

PluriAlgo et la version en ligne de Xcas

Introduction

La démarche illustrée avec AlgoBox dans l'article peut être reprise en l'état avec la version standard de Xcas :

Editeur 1 Editeur 2 Aide Execution

```
saisir("quantite : ",quantite);  
saisir("prixUnitaire : ",prixUnitaire);  
afficher(prixTotal);
```

Principal Si Boucles Console

choix du langage
langage Xcas

entrees quantite prixUnitaire

sorties prixTotal

TYPES Entiers Reels
Simple quantite prixUnitaire prixTotal

Nouveau Insérer Effacer

calculs elementaire

Pour les langages scientifiques, deux autres options de calcul (fonctions et proc_entrees) sont disponibles

Ce code est malheureusement inexploitable pour la version en ligne de Xcas, que je trouve plus conviviale que la version standard :

- la primitive « saisir » n'est pas disponible
- un programme doit être lancé par une commande, ce qui conduit à définir auparavant un sous-programme si l'algorithme à traiter ne se limite pas à un simple calcul.

Heureusement, PluriAlgo facilite l'introduction de sous-programmes : il suffit de modifier la valeur de l'option « calculs » en choisissant « proc_entrees » ou « fonctions ».

L'option de calcul « proc_entrées »

Je vais partir du programme final testé avec la version en ligne de Xcas, avant de le commenter et de montrer comment il peut être obtenu avec PluriAlgo :

Programmation

Taper votre programme dans la fenêtre du bas, puis lancez le dans la console (en haut).

Quelques notions de programmation sont données ci-dessous dans l'esprit du nouveau programme de seconde.

Attention : l'instruction "saisir" et le débogueur (qui permet de suivre l'exécution d'un programme pas à pas) ne sont pas disponibles sous Xcas en ligne. Téléchargez le "vrai" Xcas pour les utiliser.

On colle ici le code Xcas obtenu avec PluriAlgo, puis on le complète

Xcas en ligne. Tapez une instruction dans cette console (assistant avec la bouée).

`calculer(2, 10.5)` — commande 1
Impression :
21

`calculer(4, 10.5)` — commande 2
Impression :
37.8

```
//Tapez votre programme ici :
calculer(quantite, prixUnitaire):={
  local prixTotal;
  prixTotal:=quantite*prixUnitaire;
  si (quantite>=3) alors // remise 10%
    prixTotal:=prixTotal-0.1*prixTotal;
  fsi;
  afficher(prixTotal);
};
```

A ajouter

Pour expliquer les notations aux élèves (intuitivement, sans théoriser sur les sous-programmes), on peut leur dire ceci :

- l'algorithme a pour nom « calculer » (on peut changer ce nom)
- les entrées sont indiquées dans les parenthèses suivant le nom de l'algorithme
- l'algorithme peut être lancé plusieurs fois dans la console comme n'importe quelle commande prédéfinie

Le code Xcas peut être obtenu simplement de la façon suivante :

- complétez l'onglet Principal de PluriAlgo comme indiqué dans l'introduction
- remplacez l'option de calcul **élémentaire** par **proc_entrées**
- cliquez sur le bouton **Insérer** (de préférence au bouton **Nouveau**, pour éviter de produire une portion de code inutile dans ce contexte)
- ajoutez l'instruction conditionnelle (par exemple avec l'onglet Si, voir documentation du logiciel), puis les deux instructions d'affectation

L'option de calcul « fonctions »

Lorsqu'un algorithme n'a qu'une sortie (ici prixTotal), il est préférable de remplacer le sous-programme **affichant** la sortie par un sous-programme **retournant** la sortie :

The screenshot shows the Xcas online programming interface. On the left, there is a sidebar with the title "Programmation" and instructions: "Taper votre programme dans la fenêtre du bas, puis lancez le dans la console (en haut).", "Quelques notions de programmation sont données ci-dessous dans l'esprit du nouveau programme de seconde.", and "Attention : l'instruction 'saisir' et le débogueur (qui permet de suivre l'exécution d'un programme pas à pas) ne sont pas disponibles sous Xcas en ligne. Téléchargez le 'vrai' Xcas pour les utiliser." Below this, it says "On colle ici le code Xcas obtenu avec PluriAlgo, puis on le complète".

The main console area is titled "Xcas en ligne. Tapez une instruction dans cette console (assistant avec la bouée)." and shows the following interaction:

```
calculer( 2 , 10.5)
                21
calculer( 2 , 10.5)*1.25
                26.25
```

Red annotations highlight the function call `calculer(2 , 10.5)*1.25` and the result `26.25`. A red box around the function definition in the code editor contains the text "montre l'intérêt d'une fonction (on peut combiner les calculs)".

The code editor shows the following code:

```
//Tapez votre programme ici :
calculer(quantite, prixUnitaire):={
  local prixTotal;
  prixTotal:=quantite*prixUnitaire;
  si (quantite>=3) alors // remise 10%
    prixTotal:=prixTotal-0.1*prixTotal;
  fsi;
  retourne (prixTotal);
};
```

Red annotations highlight the function definition and the `retourne (prixTotal);` line. A blue arrow points to the `retourne (prixTotal);` line with the text "A ajouter".

L'avantage de la fonction est qu'on peut maintenant réutiliser le résultat pour le convertir en dollars (taux de conversion euros-dollars 1.25 à adapter).

Le code du sous-programme peut être obtenu comme dans la section précédente avec PluriAlgo, en remplaçant l'option de calcul **proc_entrees** par **fonctions**.