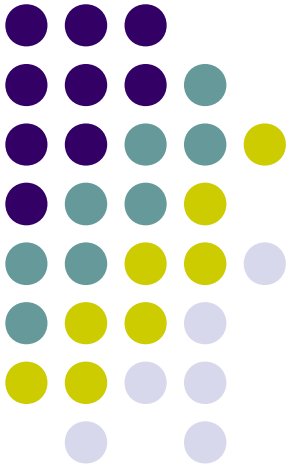
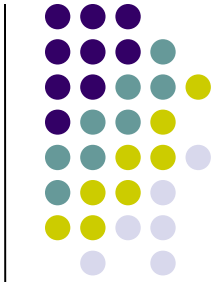
Outils informatiques 2
Cours n°6: Fonctions et paramètres

Claire Hanen

Juliette Arnal

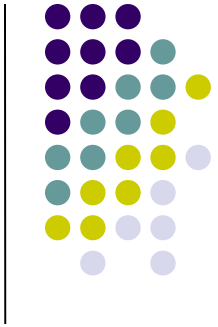


Plan



- Introduction à la notion de fonction, comparaison entre fonctions mathématiques et fonctions informatique
- En-tête de fonction
- Syntaxe d'appel par paramètres listés
- Syntaxe d'appel par paramètres nommés, fonctions complètes InputBox et MsgBox
- Définir soi-même des fonctions avec paramètres: comment, pourquoi?

Fonction en mathématiques

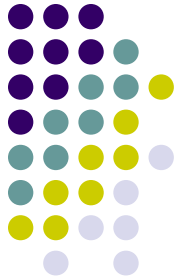


- Exemple:

$$F: IR \times IR \rightarrow IR$$
$$F(x,y)=(3x+2)/(2y-8)$$

- Une fonction mathématique est décrite par :
  - Un symbole, le nom de la fonction (ici F)
  - Un espace de départ D (ici IR X IR)
  - Un espace d'arrivée A (ici IR)
  - Une expression décrivant, à l'aide d'un symbole (ici (x,y)) représentant un élément quelconque de D, la manière dont est calculé l'élément de A associé à (x,y) par la fonction F.
- Pour calculer la valeur d'une fonction en un point, par exemple F(3,2) on remplace, dans l'expression, le symbole x par 3 et le symbole y par 2: (3*3+2)/(2*2-8)=-11/4

Notion de fonction en informatique : similitudes



- Une fonction informatique est décrite par :
  - Un symbole, le nom de la fonction (défini par le programmeur, ou prédéfini)
  - Un espace de départ D, qui représente un produit cartésien de types.
  - A chaque type est associé un symbole (comme x ou y) qui s'appelle un paramètre formel.
  - L'espace d'arrivée est décrit en définissant le type du résultat de la fonction (on parle de valeur retournée par la fonction)
  - Au lieu de décrire une expression, on décrit un programme permettant de calculer la valeur de la fonction à partir des valeurs de ses paramètres.
  - La fonction est appelée dans d'autres programmes comme une expression. L'appel spécifie des valeurs (expressions) pour les paramètres de la fonction. On parle de paramètres réels.
  - Comme précédemment, ces valeurs sont substituées aux valeurs des paramètres formels, et le programme de la fonction s'exécute.

Exemple de fonction définie par le programmeur (notre objectif)

Nom de la fonction

Liste des paramètres formels et leur types (espace de départ)

- Function ftest(x as double, y as double) As double
- Dim a as double, b as double
- a=3*x+2
- b=2*y-8
- If b<>0 then
- ftest=a/b
- Else
- ftest= 10000
- End if
- End Function

programme de calcul de la fonction

Type du résultat de la fonction

Appel de la fonction

```
Sub test_ftest()
MsgBox(ftest(3,2))
End sub
```

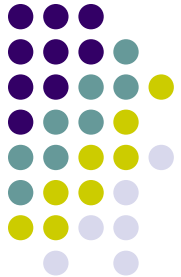
Affectation d'une valeur au résultat de la fonction

3 et 2 sont les paramètres réels

L’en-tête de la fonction

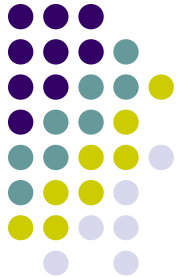
- C’est la première ligne de la déclaration d’une fonction
- Function nom(liste de paramètres formels typés) As typeduresultat
- Cet en-tête est défini pour les fonctions que le programmeur écrit, mais aussi pour les fonctions préexistantes.
- Il définit le mode d’emploi d’une fonction (celui qu’on trouve dans l’aide en ligne)

Retour sur des fonctions prédéfinies



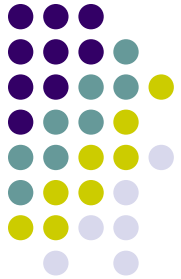
- Function Len (string As String) As long
  - Nb: long est un type comme integer, mais avec plus de nombres représentés
  - Appel: Len(“bonjour “)
- Function Mid (string As String, start As long, length as variant) As String
  - Appel : Mid(“bonjour “, 3,4) (vaut “njou “)

