg privé de sa première lettre. On peut la remplacer éventuellement par la primitive [SAUFPREMIER](#) . Voir [CHOIX](#) , [DERNIERELETTRE](#) , [MOT](#) , [OCCURRENCES](#) , [PASAPAS](#) , [PERMUTATIONS](#) , [PREMIERELETTRE](#) , [SAUFDERNIERELETTRE](#) , [SEPRE](#) , [TRIECAR](#) , [xx || ...](#) . Abrév: **SPL**

Exemple: **ECRIS SAUFPREMIERELETTRE** "ables  
rendra "ables comme résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## SAUVEECRAN:

Procédure SAUVEECRAN "image", effectue l'enregistrement sur support-mémoire de l'image de l'écran de la tortue. Une fenêtre Explorateur s'ouvre et permet de l'enregistrer à l'endroit voulu.

Exemple: SAUVEECRAN "écran tortue".

[Début du catalogue.](#)

## SAUVE :

Procédure SAUVE "nom donnée" ou SAUVE [ chemin\ "nom donnée " ] : sauvegarde une donnée LOGO et ouvre si nécessaire une fenêtre de sélection des fichiers. "nom fichier" doit avoir été auparavant défini par la primitive DONNE. Les guillemets qui figurent dans la liste après la barre oblique sont ignorés dans le nom du fichier sur l'emplacement physique de la mémoire (disque dur, mémoire flash ...). Voir [DONNE / SOIT](#) , [RAPPELLE](#) .

Exemple:

DONNE "MaDonnée1 5

DONNE "MaDonnée2 10

SAUVE "MaDonnée1

SAUVE [C:\Program Files\Education\Logoplus\ "MaDonnée2 ]

[Début du catalogue.](#)

## SCINDE :

Fonction SCINDE <arg> <rang> où <arg> est soit une liste, un mot, un nombre ou un booléen et <rang> un entier naturel (compris entre 1 et le cardinal de <arg>) - fournit en résultat une liste de deux items formés pour le premier d'une liste des (<rang>-1) premiers items de <arg>, et des <rang> items pour le deuxième. Voir [CHOIX](#) , [DERNIER](#), [INSERE](#) , [ITEM](#), [LISTE](#) , [LISTE?](#) , [LISTEVAR](#) , [OCCURRENCES](#) , [PREMIER](#), [REPLACE](#), [ROTATION](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [SAISIS](#), [SWAP](#), [TRIEITEMS](#) , [VICEVERSA](#) .

n.b. :

- Lorsque le rang se fait à partir du deuxième item et que le premier item de la liste n'est pas lui-même une liste, il sera laissé tel quel. Même chose lorsque le rang se fait à partir du dernier item de la liste : cet item ne sera pas encapsulé dans une liste. Cette remarque oblige cependant le programmeur à vérifier le type des deux items obtenus par scission avant de les utiliser.
- Lorsque le rang n'est pas positionné sur l'un des items de la liste, une liste vide sera placée soit avant (le rang est de valeur nulle ou négative) soit après la liste entière (le rang a alors une valeur trop grande) .

### Exemple 1 :

```
DONNE "resu SCINDE [123 456 [78 90 12] 345 67 ] 2.
```

Ainsi, la variable :résu équivaut à [[123 456 [78 90 12]] [ 345 67 ]].

### Exemple 2 :

```
DONNE "liste1 [ a b c d e f ]
DONNE "résultat1 SCINDE :liste1 2 // :résultat1 <-> [a [b c d e f]]
DONNE "résultat1 SCINDE :liste1 6 // :résultat1 <-> [[a b c d e] f]
DONNE "résultat1 SCINDE :liste1 0 // :résultat1 <-> [[[a b c d e f]]]
DONNE "résultat1 SCINDE :liste1 7 // :résultat1 <-> [ [a b c d e f] []]
DONNE "nombre1 12345
DONNE "résultat2 SCINDE :nombre1 2 // :résultat2 <-> [1 2345]
DONNE "résultat2 SCINDE :nombre1 5 // :résultat2 <-> [1234 5]
DONNE "résultat2 SCINDE :nombre1 0 // :résultat2 <-> [[]][12345]]
DONNE "résultat2 SCINDE :nombre1 6 // :résultat2 <-> [ [12345] []]
DONNE "mot1 "abcdefg
DONNE "résultat3 SCINDE :mot1 2 // :résultat3 <-> [a bcdefg]
DONNE "résultat3 SCINDE :mot1 7 // :résultat3 <-> [abcdef g]
DONNE "résultat3 SCINDE :mot1 0 // :résultat3 <-> [[[abcdefg]]]
DONNE "résultat3 SCINDE :mot1 8 // :résultat3 <-> [ [abcdefg] []]
DONNE "couleur1 BLEUE
DONNE "résultat4 SCINDE :couleur1 2 //:résultat4 <-> [1 6711680]
DONNE "résultat4 SCINDE :couleur1 (-1) //:résultat4 <-> [[]][16711680]]
```

### Début du catalogue.

### SECANTE :

Fonction SECANTE <arg num> - calcule la valeur de la sécante <arg num>. Voir [ARCCSC](#) , [ARCSEC](#) , [COSEC](#) .

### Exemple:

```
ECRIS SECANTE 0,5          donnera 1,00003807838574 en résultat.
```

### Début du catalogue.

### SECH :

Fonction SECH <arg num> - calcule la valeur de la sécante hyperbolique de <arg num>. Voir [ARGCOSH](#) , [ARGCOth](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGSINH](#) , [ARGTANH](#) , [COSH](#) , [CSCH](#) , [SINH](#) , [TANH](#) .

Exemple:

ECRIS SECH 0,5

donnera 0,999961924030716 en résultat.

[Début du catalogue.](#)

SEGMENT :

Procédure SEGMENT [arg1] [arg2] - trace un segment sur l'écran de la tortue entre les points fixés par les listes [arg1] et [arg2] en utilisant la couleur de la tortue. [arg1] et [arg2] sont des listes de type LIEU (rappel : [Xpos Ypos]). Voir [ARC](#), [FONDECRAN](#), [ELLIPSE](#)

Exemple: SEGMENT [ 5 10 ] [ 20 40 ]

[Début du catalogue.](#)

SEGMENT3D :

Fonction SEGMENT3D [X1 Y1 Z1] [X2 Y2 Z2] - trace un segment entre les points fixés par les listes [coordspoint1] et [coordspoint2] sur l'écran du grapheur de fonctions et retourne éventuellement la longueur du segment. Les listes [coordspoint1] et [coordspoint2] doivent avoir toutes les deux trois items immédiatement calculables (i.e : sans signe opératoire). Voir [INITGRAPH](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [GRAPHEUR](#), [PLOT](#), [SURFACE](#), [SYSLOT](#), [SYSLOTPOL](#).

Exemple:

DONNE "liste1 [ 0 100 100 ] DONNE "liste2 [ 100 (-100) (-100) ]

INITGRAPH FAUX GRAPHEUR FAUX

SEGMENT3D [100 90 50 ] [ 100 (-200) 90 ]

SEGMENT3D :liste1 [ 90 150 (-90) ] // ou ECL SEGMENT3D :liste1 [90 150 (-90) ]

SEGMENT3D [100 120 90 ] :liste2 // ou DONNE "longueur SEGMENT3D :liste1 [90 150 (-90) ]

SEGMENT3D :liste1 :liste2

GRAPHEUR VRAI

[Début du catalogue.](#)

SENSHORAIRE :

Procédure SENSHORAIRE - définit la direction prise par la tortue en fonction de son cap. Lorsque SENSHORAIRE est sélectionné, le cap de la tortue va en s'accroissant dans le sens horaire. Voir [CAP](#), [FIXECAP](#), [SENSTRIGO](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#), [MODERADIAN](#), [PASAPAS](#).

[Début du catalogue.](#)

SENSTRIGO :

Procédure **SENSTRIGO** - définit la direction prise par la tortue en fonction de son cap. Lorsque SENSTRIGO est sélectionné, le cap de la tortue va en s'accroissant dans le sens trigonométrique. Voir [CAP](#), [FIXECAP](#), [SENSHORAIRE](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#), [MODERADIAN](#), [PASAPAS](#).

[Début du catalogue.](#)

**SEPARE** :

Fonction SEPARE <arg> - fournit en résultat la liste des lettres du mot <arg>, séparées par un espace. Si <arg> est un nombre, la liste sera composée des chiffres de <arg> séparés par un espace. Lorsque <arg> est une liste, les items seront atomisés en lettres, en chiffres ou en signes tout en conservant les délimiteurs des sous-listes s'il y en a. Voir [DERNIERELETTRE](#), [OCCURRENCES](#), [PERMUTATIONS](#), [PREMIERELETTRE](#), [SAUFPREMIERELETTRE](#), [SAUFDNIERELETTRE](#), [MOT](#), [TRANSFORME](#), [TRIEITEMS](#).

Exemples:

**ECRISLIGNE SEPARE** "Bonjour! rendra comme résultat la liste [ B o n j o u r ! ].

**ECRISLIGNE SEPARE** 123,456 rendra comme résultat la liste [ 1 2 3 , 4 5 6 ].

**ECRISLIGNE SEPARE** [ janvier février mars avril ]

rendra comme résultat la liste [ j a n v i e r f é v r i e r m a r s a v r i l ].

