

La [licence](#) se trouve à la dernière page.
Conseil : imprimez ce mode d'emploi.
Vous avez à votre disposition [un forum](#)

Cette version de Dmaths fonctionne sous LibreOffice >= 7.6 ou OpenOffice >= 4.1

Vous pouvez afficher ou cacher la barre d'icônes de Dmaths **en suivant :**
Affichage > Barres d'outils > DmathsAddin.

Vous pouvez désactiver ou activer la virgule comme séparateur décimal :
Linux ou Windows Outils > Options ou *Mac OS X LibreOffice* > Préférences
puis Paramètres linguistiques > Langues > Touche séparateur de décimales.

[Introduction](#)

[Ce qu'il faut savoir avant de commencer](#)

[Éditeur de fonctions personnelles](#)

[Exemples d'utilisation](#)

[Débuguer les fonctions personnelles](#)

[La console APSO](#)

[Assurer la portabilité](#)

[Gestion des variables mémorisées](#)

Introduction

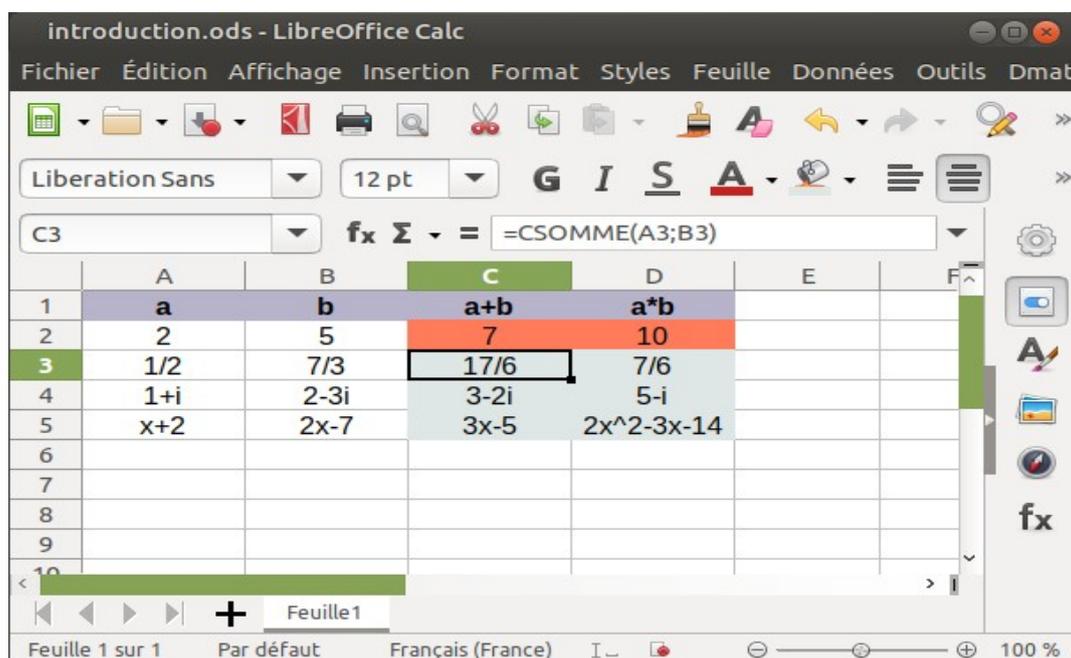
Dmaths Addin est une extension (un Addin) écrite en Python.

Je tiens à remercier chaleureusement Christophe Devalland pour l'extraordinaire travail fait avec CmathsOOoCas, qui m'a donné envie de créer un module «Dmaths Addin » pour Calc ainsi qu'Hubert Lambert pour l'excellent tutoriel « Construire un add-in » publié sur le site Ooo Forum.

Les fonctions mathématiques incluses dans Calc ne savent travailler que sur des nombres décimaux. On ne peut donc manipuler des nombres comme $\sqrt{2}$ ou des expressions comme x^2+x+1 .

Dmaths Addin étend les capacités du module Calc de LO/Aoo en introduisant le calcul formel.

Dmaths Addin donne accès à de nouvelles fonctions qui permettent de travailler sur des objets mathématiques formels.



Les cellules C2 et D2 de **couleur saumon** contiennent respectivement les formules saisies `=CSOMME(A2;B2)` et `CPRODUIT(A2;B2)`.

Les formules contenues dans les cellules de la plage C3::D5, de **couleur vert pâle** sont obtenues par « recopie automatique » vers le bas.

[Retour accueil](#)

Avant de commencer

1) La notion d'expression

Pour entrer un décimal dans une cellule de Calc, il suffit de l'écrire. Le logiciel le reconnaît, le contenu de la cellule est alors aligné à droite.

Pour entrer une chaîne de caractère, on écrit par exemple ="ma chaine". Le contenu de la cellule est alors aligné à gauche. On peut également saisir "ma chaine" ou encore 'machaine.

Ce sont ces objets que manipule Dmaths Addin. Lorsque dans l'aide vous lisez « une expression », c'est en général un décimal ou une chaîne qui sont attendus. Notez que si vous voulez utiliser une fraction comme $\frac{3}{4}$ vous devrez saisir "3/4" ou '2/3.

Un clic sur le bouton  permet d'aligner toutes les cellules de la feuille à droite.

Si vous dépassez les capacités de calcul de Calc ou un nombre non rationnel ou écrit sous la forme de fraction, vous devrez alors utiliser une chaîne de caractères par exemple :

= "1,456987974858756748714578"

= "2/3"

= "sqrt(7)" (pour racine carrée de 7)

= 'exp(2)

= 'pi/4

= "x^2+x+1"

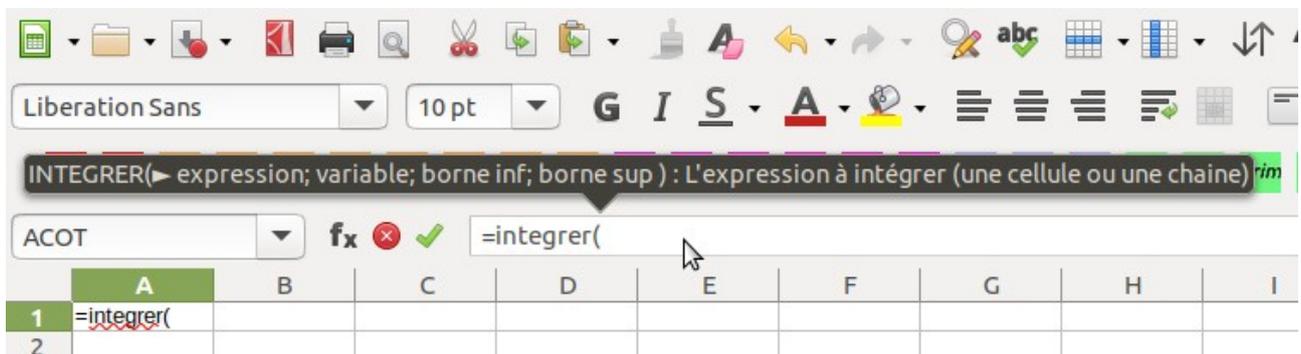
= '2-3i"

2) l'aide intégrée

Les fonctions ajoutées par Dmaths Addin disposent d'une aide intégrée.