**ECRISLIGNE SEPARE** [ janvier [ février mars ] avril ]

rendra comme résultat la liste [ j a n v i e r [ f é v r i e r m a r s ] a v r i l ]

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**SI** :

Primitive SI <condition> <expression>, où <condition> est une expression booléenne contenant au moins deux arguments et un signe de comparaison. Voir [TESTE](#), [SIFAUX \(SINON\)](#), [SIVRAI](#), [TANTQUE](#), [BOOLEEN](#).

Exemple: **SI** : MonNombre < 5 **TOURNEGAUCHE** 45.

[Début du catalogue.](#)

**SIFAUX (SINON)** :

Primitive SIFAUX <primitive> ou SIFAUX [prim1 prim2...] - effectue une ou plusieurs primitives suite au résultat d'un test. Equivalent à SINON. Voir [PASAPAS](#), [TESTE](#), [SI](#), [SIVRAI](#), [TANTQUE](#), [BOOLEEN](#).

Exemple: **TESTE** : MonNombre < 5

**SIFAUX AVANCE** :MonNombre. // équivalent à **SINON AVANCE** :MonNombre

[Début du catalogue.](#)

### **SILENCE** :

Primitive **SILENCE** - interrompt le mode trace et les calculs en cours ne sont alors plus affichés.

Voir [PASAPAS](#) , [TRACE](#).

Exemple: **SILENCE**

[Début du catalogue.](#)

### **SINH** :

Fonction **SINH** <arg num> - calcule la valeur du sinus hyperbolique de <arg num>.

Voir [ARGCOSH](#) , [ARGCOth](#) , [ARGCSCH](#) , [ARGSECH](#) , [ARGsinh](#) , [ARGTANH](#) , [COSH](#) , [CSCH](#) , [SECH](#) , [TANH](#) .

Exemple:

**ECRIS SINH** 0,5                      donnera 0,521095305493747 en résultat.

[Début du catalogue.](#)

### **SINUS** :

Fonction **SINUS** <argnum> - calcule la valeur du sinus du nombre <argnum> (\*). Voir [COSINUS](#), [INTEGRALE](#) , [TAN](#) , [ARCTAN](#) , [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#), [ZEROS?](#) .

Abréviation: **SIN**

(\*) Si l'argument utilisé est supérieur à 360° (ou 2π) , il sera recadré au modulo à 360° (ou 2 π) près.

Exemple: **ECRIS SINUS** 30                      rendra la valeur 0,5.

[Début du catalogue.](#)

### **SITUATION** :

Fonction **SITUATION** - fournit en résultat la liste de la situation actuelle de la tortue sous la forme [XCOORD YCOORD CAP] . Voir [LIEU](#) , [VA](#) , [VERS](#) , [FIXECAP](#) , [RAMENE](#) , [XCOORD](#) , [YCOORD](#) , [CAP](#) , [FIXEX](#) , [FIXEXY](#) , [FIXEY](#) , [NETTOIE](#) , [PASAPAS](#) .

Exemple: **RAMENE SITUATION**

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### **SIVRAI** :

Primitive **SIVRAI** <primitive> ou **SIVRAI** [prim1 prim2 ...] - effectue une ou plusieurs primitives

suite au résultat d'un test. Voir [BOOLEEN](#) , [PASAPAS](#) , [SI](#) , [SIFAUZ \(SINON\)](#) , [TANTQUE](#) , [TESTE](#).

Exemple: **TESTE** : MonNombre < 5  
**SIVRAI RECULE** :MonNombre.

[Début du catalogue.](#)

### **SOMME** :

Fonction SOMME <arg num> où <arg num> est une liste de valeurs ou de variables numériques calcule la somme généralisée des termes inclus dans <arg num>. Voir [ [+ - \\* /](#) ], [DEC](#) , [DIFFERENCE](#) , [DIV](#) , [DIVISEURS](#) , [FVECT](#) , [INC](#) , [MAX?](#) , [MIN?](#) , [MOD](#) , [PASAPAS](#) , [PRODUIT](#) , [QUOTIENT](#) , [RESTE \(%\)](#) , [TRIEITEMS](#) .

Exemple:

**DONNE** "v -4 // ég : **SOIT** "v -4  
**ECRIS SOMME** [ 1 5 :v 3 ] rendra 5 comme résultat.  
( 1 + 5 - 4 + 3)

[Début du catalogue.](#)

### **SOURISX** :

Fonction SOURISX - fournit en résultat la valeur de l'abscisse de la souris sur la fenêtre de l'écran. Voir [PASAPAS](#) , [SOURIXY](#) .

Exemple: **SI** (( **SOURISX** < 0) **ET** ( **SOURISX** > 400 )) **TOURNEGAUCHE** 180

[Début du catalogue.](#)

### **SOURISY** :

Fonction SOURISY - fournit en résultat la valeur de l'ordonnée de la souris sur la fenêtre de l'écran. Voir [PASAPAS](#) , [SOURISX](#) .

Exemple: **SI** (( **SOURISY** < 0) **ET** ( **SOURISY** > 200 )) **TOURNEDROITE** 180

[Début du catalogue.](#)

### **SPOTS** :

Procédure SPOTS [ arg ] - occasionne l'illumination de chacun des spots de la fenêtre des capteurs optiques pendant le nombre de millisecondes indiqué sur sa position dans la liste [arg]. Comme la fenêtre des capteurs optiques en comporte 9, cette liste inclut 9 valeurs. Cette primitive permet, en réglant la durée d'activation des spots, de commander par exemple un robot. Si une autre primitive SPOTS est ensuite rencontrée, LOGO place cette primitive en attente pendant le temps

nécessaire jusqu'à ce que tous les spots s'éteignent. Pour que les temps d'allumage soient bien respectés, éviter cependant d'utiliser les primitives de gestion des lutins, gourmandes en temps de calcul. Voir [CACHESPOTS](#) , [MONTRESPOTS](#) .

Exemple: SPOTS [ 1000 24000 180 0 0 1200 0 0 500 ]

Cette consigne indique que le spot n°1 restera illuminé pendant 1000 millisecondes (soit 1 seconde), le spot n°2, 24 secondes, le spot n°3, 180 millisecondes etc... . Les spots 4, 5, 7 et 8 ayant des valeurs nulles resteront éteint pendant 24 secondes (valeur maximale attribuée au spot n°2).

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### STOP :

Procédure STOP - effectue l'arrêt instantané de la procédure en cours. Voir [PASAPAS](#) , [POUR](#) , [FIN](#) , [REPENDRE](#) , [RETOURNE](#) , [SUSPENDRE](#).

Exemple:

```
SOIT "MonNombre 4
TESTE : MonNombre >= 10 SIFAUZ STOP
```

[Début du catalogue.](#)

### STYLELETTRE :

Procédure STYLELETTRE <arg> - change la fonte des caractères affichés sur l'écran de la tortue. Voir [COULEURELETTRE](#) , [TAILLELETTRE](#) , [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) , [ORANGE](#) , [ROSE](#) , [VIOLETTE](#) .

Exemples: STYLELETTRE "Mistral

ou:

```
STYLELETTRE [Ms Sans Sérif]
```

Voir également la liste des codes [couleurs Windows](#).

[Début du catalogue.](#)

### SUPPREP :

Procédure SUPPREP [chemin] - supprime le répertoire spécifié par la liste [chemin]. Le répertoire doit être vide pour pouvoir être supprimé. Si le répertoire ne peut être supprimé ou n'existe pas, une boîte de dialogue le signalera. Voir [CHANGEREP](#) , [CREEREP](#) , [SAUVE](#) , [RAPPELLE](#) , [SUPPRIME](#)

Exemple: **SUPPREP** [C:\Education\LogoPlus\Jeux]

[Début du catalogue.](#)

### **SUPPRIME** :

Procédure SUPPRIME "madonnée ou SUPPRIME [ chemin\"madonnée] - efface sur le disque le fichier intitulé "madonnée et ouvre une fenêtre de sélection de fichier si nécessaire. Si le fichier ne peut être supprimé ou n'existe pas, une boîte de dialogue le signalera. Les guillemets sont ignorés pendant la recherche de la donnée à l'emplacement physique (disque dur, mémoire flash...).

Voir [SAUVE](#) , [RAPPELLE](#) , [CREEREP](#) , [CHANGEREPE](#) , [SUPPREP](#) .

Exemple:

```
DONNE "MaDonnée1 5 DONNE "MaDonnée2 10
SAUVE "MaDonnée1 SAUVE [C:\Program Files\Education\Logoplus\"MaDonnée2 ]
SUPPRIME "MaDonnée1
SUPPRIME [C:\Program Files\Education\Logoplus\"MaDonnée2 ]
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### **SURECRAN?** :

Fonction booléenne SURECRAN? - fournit en résultat la valeur booléenne [VRAI](#) si le curseur de la souris se trouve sur l'écran de la tortue, [FAUX](#) sinon.

Exemple: **TESTE** SURECRAN?