Dès que l'on tape une fonction reconnue suivie de la parenthèse ouvrante, Calc affiche les arguments attendus pour cette fonction.

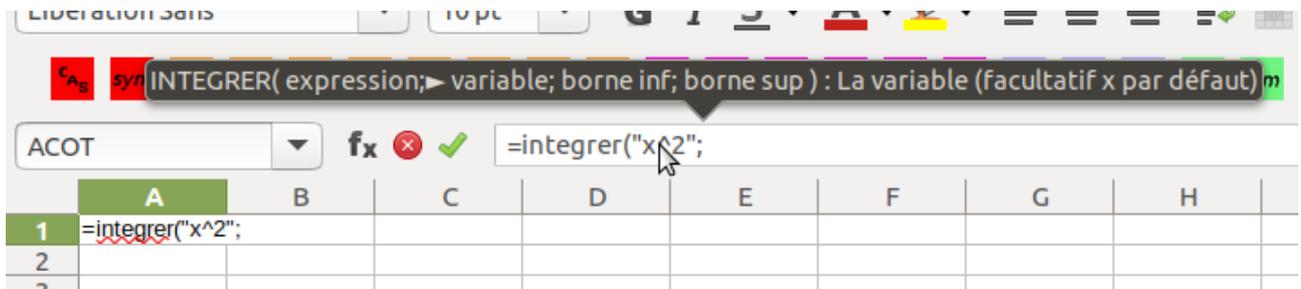
Par exemple, il suffit de cliquer sur une cellule et de taper =integrer(pour que Calc affiche l'aide correspondante :



Le petit triangle désigne l'argument demandé à cet instant de la frappe.

Le séparateur d'argument pour le tableur est le point-virgule.

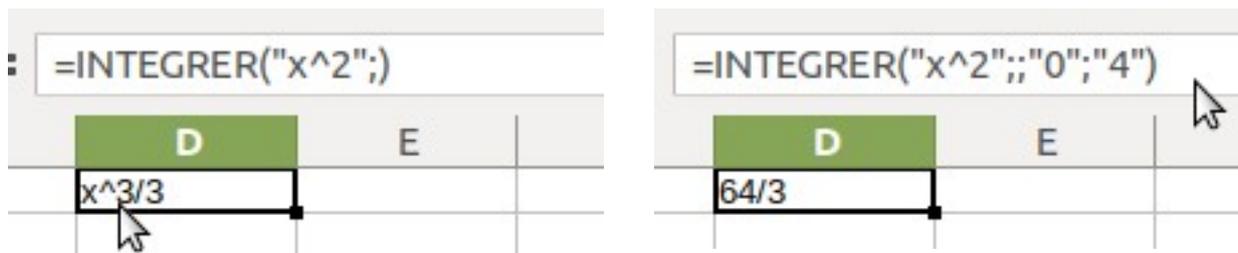
Tapez par exemple "x^2" puis point-virgule et le triangle se déplacera sur « variable ».



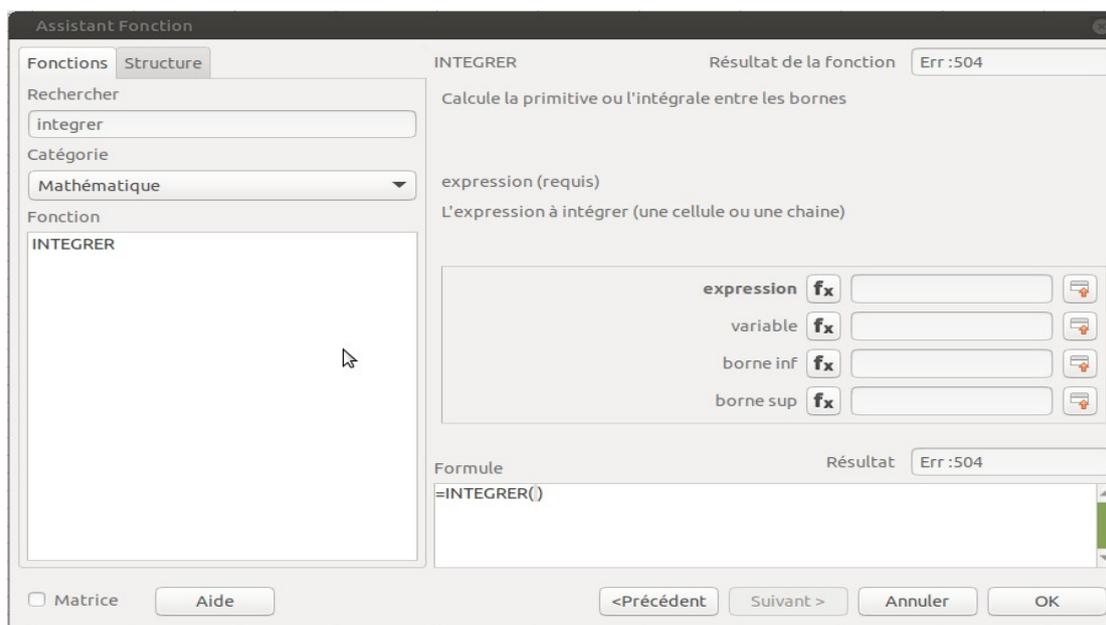
[Retour accueil](#)

Terminez le calcul en fermant la parenthèse pour avoir une primitive ou en complétant avec les

bornes pour avoir le calcul d'une intégrale.



Ces aides sont également disponibles de façon plus détaillées dans l' « Assistant Fonctions » de OOoCalc (bouton f(x)) :



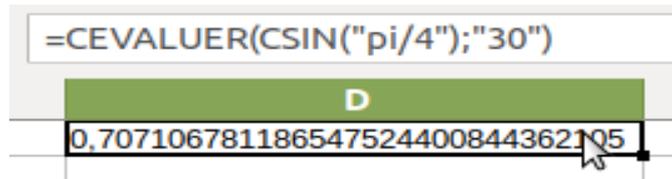
3) Les commandes les plus courantes sont regroupées dans la barre d'outils.



Ces boutons fonctionnent de deux façons différentes.

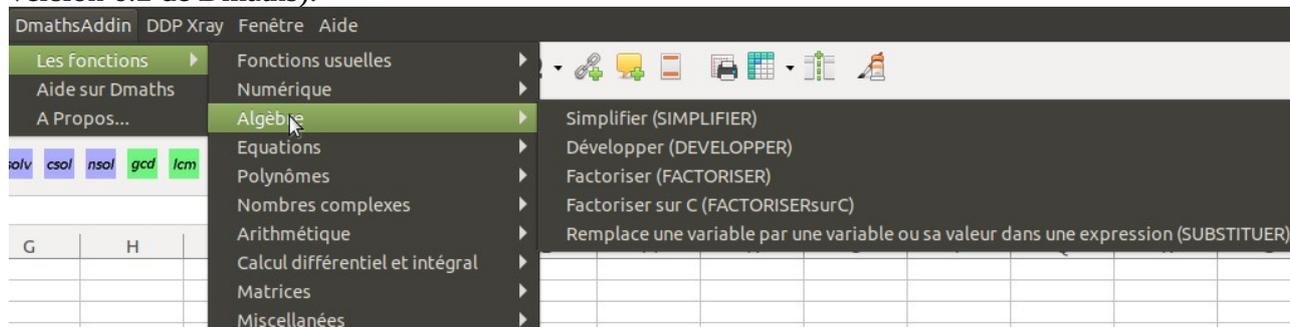
1) Ils peuvent servir de raccourcis lorsqu'on écrit une formule dans une cellule (le curseur clignotant doit être dans la cellule). A ce moment, un clic sur l'une de ces icônes écrit la fonction au point d'insertion. Exemple en cliquant sur l'icône bleue solv la cellule contiendra =RESOUDRE(expression) Cela fonctionne également si vous avez sélectionné une plage de cellules.

2) Pour les fonctions qui peuvent opérer sur un seul argument, telles que « simplifier », « évaluer », « racine carrée », si une ou plusieurs cellules contenant des formules sont sélectionnées, pour chaque cellule, la fonction cliquée sera composée avec la formule contenue dans la cellule. Par exemple, si une cellule contient ="x+2x", un clic sur le bouton « simplifier » modifiera la formule en =simplifier("x+2x") et permettra d'afficher ainsi 3x en appuyant sur la touche Entrée. Si une cellule contient =csin("pi/4") et affiche donc sqrt(2)/2, un clic sur le bouton « évaluer » modifiera la formule en =evaluer(csin("pi/4")). En ajustant le nombre de décimales, vous aurez le résultat :



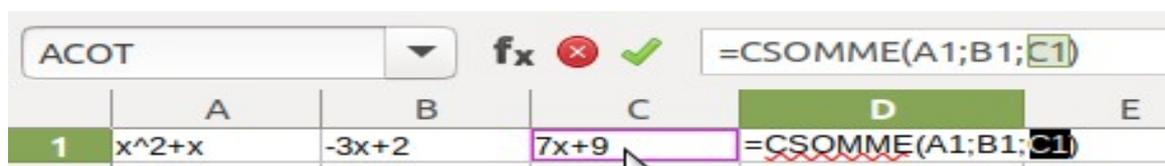
4) Utiliser les menus DmathsAddin

Les menus DmathsAddin donnent accès à toutes les fonctions nouvelles implémentées (78 dans la version 6.2 de Dmaths).



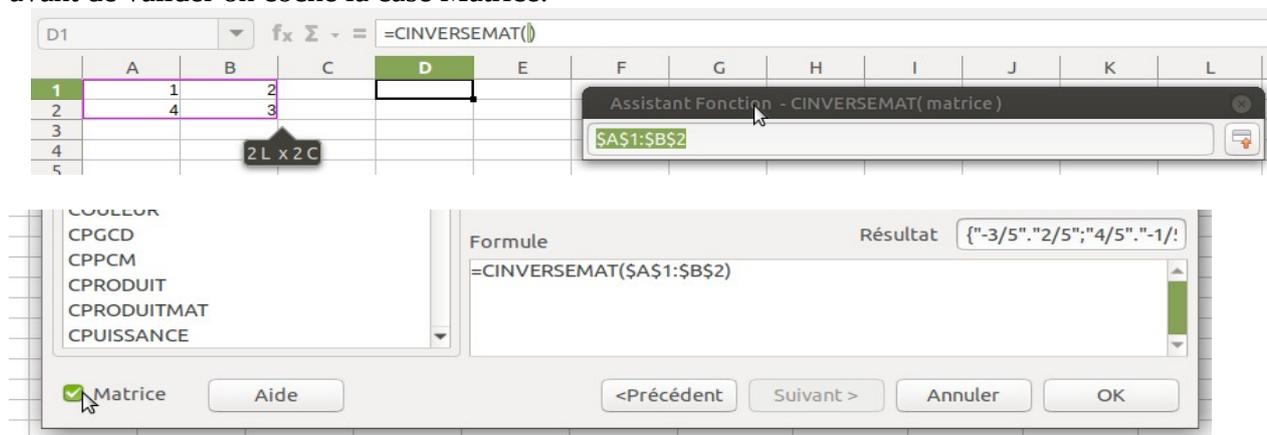
5) Utiliser la souris

Par exemple, on saisit `=csomme(` dans la cellule, puis la touche CTRL enfoncée, on clique successivement sur A1, B1 et C1.



6) Cas des fonctions renvoyant plusieurs valeurs

a) Le moins rapide : en utilisant l' « Assistant Fonctions », pour inverser une matrice on invoque la fonction CINVERSEMAT, on sélectionne la plage de cellules contenant la matrice à inverser, puis avant de valider on coche la case Matrice.



[Retour accueil](#)

D1		fx Σ ▾ =		=CINVERSEMAT(\$A\$1:\$B\$2)	
	A	B	C	D	E
1	1	2		-3/5	2/5
2	4	3		4/5	-1/5

b) Plus rapide, on saisit dans la cellule en haut et à gauche de la plage de résultat la formule =CINVERSEMAT(), puis on sélectionne la plage A1:B2, puis on clique sur CTRL+MAJ+Entrée (⌘+Maj+Entrée pour MacOS).

7) Différents types d'affichage

Sept fonctions permettent de choisir le type d'affichage (RESOUDRE, RESOUDREdansC, RESOUDRENumerique, ZEROS, CZEROS, DIVISEURSEntier, DIVISEURSPolynome) à l'aide du paramètre « affichage ».

0 pour le résultat dans une cellule (par défaut), 1 sur une ligne, 2 sur une colonne.

Dans les deux derniers cas, ne pas oublier de taper Ctrl+Maj+Entrée (⌘+Maj+Entrée pour MacOS)

=RESOUDRE("x^2-3=0";0)	{=RESOUDRE("x^2-3=0";1)}	{=RESOUDRE("x^2-3=0";2)}
D E	D E	D E
sqrt(3) ; -sqrt(3)	sqrt(3) -sqrt(3)	sqrt(3) -sqrt(3)

8) Résultat décimal ou exact

Par exemple dans un calcul d'intégrale, on obtient la valeur exacte en utilisant des chaînes de caractère pour les bornes

=INTEGRER("x^2";;0;10)	=INTEGRER("x^2";;"0";"10")
D E	D E
333.333333333333	1000/3

9) La fonction SYMPY (icône rouge SIM)

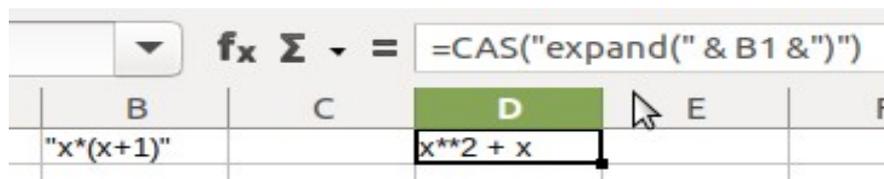
permet d'utiliser des commandes Python ou Sympy dans le tableur.

Par exemple en saisissant =SYMPY("nextprime";20) vous obtiendrez l'entier premier immédiatement supérieur à 20 soit 23.