```
SIVRAI ECRISLIGNE [ Le curseur de la souris se trouve sur l'écran. ]
SIVRAI ECRISLIGNE [ Le curseur de la souris se trouve hors de l'écran. ]
```

[Début du catalogue.](#)

### **SURFACE** :

Fonction SURFACE [ liste\_x liste\_y nom\_fonction\_xy discret ] - affiche le graphique de la fonction numérique désignée par nom\_fonctionxy suivant les valeurs en x et y qui se trouvent dans les listes liste\_x et liste\_y et retourne éventuellement la valeur approchée de l'aire de cette surface (même si le paramètre discret est fixé à **VRAI**). La précision dépend du nombre de points à déterminer ainsi que de la rapidité des calculs. Le bloc nom\_fonction\_xy doit être défini dans l'éditeur et ne doit comporter que deux arguments numériques dans sa définition. Le paramètre booléen discret indique si la fonction doit être tracée selon une succession de points (**VRAI**) ou sous forme d'un maillage (**FAUX**). Voir [CHAMPVECT](#) , [INITGRAPH](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [GRAPHEUR](#), [PLOT](#) , [PLOTPOLE](#) , [INTEGRALE](#), [SEGMENT3D](#) , [SYSPLOTE](#), [SYSPLOTPOLE](#) , [ZEROS?](#) .

Exemple:

```
SOIT "lx LISTEVARS -10 10 40
SOIT "ly LISTEVARS -10 10 40
EFFTXT INITGRAPH FAUX MODEDEGRE
DONNE "l [ :lx :ly fxy VRAI ]
SURFACE :l // ou ECL SURFACE :l ; // ou DONNE "aire SURFACE :l
GRAPHEUR VRAI

POUR fxy :x :y
RENDS 5 * TANH (5* :x * :y )
FIN
```

[Début du catalogue.](#)

SUSPENDRE :

Procédure **SUSPENDRE** - primitive système utilisée pour le débogage d'un programme LOGO. Voir [PASAPAS](#) , [STOP](#) .

Exemple:

```
SOIT "MonNombre 4
TESTE : MonNombre <= 8 SIVRAI SUSPENDRE
```

[Début du catalogue.](#)

SWAP :

Fonction SWAP <arg> <rang> où <arg> est soit une liste, un mot, un nombre ou un booléen et <rang> un entier naturel (compris entre 1 et le cardinal de <arg>) - fournit un résultat de même type que <arg> (liste, mot ou nombre) formé des <rang> derniers items (chiffres ou caractères) de <arg> suivis par les (<rang>-1) premiers items (chiffres ou caractères). Voir [CHOIX](#) , [DERNIER](#), [INSERE](#) , [ITEM](#), [LISTE](#), [LISTE?](#) , [LISTEVARS](#) , [OCCURRENCES](#) , [PREMIER](#), [REPLACE](#), [ROTATION](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [SAISIS](#), [SCINDE](#), [TRIEITEMS](#) , [VICEVERSA](#) .

Exemple 1 : DONNE "resu SWAP [123 456 [78 90 12] 345 67 ] 3

Ainsi, la variable :résu équivaut à `[[78 90 12] 345 67 123 456]`.

n.b. :

Lorsque <arg> est une liste et que le rang se fait à partir du deuxième item alors que le premier item de la liste n'est pas lui-même une liste, il sera laissé tel quel. Même chose lorsque le rang se fait à partir du dernier item de la liste : cet item ne sera pas encapsulé dans une liste.

Exemple 2 : DONNE "resu SWAP [123 456 [78 90 12] 345 67 ] 2

La variable :résu équivaut à [456 [78 90 12] 345 67 123]

[Début du catalogue.](#)

### SYSLOT :

Fonction SYSLOT [[vecteurX][vecteurY][vecteurZ][couleurs]] - où <vecteurX>, <vecteurY>, <vecteurZ> et <couleurs> sont quatre listes vectorielles de mêmes dimensions chacune qui représentent respectivement les abscisses (<vecteurX>), les ordonnées (<vecteurY>) les profondeurs (<vecteurZ>) et les valeurs de couleur successives, sous forme nominale ou numérique, des points en coordonnées rectangulaires ; occasionne le tracé des points sur l'écran de la fenêtre du grapheur. Voir [CHAMPVECT](#) , [INITGRAPH](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [GRAPHEUR](#), [PLOTPOL](#) , [SEGMENT3D](#) , [SURFACE](#), [SYSLOTPOL](#) .

Exemple:

```
SYSLOT [[3,5 12 (-14) 27,5][[-18 0,75 109 (-45)][[-58 32,8 (-46,7) 18][grise 255 orange noire]]
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### SYSLOTPOL :

Fonction SYSLOTPOL [[vecteurX][vecteurY][vecteurZ][couleurs]] - où <vecteurX>, <vecteurY>, <vecteurZ> et <couleurs> sont quatre vecteurs de mêmes dimensions chacun qui représentent respectivement les abscisses (<vecteurX>), les ordonnées (<vecteurY>) les profondeurs (<vecteurZ>) et les valeurs de couleur successives, sous forme nominale ou numérique, des points en coordonnées polaires ; occasionne le tracé des points sur l'écran de la fenêtre du grapheur. Voir [CHAMPVECT](#) , [INITGRAPH](#), [GRAPHE](#), [GRAPH2D](#), [GRAPH3D](#), [GRAPHEUR](#), [PLOTPOL](#) , [SEGMENT3D](#) , [SURFACE](#), [SYSLOT](#) .

Exemple:

MODEDEGRE

```
SYSLOTPOL [[24,8 (-34) 17,3 (-21)][[81 70,5 9 54][[85 28,3 67 40][ 255 rouge grise verte]]
```

[Début du catalogue.](#)

### TAILLELETTRE :

Procédure TAILLELETTRE <argnum> - change la hauteur des lettres affichées sur l'écran de la tortue. Voir [COULEURLETTRE](#) , [STYLELETTRE](#) , [BLANCHE](#) , [VERTE](#) , [JAUNE](#) , [ROUGE](#) , [BLEUE](#) , [NOIRE](#) , [TILLEUL](#) , [EAU](#) , [FUCHSIA](#) , [SARCELLE](#) , [GRISE](#) , [OLIVE](#) , [MARRON](#) , [ARGENT](#) ,

[ORANGE](#), [ROSE](#), [VIOLETTE](#). Voir également la liste des codes [couleurs Windows](#).

Exemple: `TAILLELETTRE 12`

[Début du catalogue.](#)

**TAN** :

Fonction TAN <argnum> - calcule la valeur de la tangente du nombre <argnum> (\*). Voir [COSINUS](#), [INTEGRALE](#), [SINUS](#), [ARCTAN](#), [ARCCOT](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#), [MODERADIAN](#).

(\*) Si l'argument utilisé est supérieur à 360° (ou 2π), il sera recadré au modulo à 360° (ou 2π) près.

Exemple:

`ECRIS TAN 60` rendra 1,732050808.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**TANH** :

Fonction TANH <arg num> - calcule la valeur de la tangente hyperbolique de <arg num>.

Voir [ARGCOSH](#), [ARGCOth](#), [ARGCSCH](#), [ARGSECH](#), [ARGSINH](#), [ARGTANH](#), [COSH](#), [CSCH](#), [SECH](#), [SINH](#).

Exemple:

`ECRIS TANH 0,5` donnera 0,46211715726001 en résultat.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**TANTQUE (TANT QUE)** :

Primitive TANTQUE <condition> <primitive> ou TANTQUE <condition> [prim1 prim2 ... ] - déclenche la répétition d'une primitive ou d'un groupe de primitives tant que <condition> est vérifiée. Cette primitive peut se décliner en deux parties (i.e. TANT QUE) pour se rapprocher du langage naturel.

Voir [PASAPAS](#), [SI](#), [SIVRAI](#), [SIFAUX \(SINON\)](#), [TESTE](#).

Exemple 1:

```
SOIT "i 5 // éq : DONNE "i 5
TANTQUE (:i > 0) [ECRISLIGNE :i DEC :i ]
fera afficher le contenu de la variable i tant qu'elle restera positive.
```

Exemple 2:

```
SOIT "i 0 // éq : DONNE "i 0
TANT QUE (:i <= 5) [ECRISLIGNE :i INC :i ]
```

fera afficher le contenu de la variable i tant qu'elle restera inférieure ou égale à 5.

[Début du catalogue.](#)

TEMPO :

Procédure TEMPO <pulsation> - en relation avec la primitive JOUE, fixe le nombre de pulsations à la minute de la mélodie à faire entendre. <pulsation> est un nombre compris entre 40 et 208. Par convention, la noire représente l'unité de cette pulsation. Voir [JOUE](#) .

Exemple: TEMPO 60

fixe la pulsation à 60 battements par minute.

[Début du catalogue.](#)

TEMPS :

Fonction TEMPS - fournit en résultat la liste contenant l'heure, les minutes, les secondes et les millièmes de seconde en cours. Voir [DATE](#) .

Exemple:

ECRISLIGNE TEMPS                      affichera [ 12 18 7 632 ] .

[Début du catalogue.](#)

TESTE :

Primitive de test TESTE <condition>, où <condition> est une expression contenant au moins deux arguments et un signe de comparaison. Voir [PASAPAS](#) , [SI](#) , [SIVRAI](#) , [SIFAUX \(SINON\)](#) , [TANTQUE](#) , [BOOLEEN](#) .

Exemple:

SOIT "MonNombre 4  
TESTE : MonNombre < 5 SIVRAI STOP

[Début du catalogue.](#)

TOPS? :

[K8055/VM110](#)> Fonction TOPS? :compteur - lit le contenu du compteur d'impulsions (:compteur = 1 ou 2) et retourne une valeur numérique comprise entre 0 et 255. Voir [INIT\\_COMPTEUR](#) , [FIXEREBOND](#) .