=SYMPY("nextprime";20)			=SYMPY("expand";D1)	
D	E	F	D	E
23			(x+1)^2	x^2+2x+1

10) La fonction CAS

permet d'utiliser directement des commandes CAS. Voyez la syntaxe :

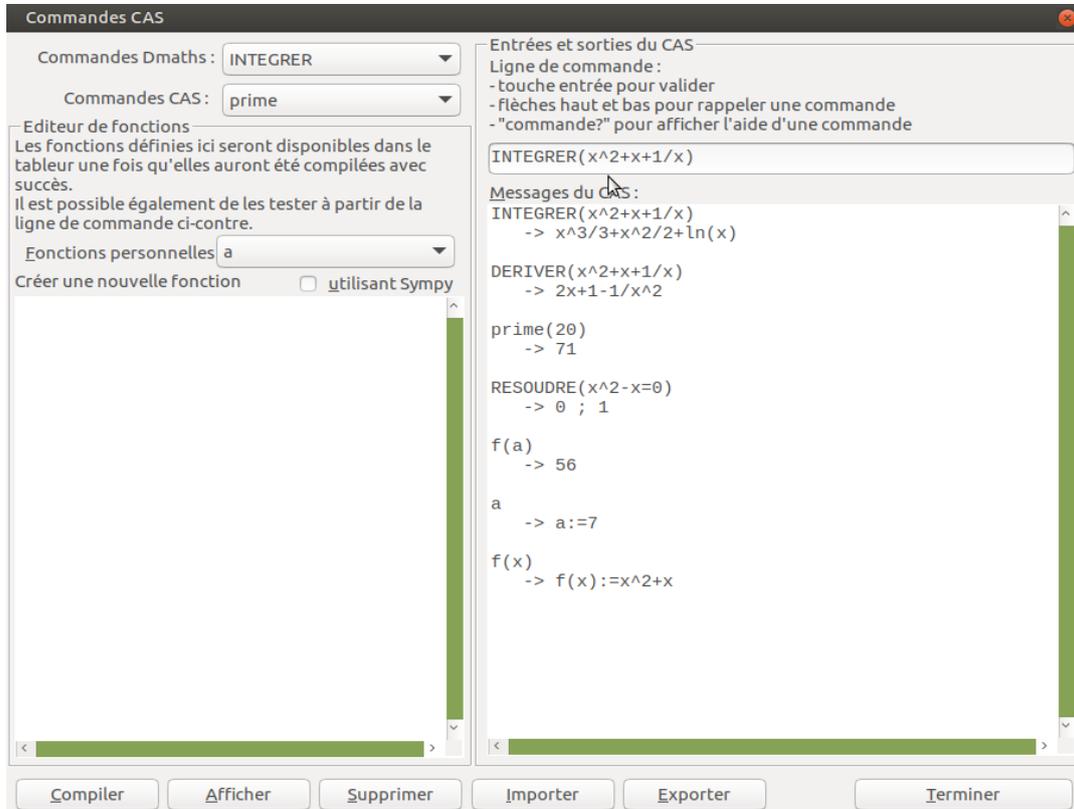


[Retour accueil](#)

ÉDITEUR DE FONCTIONS ET LIGNE DE COMMANDE

Une boîte de dialogue contenant un éditeur de fonctions ainsi qu'une ligne de commandes de calculs formels est accessible par l'icône de la barre d'outils  ou le raccourci Alt+W (Ctrl+W sous Mac)

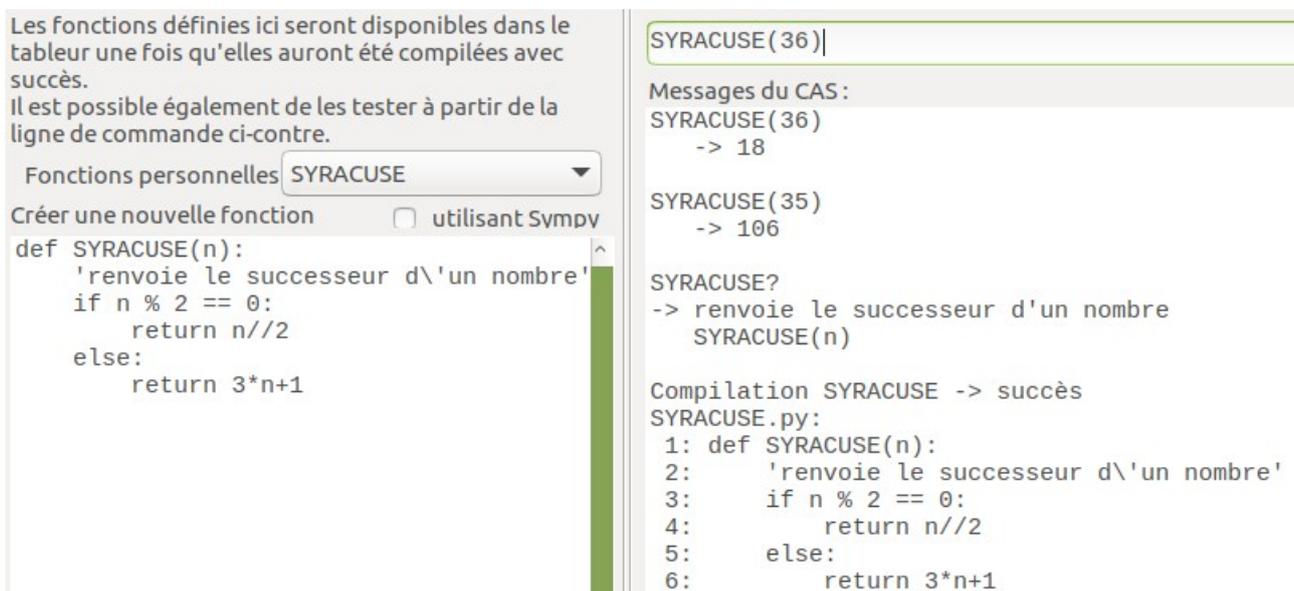
1) Une ligne de commande intégrée



2) L'éditeur permet de créer des fonctions personnelles.

À titre d'exemple voici la suite de Syracuse.

Attention à bien respecter la syntaxe de Python :



Vous pouvez documenter la fonction avec une chaîne de caractère. Notez le \ pour gérer le '. Elles sont immédiatement disponibles dans le tableur. Dans la cellule D2, on saisit =SYRACUSE(D1) puis on recopie vers le bas :

=SYRACUSE(D1)

D
137
412
206
103
310
155
466

Les fonctions créées sont utilisables dans l'éditeur et le tableur :

Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès. Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles: DUREEDEVOL

Créer une nouvelle fonction utilisant Sympy

```
def DUREEDEVOL(n):
    i = 0
    while n!= 1:
        n = SYRACUSE(n)
        i = i+1
    return i
```

DUREEDEVOL(137)|

Messages du CAS:
DUREEDEVOL(137)
-> 90

Compilation DUREEDEVOL -> succès
DUREEDEVOL.py:
1: import sys
2: CheUtil = '/home/dmaths/.config/libreoffice/4/user/Scripts/'
3: sys.path.append(CheUtil)
4: from SYRACUSE import *
5: def DUREEDEVOL(n):
6: i = 0
7: while n!= 1:
8: n = SYRACUSE(n)
9: i = i+1
10: return i

La durée de vol du nombre 137 est 90 c'est à dire qu'il faut 90 itérations successives de la fonction Syracuse pour aller de 137 à 1.

3) Fonctions définies au préalable dans le CAS

En tapant $f(x)=x^2$ puis F10 dans Writer ou encore $f(x)=x^2$ dans la ligne de commande de la boîte de dialogue dans Calc, on définit la fonction correspondante dans le CAS.

En cliquant sur afficher puis sur compiler, on définit alors la fonction personnelle tableur F correspondante.

Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès. Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles: F

Créer une nouvelle fonction utilisant Sympy

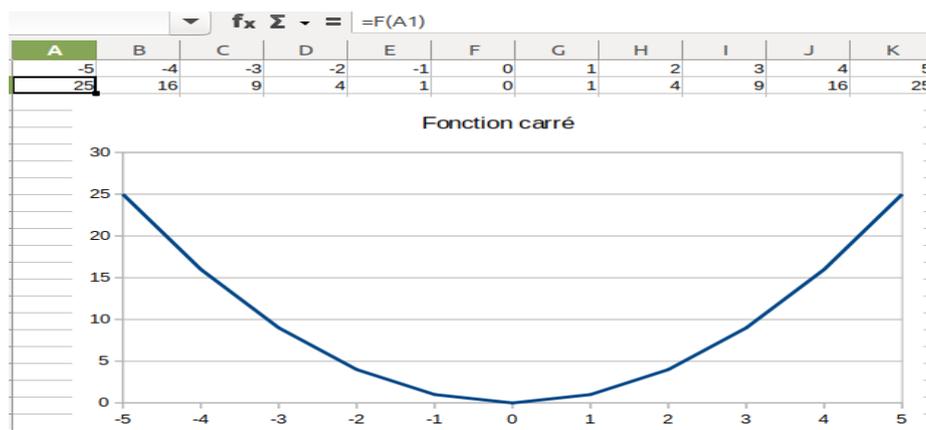
```
def F(x):
    return x^2
```

F|

Messages du CAS:
Compilation F -> succès
F.py:
1: from sympy import *
2: x = Symbol('x')
3:
4: def F(x):
5: x = sympify(x)
6: return x**2

f(x)
-> f(x):=x^2

Fonction que l'on peut facilement utiliser dans Calc pour obtenir un tableau de valeurs et une diagramme.



4) L'utilité du calcul formel

Le calcul formel peut permettre de conjecturer un résultat. Par exemple on peut chercher à conjecturer le nombre dérivé en a de la fonction carré.

Commençons par définir une fonction tableur personnelle qui calcule $\lim_{x \rightarrow a} \left(\frac{f(x) - f(a)}{x - a} \right)$.

Pour la syntaxe, une recherche en ligne avec les mots clés Sympy et limit conduit à :

Sympy can compute symbolic limits with the `limit` function. The syntax to compute

$$\lim_{x \rightarrow x_0} f(x)$$

is `limit(f(x), x, x0)`.

Run code block in SymPy Live

```
>>> limit(sin(x)/x, x, 0)
1
```

`f.subs(x,a)` pour évaluer f en a, puis dans l'éditeur de fonctions :

Commandes CAS: CABS
Commandes CAS: expand

Éditeur de fonctions
Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès.
Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles: NBREDERIVE

Créer une nouvelle fonction utilisant Sympy

```
def NbreDerive(f,a):
    g = (f-f.subs(x,a))/(x-a)
    return limit(g,x,a)
```

Entrées et sorties du CAS
Ligne de commande:
- touche entrée pour valider
- flèches haut et bas pour rappeler une commande
- "commande?" pour afficher l'aide d'une commande

```
NBREDERIVE(x**2;a)
Messages du CAS:
NBREDERIVE(x**2;a)
-> 2a
NBREDERIVE(x**2;3)
-> 6
NBREDERIVE(x**2;2)
-> 4
```

Compilation NBREDERIVE -> succès

```
import sys
chemin = 'file:///home/dmaths/.config/libreoffice/4/u'
sys.path.append(chemin)
from sympy import *
x = Symbol('x')

def cas21(f,a):
    f = sympify(f)
    a = sympify(a)
    g = (f-f.subs(x,a))/(x-a)
    return limit(g,x,a)
```

Calculons le nombre dérivé de la fonction carré en 3 puis en a en modifiant le contenu de B2 :

		A	B	C	D
C2		fx Σ ▾ = =NBREDERIVE(A2;B2)			
		1	Fonction	Valeur de a	Nbre dérivé en a
		2	x^2	a	2a
1	Fonction	Valeur de a	Nbre dérivé en a	Fonction dérivée	
2	x^2	a	2a		

Enfin, le calcul formel de la dérivée de la fonction, permet de valider (de ce point de vue), la conjecture.

		A	B	C	D
D2		fx Σ ▾ = =DERIVER(A2)			
		1	Fonction	Valeur de a	Nbre dérivé en a
		2	x^2	a	2a
1	Fonction	Valeur de a	Nbre dérivé en a	Fonction dérivée	
2	x^2	a	2a	2x	

5) Fonctions renvoyant plusieurs valeurs

Programmons la fonction personnelle DICH0 qui va nous retourner un encadrement des solutions d'une équation :

Commandes CAS

Commandes Dmaths : CABS

Commandes CAS : expand

Editeur de fonctions
Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès.
Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles DICH0

Créer une nouvelle fonction utilisant Sympy

```
def DICH0(f, a, b, ep):
    while (b-a) > ep:
        m = f.subs(x, a)
        n = f.subs(x, (a+b)/2)
        if m*n > 0:
            a = (a+b)/2
        else:
            b = (a+b)/2
    return [[a], [b]]
```

Entrées et sorties du CAS

Ligne de commande :
- touche entrée pour valider
- flèches haut et bas pour rappeler une commande
- "commande?" pour afficher l'aide d'une commande

DICH0(x^2-3;1.1;2;0.0001)

Messages du CAS :
DICH0(x^2-3;1.1;2;0.0001)
-> [[1, 73204345703125]; [1, 73209838867187]]

DICH0(x^2-3;1;2;0.0001)
-> [[28377/16384]; [14189/8192]]

Compilation DICH0 -> succès

```
import sys
chemin = 'file:///home/dmaths/.config/libreoffice/4/u
sys.path.append(chemin)
from sympy import *
x = Symbol('x')

def cas41(f, a, b, ep):
    f = sympify(f)
```

Dmaths renvoie une liste de listes. La ligne return doit donc contenir [[a],[b]] pour avoir une matrice avec une colonne et deux lignes et donc un résultat sur une colonne dans le tableur. Pour un résultat en ligne il aurait fallu écrire [[a,b]].

Il faut également penser « Python » : si a et b sont deux entiers, la fonction retourne un encadrement par des fractions. Si nous voulons obtenir un encadrement par des décimaux alors il faut poser a=1.1 ou a=1.0

Pour obtenir le résultat dans le tableur, il convient de se positionner en E2, de saisir la formule =DICH0(A2;B2;C2;D2), puis de taper CTRL+MAJ+Entrée (⌘+Maj+Entrée pour MacOS).