Exemple:

CARTEUSB "VM110

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### TORTUES? :

Fonction **TORTUES?** - fournit en résultat la liste du rang (de 1 à 16 ) des tortues activées par la primitive OBEIS. Voir [OBEIS](#), [FORME](#) .

Exemple: **ECRISLIGNE TORTUES?**

[Début du catalogue.](#)

### TOUCHE? :

Fonction booléenne **TOUCHE?** - fournit en résultat la valeur booléenne **VRAI** si une touche du clavier est enfoncée, **FAUX** sinon. Voir [CLIC?](#) , [SURECRAN?](#) .

Exemple: **ECRISLIGNE TOUCHE?**

rendra **VRAI** si on appuie sur l'une des touches du clavier au moment de son appel.

[Début du catalogue.](#)

### TOURNEDROITE :

Procédure **TOURNEDROITE** <argnum> - fait tourner la tortue de <argnum> degrés vers la droite (\*). Voir [AVANCE](#), [COMPAS](#) , [MODEDEGRE](#), [MODERADIAN](#) , [PASAPAS](#) , [RECULE](#), [SENSHORAIRE](#), [SENSTRIGO](#), [TOURNEGAUCHE](#). Abréviation: TD

Exemple: **TOURNEDROITE 50** fera tourner la tortue de 50 degrés sur sa droite.

(\*) Si l'argument utilisé est supérieur à 360° (ou  $2\pi$ ) , il sera recadré au modulo à 360° (ou  $2\pi$ ) près.

[Début du catalogue.](#)

### TOURNEGAUCHE :

Procédure **TOURNEGAUCHE** <argnum> - fait tourner la tortue de <argnum> degrés vers la gauche (\*). Voir [AVANCE](#), [COMPAS](#) , [PASAPAS](#) , [MODEDEGRE](#), [MODERADIAN](#) , [RECULE](#), [SENSHORAIRE](#), [SENSTRIGO](#) , [TOURNEDROITE](#) . Abréviation: TG

Exemple: **TOURNEGAUCHE 50** fera tourner la tortue de 50 degrés sur sa gauche.

(\*) Si l'argument utilisé est supérieur à 360° (ou  $2\pi$ ) , il sera recadré au modulo à 360° (ou  $2\pi$ ) près.

## [Début du catalogue.](#)

### TOURNELUTINDROITE :

Procédure TOURNELUTINDROITE <lutin> <arg num> - effectue la rotation de <lutin> de <arg num> (radians ou degrés) vers la droite (\*). \_Voir [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [AVANCELUTIN](#) , [RECULELUTIN](#) , [TOURNELUTINGAUCHE](#) , [ENDORS](#) , [REVEILLE](#) , [CACHELUTIN](#) , [MONTRELUTIN](#) , [FIXELUTINX](#) , [FIXELUTINY](#) , [COLORIELUTIN](#) , [COULEURLUTIN?](#) , [VITESSE](#) , [POSITIONLUTIN?](#) , [LUTINX?](#) , [LUTINY?](#) , [LUTINVX](#) , [LUTINVY](#) , [CAPLUTIN](#) , [CAPLUTIN?](#) , [VELOCITE?](#) , [COLLISIONS?](#) . Abréviation: **TLD**

(\*) Si l'argument utilisé est supérieur à 360° (ou 2 $\pi$ ) , il sera recadré au modulo à 360° (ou 2  $\pi$ ) près.

Exemples: **TOURNELUTINDROITE** "lutin1 50

ou:

**TOURNELUTINDROITE** [ "lutin1 "lutin2 ] 50 .

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

### TOURNELUTINGAUCHE :

Procédure TOURNELUTINGAUCHE <lutin> <arg num> - effectue la rotation de <lutin> de <arg num> (radians ou degrés) vers la gauche (\*). \_Voir [DEGRE](#) , [RADIAN](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [AVANCELUTIN](#) , [RECULELUTIN](#) , [TOURNELUTINDROITE](#) , [ENDORS](#) , [REVEILLE](#) , [CACHELUTIN](#) , [MONTRELUTIN](#) , [FIXELUTINX](#) , [FIXELUTINY](#) , [COLORIELUTIN](#) , [COULEURLUTIN?](#) , [VITESSE](#) , [POSITIONLUTIN?](#) , [LUTINX?](#) , [LUTINY?](#) , [LUTINVX](#) , [LUTINVY](#) , [CAPLUTIN](#) , [CAPLUTIN?](#) , [VELOCITE?](#) , [COLLISIONS?](#) . Abréviation: **TLG**

Exemples: **TOURNELUTINGAUCHE** "lutin1 50

ou:

**TOURNELUTINGAUCHE** [ "lutin1 "lutin2 ] 50 .

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

(\*) Si l'argument utilisé est supérieur à 360° (ou 2 $\pi$ ) , il sera recadré au modulo à 360° (ou 2  $\pi$ ) près.

[Début du catalogue.](#)

### TOUTES :

Procédure TOUTES - active les 16 tortues gérées par Logoplus. Cette primitive est équivalente à la consigne : **OBEIS** [ 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 ] .

Voir [AUCUNE](#) , [PASAPAS](#) .

Exemple: TOUTES

[Début du catalogue.](#)

TRACE :

Procédure TRACE - autorise l'affichage des consignes en cours de traitement dans l'afficheur de textes. Voir [PASAPAS](#) , [SILENCE](#).

Exemple: TRACE

[Début du catalogue.](#)

TRAIT :

Procédure TRAIT <arg num> - fixe la largeur du trait à <arg num>. A l'origine fixé à 1, plus la valeur donnée en argument est grande, plus la largeur du tracé de la tortue sera grande. Cette procédure peut rendre inopérante les fonctions CAPTEURAVANT?, CAPTEURDROIT? et CAPTEURGAUCHE? si cet argument est trop important. Voir [CAPTEURAVANT?](#) , [CAPTEURDROIT?](#) , [CAPTEURGAUCHE?](#) , [PASAPAS](#) .

Exemple: TRAIT 5

[Début du catalogue.](#)

TRANSFORME :

Fonction TRANSFORME <arg> - trois cas possibles selon le type de <arg> -

- 1) [arg] est une liste TRANSFORME renverra "arg en résultat sous forme d'un mot, si la liste d'origine ne contient qu'un seul item, sinon elle laisse la liste inchangée.
- 2) "arg est un mot TRANSFORME renverra [arg] en résultat sous forme d'une liste ne contenant qu'un seul item.
- 3) arg est une valeur numérique TRANSFORME renverra "arg en résultat sous forme d'un mot.

TRANSFORME s'utilise avec l'instruction DONNE exclusivement lorsque son argument est numérique. Par exemple, on écrira: DONNE "n TRANSFORME 5 et :n aura pour contenu "5 . Lorsqu'on écrit TRANSFORME :m, il faudra alors rappeler la variable :m pour connaître son contenu. Voir [PASAPAS](#) .

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## TRANSLATE :

Fonction TRANSLATE <arg> <base> - fournit en résultat la valeur de <arg> exprimée dans la base indiquée. <arg> et <base> doivent être exprimés soit en base 10 soit en base 16 (Hexa \$). Si <base> est positive, la conversion se fera de la base 10 (ou 16) vers la base indiquée. Par contre, si <base> est négative, cela indique que la conversion se fera de la base indiquée vers la base 10 .

Voir [HEXA?](#) .

Exemples:

ECRISLIGNE TRANSLATE \$AE5 3 occasionnera l'affichage de 10211022 (en base 3).

ECRISLIGNE TRANSLATE \$B2F -16 occasionnera l'affichage de 2863 (en base 10).

### [Début du catalogue.](#)

## TRIECAR :

Fonction TRIECAR <mot> {"</>} - fournit en résultat l'anagramme de <mot> formé des lettres rangées suivant l'ordre croissant ou décroissant de leur code ASCII. Remarque : les signes < ou > doivent être obligatoirement précédés des guillemets. Voir [DERNIERELETTRE](#) , [MOT](#) , [MOT?](#) , [PREMIERELETTRE](#) , [SAUFPREMIERELETTRE](#) , [SAUFDERNIERELETTRE](#) , [SEPRE](#) , [xx || ...](#) .

Exemples:

EFFTXT

SOIT "m TRIECAR "0123456789 "> rendra 9876543210 en résultat.

ECL :m

SOIT "m TRIECAR "9876543210 "< rendra 0123456789 en résultat.

ECL :m

### [Début du catalogue.](#)

## TRIEITEMS :

Fonction TRIEITEMS <liste> [ {<, >} {ascii, nombre, taille, cardinal(card)} ] - fournit en résultat la liste de items de <liste> rangés suivant leur ordre croissant ou décroissant de leur code ASCII, numérique, leur longueur (taille) ou leur cardinal. Voir [CHOIX](#) , [DERNIER](#) , [INSERE](#) , [ITEM](#) , [LISTE](#) , [LISTE?](#) , [MEMBRE?](#) , [PHRASE](#) , [PERMUTATIONS](#) , [PREMIER](#) , [REMPLECE](#) , [SAISIS](#) , [SAUFPREMIER](#) , [SAUFDERNIER](#) , [xx || ...](#) , [ /e ou /E ] .