		A	B	C	D	E
E2		fx Σ ▾ = {=DICH0(A2;B2;C2;D2)}				
		1	Fonction	a	b	epsilon
		2	x^2-3	1,1	2	0,00001
1	Fonction	a	b	epsilon		encadrement
2	x^2-3	1,1	2	0,00001		1,73205032348633
3						1,73205718994141

6) Gérer l'affichage du résultat des fonctions personnelles renvoyant une liste de valeurs

La fonction LISTEPREMIERS renvoie la liste des nombres premiers compris entre les entiers a et b. La fonction personnelle peut retourner une chaîne ou une liste de listes.

a) Affichage du résultat dans une cellule : dans ce cas le résultat doit être une chaîne de caractères.

The screenshot shows the CAS interface. On the left, a code editor contains the function definition for LISTEPREMIERS. The main window shows the command LISTEPREMIERS(10;23) entered in a cell. The CAS messages display the function call and its output: 11;13;17;19;23. The spreadsheet grid shows the result in cell D1.

```
Créer une nouvelle fonction  utilisant Sympy
def listepremiers(a,b):
    l = []
    for x in range(a,b+1):
        if isprime(x):
            l.append(str(x))
    return ','.join(l)
```

LISTEPREMIERS(10;23) =LISTEPREMIERS(10;23)

Messages du CAS :
LISTEPREMIERS(10;23)
-> 11;13;17;19;23

	D	E
	11;13;17;19;23	

b) Affichage du résultat sur une ligne : dans ce cas le résultat doit être une liste comportant une liste comme seul terme. N'oubliez pas de saisir CTRL+MAJ+Entrée (⌘+Maj+Entrée pour MacOS)

The screenshot shows the CAS interface. On the left, a code editor contains the function definition for LISTEPREMIERSL. The main window shows the command LISTEPREMIERSL(10;23) entered in a cell. The CAS messages display the function call and its output: [[11;13;17;19]]. The spreadsheet grid shows the result in cell C1.

```
Créer une nouvelle fonction  utilisant Sympy
def LISTEPREMIERSL(a,b):
    L = []
    l = []
    for x in range(a,b):
        if isprime(x):
            l.append(str(x))
    L.append(l)
    return L
```

LISTEPREMIERSL(10;23) Σ = {=LISTEPREMIERSL(10;23)}

Messages du CAS :
LISTEPREMIERSL(10;23)
-> [[11;13;17;19]]

	C	D	E
	[[11;13;17;19]]		

b) Affichage du résultat sur une colonne : dans ce cas le résultat doit être une liste comportant une liste comme seul terme. N'oubliez pas de saisir CTRL+MAJ+Entrée (⌘+Maj+Entrée pour MacOS)

The screenshot shows the CAS interface. On the left, a code editor contains the function definition for LISTEPREMIERSC. The main window shows the command LISTEPREMIERSC(10;23) entered in a cell. The CAS messages display the function call and its output: [[11];[13];[17];[19]]. The spreadsheet grid shows the result in cell D1.

```
Créer une nouvelle fonction  utilisant Sympy
def listepremieresc(a,b):
    L = []
    for x in range(a,b):
        if isprime(x):
            l = []
            l.append(str(x))
            L.append(l)
    return L
```

LISTEPREMIERSC(10;23) {=LISTEPREMIERSC(10;23)}

Messages du CAS :
LISTEPREMIERSC(10;23)
-> [[11];[13];[17];[19]]

	D	E
	[[11];[13];[17];[19]]	

7) Paramètres optionnels

Ils se définissent selon la syntaxe Python

The screenshot shows the CAS interface. On the left, a code editor contains the function definition for P. The main window shows the command P(9;3) entered in a cell. The CAS messages display the function call and its output: 729. Below that, P(9) is shown with output 81. At the bottom, P? is shown with output 'Fonction personnelle P(x;n = 2)'. The spreadsheet grid shows the result in cell D1.

Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès. Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles P

```
Créer une nouvelle fonction  utilisant Sympy
def P(x,n = 2):
    return x^n
```

P(9;3)

Messages du CAS :
P(9;3)
-> 729

P(9)
-> 81

P?
-> Fonction personnelle
P(x;n = 2)

	D
	729

[Retour accueil](#)

QUELQUES EXEMPLES D'UTILISATION

1) Conduire une séance d'exercices avec les corrigés

Niveau seconde : factoriser, développer

Niveau première : dériver

Niveau terminale : intégrer, manipuler les complexes.

	A	B	C	D	E	F	G	H	I	J
1	Expressions	Développer	Expressions	Factoriser	Expressions	Dériver	Expressions	Intégrer	Complexes	Forme exponentielle
2	$x(x+1)$	x^2+x	x^2-3x	$x(x-3)$	$2x^2+x$	$4x+1$	$2x^2+x$	$2x^{3/3}+x^{2/2}$	$1+i$	$\sqrt{2}e^{i\pi/4}$
3	$(x+1/2)(2x+3)$	$2x^2+4x+3/2$	x^2+4x+4	$(x+2)^2$	$1/(2x)$	$-1/(2x^2)$	$x+1/x$	$x^{2/2}+\ln(x)$	-2	$2e^{i\pi i}$
4	$(x^2)^2$	x^4	x^2-6x+9	$(x-3)^2$	$(x+1)/x$	$-1/x^2$	$\exp(2x)$	$e^{(2x)/2}$	$8-8i$	$8\sqrt{2}e^{-i\pi/4}$
5	$(-2x+3)^2$	$4x^2-12x+9$	$2x^2+5x$	$x(2x+5)$	$\sin(2x)$	$2\cos(2x)$	$\sin(2x)$	$-\cos(2x)/2$	$(1+i)(1-i)$	2

Les solutions sont proposées grâce à la fonction recopiée vers le bas du tableur.

2) Tableau de valeurs usuel des fonctions trigonométriques

	A	B	C	D	E	F
1	Variable	0	$\pi/6$	$\pi/4$	$\pi/3$	$\pi/2$
2	Sinus	0	$1/2$	$\sqrt{2}/2$	$\sqrt{3}/2$	1
3	Cosinus	1	$\sqrt{3}/2$	$\sqrt{2}/2$	$1/2$	0
4	Tangente	0	$\sqrt{3}/3$	1	$\sqrt{3}$	zoo

A noter le zoo qui signifie résultat inexistant pour Sympy.

Le tableau peut être copié dans le traitement de textes :

Variable	0	$\pi/6$	$\pi/4$	$\pi/3$	$\pi/2$
Sinus	0	$1/2$	$\sqrt{2}/2$	$\sqrt{3}/2$	1
Cosinus	1	$\sqrt{3}/2$	$\sqrt{2}/2$	$1/2$	0
Tangente	0	$\sqrt{3}/3$	1	$\sqrt{3}$	zoo

3) Niveau collègue : ajouter deux fractions

	A	B	C	D	E	F
1	Fraction 1	Fraction 2	Réduite 1	Réduite 2	Somme	Résultat
2	$3/2$	$5/4$			#VALEUR !	$11/4$
3	$1/2$	$1/3$			#VALEUR !	$5/6$
4	$5/2$	$7/3$			#VALEUR !	$29/6$
5	$11/3$	$3/4$			#VALEUR !	$53/12$
6	a/b	c/d			#VALEUR !	$(ad+bc)/(bd)$

On peut fournir une feuille de calculs préremplie comme ci-dessus et demander aux élèves de compléter les colonnes C et D avec des fractions ayant le même dénominateur par ligne.

La colonne E permet de vérifier.

A noter la formule saisie en F6 ...

On peut imaginer de nombreuses autres activités. Christophe Devalland, le développeur de CmathsOOoCAS en propose de très nombreuses sur son site cdeval.free.fr.

4) Manipuler les nombres complexes

On considère la suite (z_n) définie par z_0 et pour tout $n \in \mathbb{N}$, $z_{n+1} = i z_n - 4$.

Écrire une fonction personnelle pour calculer les termes de la suite, puis conjecturer l'expression de z_n lorsque $z_0 = 1+i$.

Notez que i est représenté dans Sympy par I et que dans le tableur, on a saisi "1+i"

Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès. Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles: EXO

Créer une nouvelle fonction utilisant Sympy

```
def EXO(z,n):
    for a in range(1,n+1):
        z = z*I-4
    return expand(z)
```

EXO(1+i;3)

Messages du CAS:

EXO(1+i;3)
-> 1-5i

Compilation EXO -> succès

EXO.py:

```
1: from sympy import *
2: x = Symbol('x')
3:
4: def EXO(z,n):
5:     z = sympify(z)
6:     n = sympify(n)
7:     for a in range(1,n+1):
8:         z = z*I-4
9:     return expand(z)
```

=EXO("1+i";C1)

	D
1	-5+i
2	-5-5i
3	1-5i
4	1+i
5	-5+i
6	-5-5i
7	1-5i
8	1+i

5) Trouver un polynôme minimal

Voici comment faire :

On saisit la matrice dans Calc

	A	B	C
1	1	2	-2
2	2	1	-2
3	2	2	-3

On la mémorise dans Writer

=SAUVEMATRICE("A";A1:C3)

D	E
A	

On l'écrit en tapant A puis Alt+C

$$A = \begin{pmatrix} 1 & 2 & -2 \\ 2 & 1 & -2 \\ 2 & 2 & -3 \end{pmatrix}$$

=POLCAR(A1:C3)

D	E
	$(x-1)(x+1)^2$

On crée la fonction python comme ci-dessous.

eye(3) désigne la matrice unité d'ordre 3

Fonctions personnelles: f

Créer une nouvelle fonction utilisant Sympy

```
def f(x):
    return x^2-eye(3)
```

On calcule f(A) :

{=F(A1:C3)}

D	E	F
0	0	0
0	0	0
0	0	0

Son polynôme caractéristique est donc $p(x) = (x-1)(x+1)^2$

Le polynôme minimal est $p(x)$ ou $(x-1)(x+1) = x^2 - 1$

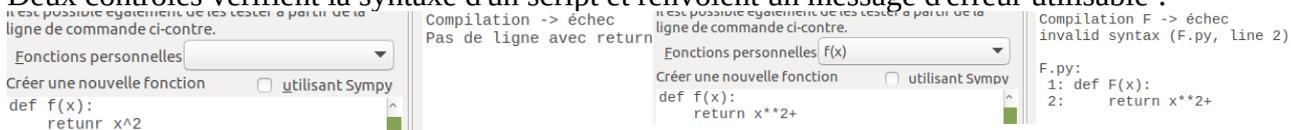
Donc f(A) est la matrice nulle. $x^2 - 1$ est le polynôme minimal.

[Retour accueil](#)

DÉBOGUEUR UNE FONCTION PERSONNELLE

1) Erreurs de syntaxe

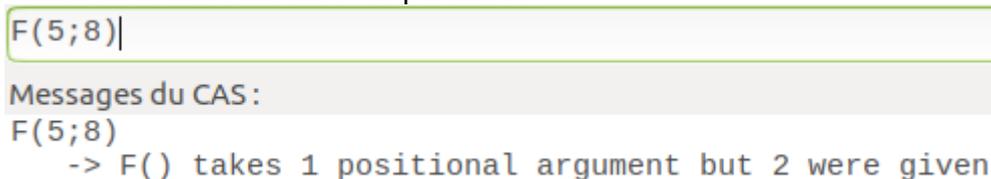
Deux contrôles vérifient la syntaxe d'un script et renvoient un message d'erreur utilisable :



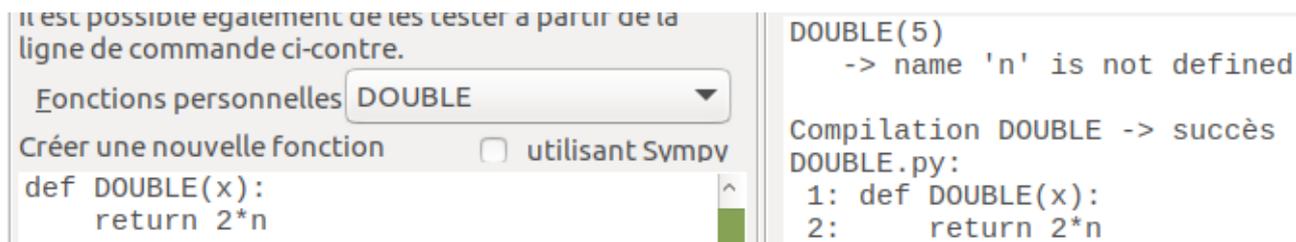
On corrige en changeant « retunr » en « return » On corrige en enlevant le « + »

2) Erreurs d'exécution

a) On peut donner un nombre incorrect de paramètres



b) Une fonction personnelle compilée avec succès peut malgré tout contenir un bogue.
Considérons la fonction DOUBLE ci-dessous. On obtient une erreur en tapant DOUBLE(5)



Le problème vient de « n » qui n'est pas défini. Il faut donc changer « return 2*n » en « return 2*x »
Un clic sur le bouton afficher permet de corriger dans l'éditeur de fonctions de Dmaths Addin.
Nous recompilons et cette fois-ci cela fonctionne.

Nous pourrions également éditer la fonction Python en cliquant sur le bouton  et tester dans notre EDI favori (par exemple Thonny)



3) La fonction msgbox

Elle permet d'afficher des valeurs intermédiaires pendant l'exécution du script.

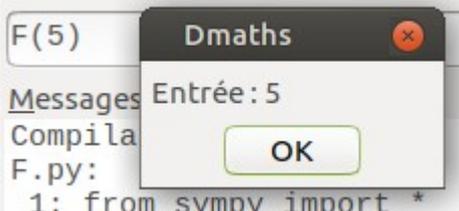
Voyez un exemple d'utilisation :

Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès.
Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles F

Créer une nouvelle fonction utilisant SymPy

```
def F(x):  
    msgbox('Entree : ' + str(x), 'Dmaths')  
    msgbox('Sortie : ' + str(f(x)), 'Dmaths')  
    return f(x)  
  
def f(x):  
    return x^2
```

Messages du CAS:
Compilation F -> succès
F.py:
1: from Konsole import msgbox
2: from sympy import *
3: x = Symbol('x')
4:
5: def F(x):
6: x = sympify(x)
7: msgbox('Entree : ' + str(x), 'Dmaths')
8: msgbox('Sortie : ' + str(f(x)), 'Dmaths')
9: return f(x)
10:
11: def f(x):
12: return x**2

[Retour accueil](#)

LA CONSOLE APSO

Elle permet d'utiliser DmathsAddin comme un mini IDE de développement Python.