Exemples:

EFFTXT

SOIT "m TRIEITEMS [a b c d e f g h] [> ASCII ]

ECL :m // occasionnera l'affichage de [h g f e d c b a]

SOIT "m TRIEITEMS [h g f e d c b a] [< ASCII ]

ECL :m // occasionnera l'affichage de [a b c d e f g h]

SOIT "m TRIEITEMS [0 1 2 3 4 5 6 7 8 9] [> NOMBRE ]

ECL :m // occasionnera l'affichage de [9 8 7 6 5 4 3 2 1 0]

SOIT "m TRIEITEMS [9 8 7 6 5 4 3 2 1 0] [< NOMBRE ]

ECL :m // occasionnera l'affichage de [0 1 2 3 4 5 6 7 8 9]

SOIT "m TRIEITEMS [7777777 666666 55555 4444 333 22 1 ] [< TAILLE ]

ECL :m // occasionnera l'affichage de [1 22 333 4444 55555 666666 7777777]

SOIT "m TRIEITEMS [1 22 333 4444 55555 666666 7777777 ] [> TAILLE ]

ECL :m // occasionnera l'affichage de [7777777 666666 55555 4444 333 22 1]

SOIT "m TRIEITEMS [ [ f f f f f f ] [ e e e e e ] [ d d d d ] [ c c c ] [ b b ] [ a ] ] [< CARD ]

ECL :m // occasionnera l'affichage de [ [ a ] [ b b ] [ c c c ] [ d d d d ] [ e e e e e ] [ f f f f f f ] ]

SOIT "m TRIEITEMS [ [ a ] [ b b ] [ c c c ] [ d d d d ] [ e e e e e ] [ f f f f f f ] ] [> CARD ]

ECL :m // occasionnera l'affichage de [ [ f f f f f f ] [ e e e e e ] [ d d d d ] [ c c c ] [ b b ] [ a ] ]

Remarque : les crochets sont invisibles à l'affichage.

[Début du catalogue.](#)

UNDESDEUX? :

Fonction logique UNDESDEUX?: UNDESDEUX? <arg1> <arg2> - fournit en résultat les valeurs booléennes suivantes selon les valeurs booléennes respectives de <arg1> et <arg2> - UNDESDEUX? VRAI VRAI -> VRAI, UNDESDEUX? FAUX VRAI -> VRAI, UNDESDEUX? VRAI FAUX -> VRAI, UNDESDEUX? FAUX FAUX -> FAUX. La fonction UNDESDEUX? a la même table de vérité que le connecteur OU. Voir OU(), ET(&), LESDEUX?.

Exemple: SOIT "x 6 // éq : DONNE "x 6

ECL UNDESDEUX? (:x > 0) (:x < 5) rendra **VRAI** en résultat (car :x > 0).

[Début du catalogue.](#)

VA :

Procédure VA [arg] - où [arg] est une liste des coordonnées du lieu où la tortue devra se positionner. VA effectue le déplacement de la tortue en ce lieu. Voir LIEU, VERS, FIXEX, FIXEXY, FIXEY, FIXECAP, PASAPAS, RAMENE, SITUATION.

Exemple: DONNE "AncienLieu LIEU

/\* On supposera que LIEU équivaut à [ 30 -40] \*/

VA :AncienLieu

a pour effet de replacer la tortue au point de coordonnées [30 -40].

[Début du catalogue.](#)

## VARIANCE :

Fonction VARIANCE <liste\_vecteur> - où <liste\_vecteur> est une liste à une dimension qui ne contient que des nombres (liste vectorielle); fournit en résultat la variance des nombres contenus dans <liste\_vecteur> . Un argument incorrect fourni à VARIANCE occasionnera l'arrêt du programme LOGO. Voir [AFFICHE](#), [DETERMINANT](#), [MODEDEGRE](#), [MODERADIAN](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [OCCURRENCES](#), [PROEXT](#), [PRODINT](#), [TRIEITEMS](#), [VCOS](#), [VDIM](#), [VPROD](#), [VSIN](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

### Exemple:

```
ECL VARIANCE [32,5 (-14,8) 59,9 46,3 ]
```

fournira 682,776875 comme résultat.

[Début du catalogue.](#)

## VCOS :

Fonction VCOS <liste\_vecteur1> <liste\_vecteur2> - où <liste\_vecteur1> et <liste\_vecteur2> sont des listes à une dimension qui ne contiennent que des nombres (listes vectorielles) ; fournit en résultat le cosinus de l'angle entre <liste\_vecteur1> et <liste\_vecteur2>. Un argument incorrect transmis à VCOS entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [MODEDEGRE](#), [MODERADIAN](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PROEXT](#), [PRODINT](#), [VDIM](#), [VPROD](#), [VSIN](#), [V\\_PROPRES](#), [V\\_UNIQUE](#) .

### Exemple:

```
ECL VCOS [12 (-5)] [-19,5 46] produira -18,2064085878221 comme résultat.
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## VDIM :

Fonction VDIM <liste\_vecteur> - où <liste\_vecteur> est une liste à une dimension qui ne contient que des nombres (liste vectorielle) ; fournit en résultat le nombre de termes contenus dans <liste\_vecteur>. Si l'argument fourni à VDIM n'est pas une liste vectorielle, le résultat vaut -1 . Voir [AFFICHE](#), [DETERMINANT](#), [MODEDEGRE](#), [MODERADIAN](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#),

[M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [VCOS](#), [VPROD](#), [VSIN](#),  
[V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

**ECL** **VDIM** [ 5,3 2,4 0 (-9)] aura pour résultat 4 dans l'afficheur de textes.

[Début du catalogue.](#)

**VELOCITE?** :

Fonction VELOCITE? <lutin> - fournit en résultat la vitesse de déplacement de <lutin>.

Voir [AVANCELUTIN](#), [RECULELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#),  
[ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#),  
[COLORIELUTIN](#), [COULEURLUTIN?](#), [VITESSE](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#),  
[LUTINXV](#), [LUTINVY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [COLLISIONS?](#), [DEGRE](#), [RADIAN](#), [MODEDEGRE](#),  
[MODERADIAN](#) .

Exemple: **VELOCITE?** "lutin.

/\* "lutin doit cependant être défini dans l'éditeur de lutin. \*/

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**VERS** :

Procédure VERS [arg] - où [arg] est une liste des coordonnées du lieu où la tortue devra se diriger.

VERS effectue le changement de cap de la tortue qui pointera alors vers le lieu indiqué. Voir

[FIXEX](#), [FIXEXY](#), [FIXEY](#), [FIXECAP](#), [LIEU](#), [PASAPAS](#), [RAMENE](#), [SITUATION](#), [VA](#) .

Exemple: **SOIT** "AncienLieu [ XCOORD YCOORD ] /\* Coordonnées actuelles \*/

**AVANCE** 50 **TOURNEDROITE** 90 **AVANCE** 60 /\* La tortue a changé de  
place. \*/

**VERS** :AncienLieu /\* La tortue a pivoté pour faire face au point où elle se  
situait auparavant. \*/

[Début du catalogue.](#)

**VERSION?** :

Fonction VERSION? :carte - fournit la liste les informations techniques relatives au circuit logique

principal de la carte K8061/VM140 . Voir [CONNEXIONS](#), [CPUOK?](#), [DECONNECTE](#),

[FERMECARTE](#) .

Exemple: **ECL** **VERSION?**

### [Début du catalogue.](#)

#### VICEVERSA :

Fonction **VICEVERSA** <arg> où <arg> est une liste ou un mot - fournit en résultat la liste inversée des items de <arg>. Voir [DERNIER](#), [INSERE](#), [ITEM](#), [LISTEVARS](#), [OCCURRENCES](#), [PREMIER](#), [REMPPLACE](#), [ROTATION](#), [SAUFPREMIER](#), [SAUFDERNIER](#), [SAISIS](#), [SCINDE](#), [SWAP](#). Abrév. **VV**.

Exemple: **ECL VICEVERSA** [[abc de [fg hij klmn ]] opq rs tuv wxy z ]  
occasionnera l'affichage de :  
[ z wxy tuv rs opq [abc de [fg hij klmn ]]].

### [Début du catalogue.](#)

#### VIDE? :

Fonction **VIDE?** - fournit la valeur booléenne **VRAI** en résultat si le contenu d'une variable (liste ou mot) est vide. **FAUX** sinon. Voir [CHOIX](#), [LISTE](#), [LISTE?](#), [MOT](#), [MOT?](#), [OCCURRENCES](#), [PASAPAS](#), [PERMUTATIONS](#), [TRIECAR](#), [TRIEITEMS](#), [xx || ...](#).

Exemple: **DONNE** "NomVar []" ou  
**DONNE** "NomVar "  
**ECRIS VIDE?** :NomVar      rendra **VRAI** comme résultat.

### [Début du catalogue.](#)

[Vers l'index des primitives >>](#)

#### VISIBLE? :

Fonction **VISIBLE?** - retourne la valeur booléenne **VRAI** en résultat si la tortue est visible à l'écran. **FAUX** sinon. Voir [CACHETORTUE](#), [MONTRETORTUE](#), [PASAPAS](#).

Exemple: **MONTRETORTUE**  
**TESTE VISIBLE?**  
**SIVRAI ECRISLIGNE** [ Tu peux voir la tortue. ]

### [Début du catalogue.](#)

#### VITESSE :

Procédure **VITESSE** <lutin> <arg num> - fixe la vitesse de déplacement de <lutin> de <arg num> sans modifier son cap. Voir [AVANCELUTIN](#), [RECELELUTIN](#), [TOURNELUTINDROITE](#), [TOURNELUTINGAUCHE](#), [ENDORS](#), [REVEILLE](#), [CACHELUTIN](#), [MONTRELUTIN](#), [FIXELUTINX](#), [FIXELUTINY](#), [COLORIELUTIN](#), [COULEURLUTIN?](#), [POSITIONLUTIN?](#), [LUTINX?](#), [LUTINY?](#),

[LUTIN VX](#), [LUTIN VY](#), [CAPLUTIN](#), [CAPLUTIN?](#), [VELOCITE?](#), [COLLISIONS?](#) [DEGRE](#) , [RADIAN](#) ,  
[MODEDEGRE](#) , [MODERADIAN](#) .

Exemples: **VITESSE** "lutin1 50

ou:

**VITESSE** [ "lutin1 "lutin2 ] 50 .

"lutin1 et "lutin2 doivent cependant être définis dans l'éditeur de lutin.

[Début du catalogue.](#)

**VPROD** :

Fonction VPROD <liste\_vecteur1> <liste\_vecteur2> <arg\_booléen> - où <liste\_vecteur1> et <liste\_vecteur2> sont des listes à une dimension qui ne contiennent que des nombres (listes vectorielles) ; fournit en résultat le produit vectoriel de <liste\_vecteur1> et <liste\_vecteur2> soit sous forme d'une liste vectorielle si <arg\_booléen> vaut VRAI (1) ou d'un nombre si <arg\_booléen> vaut FAUX (0). Voir [AFFICHE](#), [DETERMINANT](#), [MODEDEGRE](#), [MODERADIAN](#), [M\\_ALEA](#) ,  
[M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#),  
[M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [VCOS](#), [VDIM](#), [VSIN](#),  
[V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

**ECL VPROD** [-5 12,6 24][8 102 (-49,3)] **VRAI**

produira [-40 1285,2 -1183,2] en résultat alors que

**ECL VPROD** [-5 12,6 24][8 102 (-49,3)] **FAUX**

produira 62 en résultat.

[Début du catalogue.](#)

**VSIN** :

Fonction VSIN <liste\_vecteur1> <liste\_vecteur2> - où <liste\_vecteur1> et <liste\_vecteur2> sont des listes à une dimension qui ne contiennent que des nombres (listes vectorielles) ; fournit en résultat le sinus de l'angle entre <liste\_vecteur1> et <liste\_vecteur2>. Un argument incorrect transmis à VSIN entraînera immédiatement l'arrêt du programme LOGO avec un message d'erreur. Voir [AFFICHE](#), [DETERMINANT](#), [MODEDEGRE](#), [MODERADIAN](#), [M\\_ALEA](#) , [M\\_DIAG](#), [M\\_DIFF](#),  
[M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#),  
[M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [VCOS](#), [VDIM](#), [VPROD](#),  
[V\\_PROPRES](#), [V\\_UNIQUE](#) .

Exemple:

```
ECL VSIN [0,004 (-0,01)] [(-0,105) 0,046]
```

produira 0,999686339188087 comme résultat.

[Début du catalogue.](#)

VRAI :

Valeur booléenne **VRAI** (comparaison lors d'un test). A utiliser de préférence avec les fonctions booléennes [DIFFERENTS?](#), [EGALITE?](#), [LISTE?](#), [NEGATIVE?](#), [NOMBRE?](#), [PROCEDURE?](#), [LESDEUX?](#), [UNDESDEUX?](#), [VIDE?](#), [VISIBLE?](#), [ZERO?](#). Voir [BOOLEEN](#), [FAUX](#).

n.b : Les valeurs booléennes **VRAI** et **FAUX** peuvent être utilisées telles quelles dans une expression conditionnelle, c'est à dire sans avoir recours aux signes de comparaison.

Exemple:

```
SI VRAI ECL [ Cette consigne sera toujours exécutée. ]
```

```
SI FAUX ECL [ Cette consigne ne sera jamais exécutée. ]
```

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

V PROPRES :

Fonction V\_PROPRES <liste\_matrice> <option> ; où <liste\_matrice> est une liste matricielle carrée de rang 2, 3 ou 4 - produit un résultat selon la valeur de <option> avec  $0 \leq \text{option} \leq 1$ . Lorsque <option> vaut 0, la liste (ou vecteur) des valeurs propres de <liste\_matrice> sera retournée (aussi appelée spectre). Lorsque <option> vaut 1, la fonction calculera la matrice des vecteurs propres de <liste\_matrice>. Rappel: les sous-listes de <liste\_matrice> représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_UNIQUE](#).

Exemple:

```
SOIT "matrice [[1 3][2 2]] AFFICHE V_PROPRES :matrice
```

produira l'affichage de [[1 1][1,5 -1]] comme résultat.

[Début du catalogue.](#)

## V\_UNIQUE :

Fonction V\_UNIQUE <liste\_matrice> ; construit à partir de <liste\_matrice>, une liste matricielle dans laquelle aucun terme n'est répété. Les valeurs redondantes seront alors remplacées par des valeurs aléatoires situées dans l'intervalle formé par les bornes supérieures et inférieures prises par les termes de <liste\_matrice>. Rappel: les sous-listes de <liste\_matrice> représentent les vecteurs-colonnes. Voir [AFFICHE](#), [DETERMINANT](#), [M\\_ALEA](#), [M\\_DIAG](#), [M\\_DIFF](#), [M\\_IDENT](#), [M\\_INVERSE](#), [M\\_NORME](#), [M\\_NULLE](#), [M\\_SOMME](#), [M\\_SUPPR](#), [M\\_TRACE](#), [M\\_TRANSPOSE](#), [M\\_TRIINF](#), [M\\_TRISUP](#), [PRODEXT](#), [PRODINT](#), [V\\_PROPRES](#) .

Dans l'exemple qui suit, on trouve plusieurs fois la valeur -8 dans :matrice . Cette valeur va être remplacée à chaque occurrence par une valeur différente choisie au hasard, ici -3, -2 -1 et -5.

Exemple:

```
SOIT "matrice [[-8 7 (-3)][-8 0 (-8)][14 (-8) 3]]  
AFFICHE V_UNIQUE :matrice FAUX
```

aura entre autres [[-3 7 (-3)][-2 0 (-1)][14 -5 3]] comme résultat.

[Début du catalogue.](#)

## WINUSB1 :

Procédure d'appel de la fenêtre de la carte interface 8055/VM110 . Cette fenêtre permet de fixer à l'aide de la souris les valeurs des tensions logiques et analogiques sur les sorties correspondantes ainsi que la lecture des entrées digitales. L'ouverture de cette fenêtre est également possible dans l'éditeur en cliquant sur la primitive WINUSB1 du catalogue. Voir [CONNECTE](#) , [DECONNECTE](#) .

Exemple: WINUSB1

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

## WINUSB2 :

Procédure d'appel de la fenêtre de la carte interface 8055/VM110 . Cette fenêtre permet de fixer à l'aide de la souris les valeurs des tensions logiques et analogiques sur les sorties correspondantes ainsi que la lecture des entrées digitales. L'ouverture de cette fenêtre est également possible dans l'éditeur en cliquant sur la primitive WINUSB2 du catalogue. Voir [CONNEXIONS](#) , [CPUOK?](#) , [DECONNECTE](#) , [FERMECARTE](#) .

Exemple: WINUSB2

[Début du catalogue.](#)

## XCOORD :

Fonction XCOORD - fournit en résultat la valeur numérique correspondant à l'abscisse horizontale de la tortue à l'écran.

Voir [YCOORD](#) , [DEROULE](#) , [ENROULE](#) , [FIXEX](#) , [FIXEXY](#) , [FIXEY](#) , [PASAPAS](#) .

Exemple: `DONNE "x XCOORD // éq : SOIT "x XCOORD`

[Début du catalogue.](#)

**YCOORD :**

Fonction YCOORD - fournit en résultat la valeur numérique correspondant à l'ordonnée verticale de la tortue à l'écran. Voir [XCOORD](#) , [DEROULE](#) , [ENROULE](#) , [FIXEX](#) , [FIXEXY](#) , [FIXEY](#) , [PASAPAS](#) .

Exemple: `DONNE "y YCOORD // éq : SOIT "y YCOORD`

[Début du catalogue.](#)

**ZEROS? :**

Fonction ZEROS? [ binf bsup précision nom\_fonction ] - explore l'intervalle [binf bsup](avec binf < bsup) à la recherche des racines de la fonction désignée par nom\_fonction et calcule la liste des racines. nom\_fonction ne doit comporter qu'une seul argument dans sa définition. Le temps de recherche dépend de la précision demandée et peut durer plusieurs minutes. Lorsqu'aucune racine n'a été trouvée, une liste vide est retournée. Le paramètre <précision> est fixé à 0,01 par défaut lorsqu'il est affecté à 0 dans l'appel de cette recherche. Voir [ABS](#) , [ARCCOS](#) , [ARCSIN](#) , [ARCTAN](#) , [COSINUS](#) , [DEGRE](#) , [EXP](#) , [FVECT](#) , [HEXA?](#) , [INTEGRALE](#) , [LN](#) , [LOG](#) , [MODEDEGRE](#) , [MODERADIAN](#) , [NON](#) , [OPPOSE](#) , [POLYSOLVE](#) , [RACINE2](#) , [RACINE3](#) , [RADIAN](#) , [SENSHORAIRE](#) , [SENSTRIGO](#) , [SINUS](#) , [TAN](#) , [TRANSLATE](#) .

Exemple:

```
POUR fx :x
RENDS (3 * (:x ^ 2)) + (2 * :x) - 1
FIN
```

```
EFFTXT
```

```
DONNE "a -1,2 DONNE "b 0,5 DONNE "précision 1e-10
```

```
DONNE "l [ :a :b :précision fx ]
```

```
ECL ZEROS? :l
```

**Méthode utilisée :**

Cette recherche procède en premier lieu par la segmentation de l'intervalle [binf bsup] en n sous-intervalles de la forme

$[(\text{binf} + (2 * (k - 1) * \text{précision})) (\text{binf} + (2 * k * \text{précision}))]$  pour k variant de 1 à n ( où  $n = (\text{bsup} - \text{binf}) / (2 * \text{précision})$  ).  
Une recherche par dichotomie très classique est ensuite effectuée sur chaque sous-intervalle.

[Début du catalogue.](#)

**ZERO?** :

Fonction ZERO? <argnum> - fournit en résultat la valeur booléenne [VRAI](#) si <argnum> a la valeur zéro. [FAUX](#) sinon.

Exemple: **ECRIS ZERO?** 6-6      rendra **VRAI** comme résultat.

// Attention ! LOGO préfère l'écriture **ZERO?** (6-6)

[Début du catalogue.](#)

**ZMAX** :

Procédure ZMAX <profondeur écran> - représente la distance du point de fuite à l'écran: c'est donc la distance maximale de la profondeur d'écran. ZMAX vaut 0 par défaut. Voir [ZMAX?](#) , [ZPOINT](#) , [ZPOINT?](#) et [ZSEGMENT](#) .

Exemple: **ZMAX** 100

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)

**ZMAX?** :

Fonction ZMAX? - fournit en résultat la distance du point de fuite. Voir [ZMAX](#) , [ZPOINT](#) , [ZPOINT?](#) et [ZSEGMENT](#) .

Exemple: **ECRISLIGNE ZMAX?**

[Début du catalogue.](#)

**ZPOINT** :

Procédure ZPOINT <X> <Y> <Z> - trace sur l'écran de la tortue un point aux coordonnées respectives X , Y et Z . Voir [ZMAX](#) , [ZMAX?](#) , [ZPOINT](#) et [ZSEGMENT](#) .

Exemple: **ECRIS ZPOINT?** 32 15 (-20) .

[Début du catalogue.](#)

**ZPOINT?** :

Fonction ZPOINT? <X> <Y> <Z> - trace sur l'écran de la tortue un point aux coordonnées respectives



<p><b><u>A</u></b></p> <p><a href="#">ABS</a> , <a href="#">ACCELERE</a> , <a href="#">AFFICHE</a> , <a href="#">ARC</a> , <a href="#">ARCCOS</a> , <a href="#">ARGCOSH</a> , <a href="#">ARCCOT</a> , <a href="#">ARGCOTH</a> , <a href="#">ARCCSC</a> , <a href="#">ARGCSCH</a> , <a href="#">ARCSEC</a> , <a href="#">ARGSECH</a> , <a href="#">ARCSIN</a> , <a href="#">ARGSINH</a> , <a href="#">ARCTAN</a> , <a href="#">ARGTANH</a> , <a href="#">ARRONDIR</a> , <a href="#">ASCII</a> , <a href="#">ATTENDS</a> , <a href="#">AUCUNE</a> , <a href="#">AVANCE (AV)</a> , <a href="#">AVANCELUTIN</a></p>	<p><b><u>B</u></b></p> <p><a href="#">BAISSECRAYON (BC)</a> , <a href="#">BARRES</a> , <a href="#">BOUCLE</a> , <a href="#">BOOLEEN</a></p>	
<p><b><u>C</u></b></p> <p><a href="#">CABLEUSB?</a> , <a href="#">CACHELUTIN (CL)</a> , <a href="#">CACHESPOTS</a> , <a href="#">CACHETORTUE (CT)</a> , <a href="#">CANAL?</a> , <a href="#">CANAL_MAX</a> , <a href="#">CANAL_MIN</a> , <a href="#">CANAUX?</a> , <a href="#">CANAUX_MAX</a> , <a href="#">CANAUX_MIN</a> , <a href="#">CAP</a> , <a href="#">CAPLUTIN</a> , <a href="#">CAPLUTIN?</a> , <a href="#">CAPTEURAVANT?</a> , <a href="#">CAPTEURDROIT?</a> , <a href="#">CAPTEURGAUCHE?</a> , <a href="#">CARACTERE (CAR)</a> , <a href="#">CARDINAL (CARD)</a> , <a href="#">CARTEPOINTS</a> , <a href="#">CARTEUSB</a> , <a href="#">CHAMPVECT</a> , <a href="#">CHANGEREPI</a> , <a href="#">CHOSE</a> , <a href="#">CHOIX</a> , <a href="#">CLIC?</a> , <a href="#">COLLISIONS?</a> , <a href="#">COLORIE</a> , <a href="#">COLORIELUTIN</a> , <a href="#">COMPAS</a> , <a href="#">COMPLEXE</a> , <a href="#">CONJUG</a> , <a href="#">CPLXADD</a> , <a href="#">CPLXPROD</a> , <a href="#">CONNECTE</a> , <a href="#">CONNEXIONS</a> , <a href="#">CONTENU</a> , <a href="#">COPIE</a> , <a href="#">COSH</a> , <a href="#">COSEC</a> , <a href="#">COSINUS (COS)</a> , <a href="#">COTAN</a> , <a href="#">COTH</a> , <a href="#">CSCH</a> , <a href="#">COULEUR?</a> , <a href="#">COULEURFOND (FCFG)</a> , <a href="#">COULEURFOND? (CF)</a> , <a href="#">COULEURLETTRE</a> , <a href="#">COULEURLUTIN?</a> , <a href="#">COULEURPOINT?</a> , <a href="#">COULEURTORTUE (FCC)</a> , <a href="#">COULEURTORTUE? (CC)</a> , <a href="#">CPUOK?</a> , <a href="#">CREEREPI</a></p>	<p><b><u>D</u></b></p> <p><a href="#">DATE</a> , <a href="#">DEC</a> , <a href="#">DECONNECTE</a> , <a href="#">DEFINIS (DEF)</a> , <a href="#">DEGRE</a> , <a href="#">DERNIER (DER)</a> , <a href="#">DERNIERELETTRE (DERL)</a> , <a href="#">DEROULE</a> , <a href="#">DESSINETEXTE</a> , <a href="#">DETERMINANT</a> , <a href="#">DETRUIS</a> , <a href="#">DIFFERENCE (-, DIFF)</a> , <a href="#">DIFFERENTS? (DIFF?)</a> , <a href="#">DIV</a> , <a href="#">DIVISEURS</a> , <a href="#">DONNE / SOIT</a></p>	
<p><b><u>F</u></b></p> <p><a href="#">FAIRE</a> , <a href="#">FAUX</a> , <a href="#">FERMECANAL</a> , <a href="#">FERMECANAUX</a> , <a href="#">FERMECARTE</a> , <a href="#">FERMEDLL</a> , <a href="#">FIN</a> , <a href="#">FIXECANAL</a> , <a href="#">FIXECANAL_L</a> , <a href="#">FIXECANAUX_A</a> , <a href="#">FIXECANAUX_L</a> , <a href="#">FIXECAP (FCAP)</a> , <a href="#">FIXEHASARD</a> , <a href="#">FIXELUTINX</a> , <a href="#">FIXELUTINXY</a> , <a href="#">FIXELUTINY</a> , <a href="#">FIXEMODULATION (FIXPWM)</a> , <a href="#">FIXEREBOND</a> , <a href="#">FIXEX</a> , <a href="#">FIXEXY</a> , <a href="#">FIXEY</a> , <a href="#">FONDECRAN</a> , <a href="#">FORME</a> , <a href="#">FVECT</a></p>	<p><b><u>G</u></b></p> <p><a href="#">GRAPHE</a> , <a href="#">GRAPH2D</a> , <a href="#">GRAPH3D</a> , <a href="#">GRAPHEUR</a> , <a href="#">" (guillemets)</a> , <a href="#">: (semi-colonne)</a></p>	<p><b><u>H</u></b></p> <p><a href="#">HASARD</a> , <a href="#">HEXA?</a> , <a href="#">HISTO</a> , <a href="#">\$</a></p>
<p><b><u>J</u></b></p> <p><a href="#">JOUE</a></p>	<p><b><u>I</u></b></p> <p><a href="#">IMAGE</a> , <a href="#">IMPRIMEECRAN</a> , <a href="#">IMPRIMETEXTE (IMTXT)</a> , <a href="#">INC</a> , <a href="#">INITGRAPH</a> , <a href="#">INSERE</a> , <a href="#">INTEGRALE</a> , <a href="#">INIT_COMPTEUR</a> , <a href="#">ITEM</a></p> <p><b><u>K</u></b></p> <p><a href="#">"K8055</a> , <a href="#">"K8061</a></p>	