On l'affiche en cliquant sur le bouton .

Les fonctions personnelles sélectionnées, msgbox ainsi que sympy sont importées.

Par contre une variable alphanumérique doit être définie : « $x = \text{Symbol}('x')$ ».



Commandes Dmaths : CABS

Commandes CAS : expand

Editeur de fonctions

Les fonctions définies ici seront disponibles dans le tableur une fois qu'elles auront été compilées avec succès.

Il est possible également de les tester à partir de la ligne de commande ci-contre.

Fonctions personnelles F

Créer une nouvelle fonction utilisant Sympy

```
def F(x):  
    return x^2
```

APSO python console by Dmaths Addin [LibreOffice]
3.8.10 (default, Sep 9 2021, 12:10:03)
[GCC 7.3.1 20180303 (Red Hat 7.3.1-5)]
Type "help", "copyright", "credits" or "license" for more information.
>>> 2+3
5
>>> F(5)
25
>>> F('a')
a**2
>>> x = Symbol('x')
>>> F(x)
x**2
>>> expand((x+3)**2)
x**2 + 6*x + 9
>>> factor(x**2+6*x)
x*(x + 6)
>>> isprime(73)
True
>>> factorint(126)
{2: 1, 3: 2, 7: 1}
>>> divisors(126)
[1, 2, 3, 6, 7, 9, 14, 18, 21, 42, 63, 126]
>>> msgbox('C'est fini !', 'Dmaths Addin')

Dmaths Addin

C'est fini !

OK

[Retour accueil](#)

ASSURER LA PORTABILITÉ

1) Les feuilles de calcul avec formules Dmaths Addin

Vous avez besoin d'utiliser une feuille de calcul sur une machine qui n'a pas Dmaths Addin

installée : cliquer sur le bouton jaune **C** :

- les formules venant d'Excel seront converties au format Calc.
- la feuille active sera alors dupliée avec les résultats sans les formules.

Avant				Après			
B	C	D	E	B	C	D	E
3x	-x+1	2x+1	True	3x	-x+1	2x+1	True
			1/4				1/4

D	E	F	G
		-2sqrt(3)	
		2sqrt(3)	

Feuille1

D	E	F
		-2sqrt(3)
		2sqrt(3)

Feuille1 Feuille1_copie

2) Les fonctions personnelles

Le bouton **Exporter** de la boîte de dialogue permet d'enregistrer facilement le script Python d'une fonction personnelle, par exemple sur une clé USB.

```
Fonctions personnelles DOUBLE
Créer une nouvelle fonction  utilisant Sympy
def double(x):
    return 2*x
```

le script dans Dmaths Addin

```
double.py x
1 def double(x):
2     return 2*x
```

le script exporté

[Retour accueil](#)

GESTION DES VARIABLES MÉMORISÉES

Voyons comment gérer les variables mémorisées dans le CAS (Writer) et dans l'Addin (Calc).

1) Cas des variables numériques et des fonctions

Mémorisons une variable numérique a et une fonction f dans Writer.

On tape $a=12$ suivi de F10, puis $f(x)=x^2$ suivi de F10.

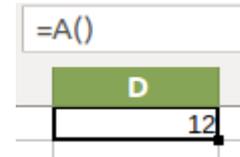
Au lancement de la boîte de dialogue de DmathsAddin On clique sur compiler pour définir une fonction dans Calc

La fonction est disponible dans Calc

```
Fonctions personnelles a
Créer une nouvelle fonction
def a():
    return 12
```

```
a
Messages du CAS :
a
-> 12

A()
-> 12
```



```
Fonctions personnelles f(x)
Créer une nouvelle fonction
def f(x):
    return x**2
```

```
f(a)
-> 144

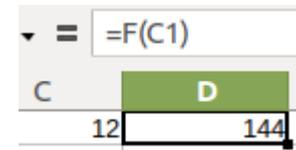
f(5)
-> 25

f(x)
-> x^2

F(a)
-> a^2

F(5)
-> 25

F(x)
-> x^2
```

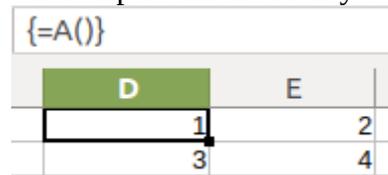
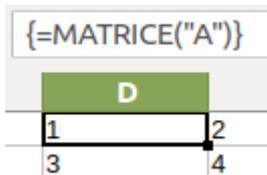


2) Cas des matrices mémorisées dans le CAS (Writer)

$A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$. On peut afficher A dans Calc de deux façons différentes :

Avec la fonction « matrice »

En affichant A dans la boîte de dialogue et en créant un fonction personnelle A . Il y a alors deux variables A

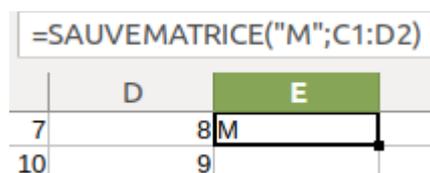


```
A
-> Matrix([[1;2];[3;4]])

A()
-> [[1;2];[3;4]]
```

3) Mémoriser une matrice écrite dans Calc dans le CAS (Writer)

On utilise la fonction « SAUVEMATRICE »



Dans Writer on tape M suivi de Alt+C (Ctrl+F10 sous Mac)

$$M = \begin{pmatrix} 7 & 8 \\ 10 & 9 \end{pmatrix}$$

[Retour accueil](#)

**Dmaths pour LibreOffice >= 7.4.4 ou OpenOffice >= 4.1
version 7.1**

Copyright (C) 2006-2024 Didier DORANGE-PATTORET

38, chemin de l'Abbaye 74940 Annecy.

mail: ddorange@dmaths.org.

Ce programme est libre, vous pouvez le redistribuer et/ou le modifier selon les termes de la Licence Publique Générale GNU publiée par la Free Software Foundation (version 3 ou bien toute autre version ultérieure choisie par vous). Pour en savoir plus ouvrir un nouveau fichier texte, taper copie puis F3.

Ce programme est distribué car potentiellement utile, mais SANS AUCUNE GARANTIE, ni explicite ni implicite, y compris les garanties de commercialisation ou d'adaptation dans un but spécifique. Reportez-vous à la Licence Publique Générale GNU pour plus de détails. Pour en savoir plus taper garantie puis F3.

Vous devez avoir reçu une copie de la Licence Publique Générale GNU en même temps que ce programme ; si ce n'est pas le cas, écrivez à la Free Software Foundation, Inc., 59 Temple Place, Suite 330, Boston, MA 02111-1307, États-Unis.

La licence est consultable dans le fichier joint **licence.odt** ou sur www.fsf.org

[Retour accueil](#)