<p><b><u>L</u></b></p> <p><a href="#">LESDEUX?</a>, <a href="#">LEVECRAYON (LC)</a>, <a href="#">LIEU</a>,  <a href="#">LIMITES?</a>, <a href="#">LISCAR</a>,  <a href="#">LISLISTE (LL)</a>, <a href="#">LISMAT</a>, <a href="#">LISNOMBRE</a>,  <a href="#">LISPORT</a>, <a href="#">LISTE</a>, <a href="#">LISTE?</a>, <a href="#">LISTEVARS</a>, <a href="#">LN</a>,  <a href="#">LOG</a>, <a href="#">LUTINVERS</a>, <a href="#">LUTIN VX</a>, <a href="#">LUTIN VY</a>,  <a href="#">LUTIN X?</a>, <a href="#">LUTIN Y?</a></p>	<p><b><u>M</u></b></p> <p><a href="#">MEMBRE?</a>, <a href="#">MAX?</a>, <a href="#">MIN?</a>, <a href="#">MOD (%)</a>,  <a href="#">MODEDEGRE</a>, <a href="#">MODERADIAN</a>,  <a href="#">MODULATION?</a>, <a href="#">MONTRELUTIN (ML)</a>,  <a href="#">MONTRESPOTS</a>, <a href="#">MONTRETORTUE (MT)</a>,  <a href="#">MOT</a>, <a href="#">MOT?</a>, <a href="#">M_ALEA</a>, <a href="#">M_DIAG</a>, <a href="#">M_DIFF</a>,  <a href="#">M_IDENT</a>, <a href="#">M_INVERSE</a>, <a href="#">M_NORME</a>,  <a href="#">M_NULLE</a>, <a href="#">M_SOMME</a>, <a href="#">M_SUPPR</a>, <a href="#">M_TRACE</a>,  <a href="#">M_TRANSPOSE</a>, <a href="#">M_TRINF</a>, <a href="#">M_TRISUP</a>.</p>
<p><b><u>N</u></b></p> <p><a href="#">NEGATIVE?</a>, <a href="#">NETTOIE</a>, <a href="#">NOMBRE</a>, <a href="#">NOMBRE?</a>,  <a href="#">NON</a></p>	<p><b><u>O</u></b></p> <p><a href="#">OBEIS</a>, <a href="#">OCCURRENCES (OCCURS)</a>, <a href="#">OPPOSE</a>,  <a href="#">ORIGINE</a>, <a href="#">OU()</a>, <a href="#">UNDESDEUX?</a>, <a href="#">OUVERT?</a>,  <a href="#">OUVERTS?</a>, <a href="#">OUVERTS_A?</a>, <a href="#">OUVRECANAL</a>,  <a href="#">OUVRECANAUX</a>, <a href="#">OUVREDLL</a>, <a href="#">OUVREPORT</a></p>
<p><b><u>P</u></b></p> <p><a href="#">PARTAGE</a>, <a href="#">PASAPAS</a>, <a href="#">PERMUTATIONS</a>,  <a href="#">PERMUTE</a>, <a href="#">PHRASE (PH)</a>, <a href="#">PI</a>, <a href="#">PIXELS?</a>,  <a href="#">PLEINECRAN</a>,  <a href="#">PLOT</a>, <a href="#">PLOT POL</a>, <a href="#">POINT</a>, <a href="#">POLYSOLVE</a>,  <a href="#">PORT</a>, <a href="#">PORTS?</a>, <a href="#">POSITIONLUTIN?</a>, <a href="#">POUR</a>,  <a href="#">PRODEXT</a>, <a href="#">PRODINT</a>, <a href="#">PREMIER (PREM)</a>,  <a href="#">PREMIERELETTRE (PREML)</a>, <a href="#">PROCEDURES?</a>,  <a href="#">PRODUIT (PROD)</a>, <a href="#">PUISSANCE (^)</a></p>	<p><b><u>Q</u></b></p> <p><a href="#">QUOTIENT (QUOT)</a></p> <p><b><u>R</u></b></p> <p><a href="#">RACINE2 (RAC2)</a>, <a href="#">RACINE3 (RAC3)</a>,  <a href="#">RADIAN</a>, <a href="#">RALENTIS</a>, <a href="#">RAMENE</a>, <a href="#">RANDDEC</a>,  <a href="#">RANDOM</a>, <a href="#">RAPPELLE</a>, <a href="#">RECULE (RE)</a>,  <a href="#">RECULELUTIN</a>, <a href="#">REEL</a>, <a href="#">REPLACE</a>, <a href="#">REPETE</a>,  <a href="#">REPENDRE (RENDS)</a>, <a href="#">RESTE (%)</a>, <a href="#">RETOURNE (RET)</a>,  <a href="#">REVEILLE</a>, <a href="#">ROTATION (ROT)</a>, <a href="#">RVB</a></p>

**S**

[SAISIS](#) , [SAUFDERNIER \(SD\)](#) ,  
[SAUFDERNIERELETTRE \(SDL\)](#) ,  
[SAUFPREMIER \(SP\)](#) ,  
[SAUFPREMIERELETTRE \(SPL\)](#) ,  
[SAUVE](#) , [SAUVEECRAN](#) , [SCINDE](#) , [SECANTE](#) ,  
[SECH](#) , [SINH](#) , [SEGMENT](#) , [SEGMENT3D](#) ,  
[SENSHORAIRE](#) , [SENSTRIGO](#) , [SEPARE](#) , [SI](#) ,  
[SIFAUX \(SINON\)](#) , [SILENCE](#) ,  
[SINUS \(SIN\)](#) , [SITUATION](#) , [SIVRAI](#) ,  
[SOMME](#) , [SOURISX](#) , [SOURISY](#) , [SPOTS](#) ,  
[STOP](#) , [STYLELETTRE](#) , [SUPPREP](#) , [SUPPRIME](#) ,  
[SURECRAN?](#) , [SURFACE](#) , [SUSPENDRE](#) ,  
[SURFACE](#) , [SWAP](#) , [SYSPLOT](#) , [SYSPLOTPOL](#) .

**T**

[TAILLELETTRE](#) , [TAN](#) , [TANH](#) ,  
[TANTQUE \(TANT QUE\)](#) , [TEMPO](#) ,  
[TEMPS](#) , [TESTE](#) , [TOPS?](#) , [TORTUES?](#) ,  
[TOUCHE?](#) , [TOURNEDROITE \(TD\)](#) ,  
[TOURNEGAUCHE \(TG\)](#) ,  
[TOURNELUTINDROITE \(TLD\)](#) ,  
[TOURNELUTINGAUCHE \(TLG\)](#) , [TOUTES](#) ,  
[TRACE](#) , [TRAIT](#) , [TRANSFORME](#) , [TRANSLATE](#) ,  
[TRIECAR](#) , [TRIEITEMS](#)

**U**

[UNDESDEUX?](#)

<p><u>V</u></p> <p><a href="#">VA</a> , <a href="#">VARIANCE</a> , <a href="#">VCOS</a> , <a href="#">VDIM</a> , <a href="#">VELOCITE?</a> ,  <a href="#">VERS</a> , <a href="#">VERSION?</a> , <a href="#">VICEVERSA (VV)</a> , <a href="#">VIDE?</a> ,  <a href="#">VISIBLE?</a> , <a href="#">VITESSE</a> , <a href="#">"VM110"</a> , <a href="#">"VM140"</a> ,  <a href="#">VPROD</a> , <a href="#">VSIN</a> , <a href="#">V PROPRES</a> , <a href="#">V UNIQUE</a> ,  <a href="#">VRAI</a> .</p>	<p><u>W</u></p> <p><a href="#">WINUSB1</a> , <a href="#">WINUSB2</a></p>
<p><u>X</u></p> <p><a href="#">XCOORD</a> , <a href="#">YCOORD</a></p>	<p><u>Z</u></p> <p><a href="#">ZEROS?</a> , <a href="#">ZERO?</a> , <a href="#">ZMAX</a> , <a href="#">ZMAX?</a> , <a href="#">ZPOINT</a> ,  <a href="#">ZPOINT?</a> , <a href="#">ZSEGMENT</a></p>
<p><b><u>Autres signes:</u></b></p> <p><a href="#">≤</a> , <a href="#">≤</a> , <a href="#">≥</a> , <a href="#">≥</a> , <a href="#">≡</a> , <a href="#">()</a> , <a href="#">[]</a> , <a href="#">/* */</a> , <a href="#">//</a> ,  <a href="#">[ , ]</a> , <a href="#">[ * *     " " ]</a> , <a href="#">[ /e ou /E ]</a> , <a href="#">&amp;</a> , <a href="#"> </a> , <a href="#">*"</a> , <a href="#">~*</a></p>	<p><b><u>Les couleurs :</u></b></p> <p><a href="#">BLANCHE</a> , <a href="#">BLEUE</a> , <a href="#">JAUNE</a> , <a href="#">NOIRE</a> , <a href="#">ROUGE</a> ,  <a href="#">VERTE</a> , <a href="#">TILLEUL</a> , <a href="#">EAU</a> , <a href="#">FUCHSIA</a> , <a href="#">SARCELLE</a> ,  <a href="#">GRISE</a> , <a href="#">OLIVE</a> , <a href="#">MARRON</a> , <a href="#">ARGENT</a> , <a href="#">ORANGE</a> ,  <a href="#">ROSE</a> , <a href="#">VIOLETTE</a></p>

[Début du catalogue.](#)

[Vers l'index des primitives >>](#)