Appel d'une fonction par paramètres listés



- Lorsqu'une fonction est définie par le programmeur ou pré-définie, on peut l'appeler :
- Soit depuis un programme VBA dans une expression:
  - Nomfonction(liste de paramètres réels)
  - Exemples: ftest(3,2) ou Mid(mot, i,1)
  - La liste de paramètres réels est une suite d'expressions séparées par des virgules dont les types correspondent aux types des paramètres formels (dans le même ordre). Le type de l'expression est celui du résultat de la fonction.
- Soit depuis Excel (seulement pour les fonctions définies par le programmeur), en employant la syntaxe tableur:
  - Le nom de la fonction est en majuscules
  - Les paramètres réels sont séparés par des points-virgules.
  - Exemple: dans la cellule C1: =FTEST(A1;B1)
- Ce mécanisme permet d'étendre les fonctionnalités du tableur

Spécificités des fonctions en informatique

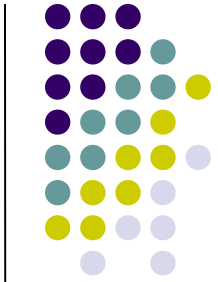


- Il existe des fonctions sans paramètres.
  - Ce ne sont donc pas des fonctions au sens mathématique du terme (pas d'espace de départ).
  - Cependant le mécanisme est le même que précédemment: l'appel à la fonction déclenche l'exécution des instructions correspondantes et le calcul du résultat de la fonction.
  - Exemple: la fonction Rnd(): fait appel au générateur aléatoire de nombres, et produit un nombre entre 0 et 1 tiré au hasard.

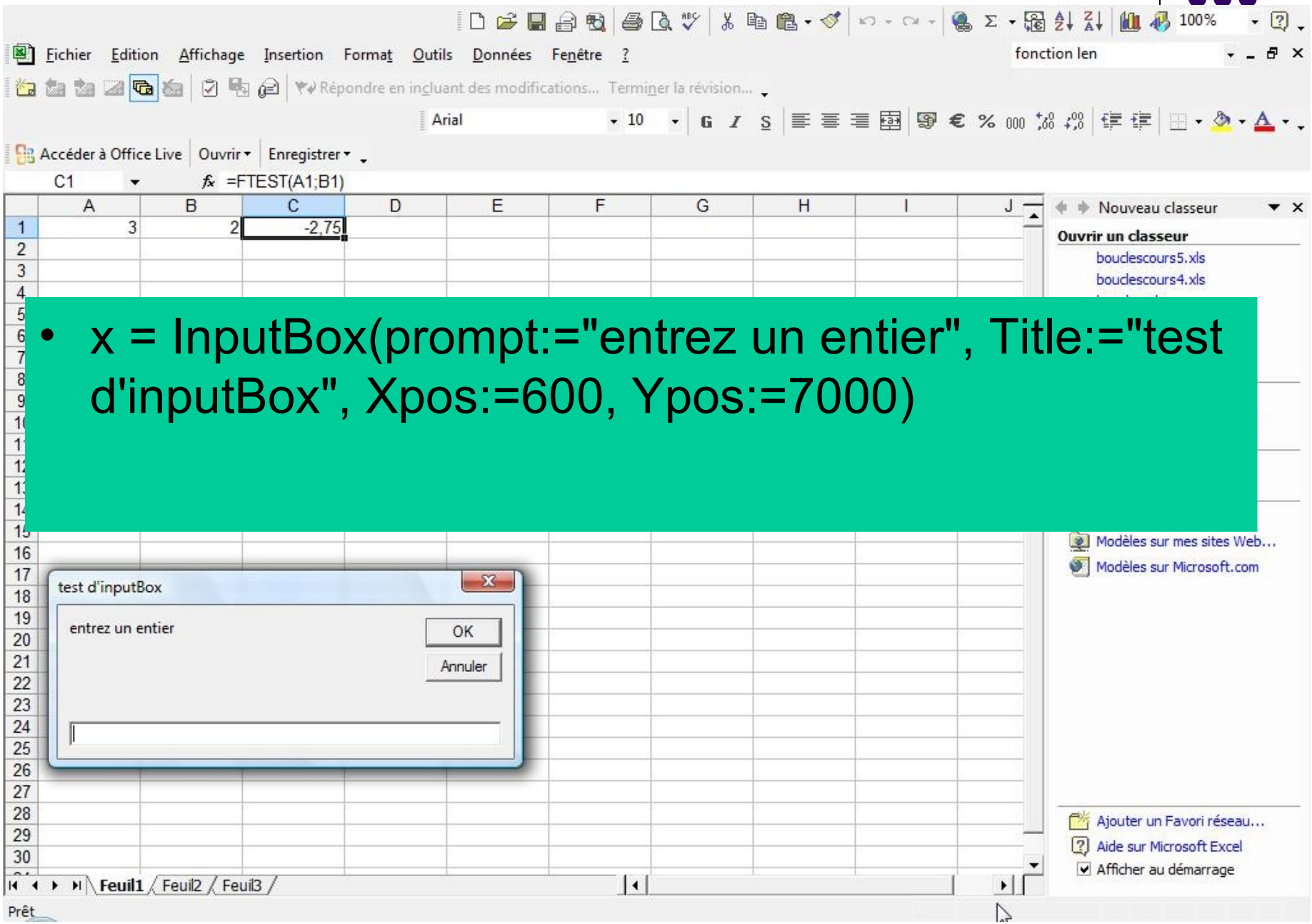
Spécificités des fonctions en VBA

- Les fonctions ont en fait deux types de paramètres:
  - Les paramètres obligatoires
  - Les paramètres facultatifs.
- Dans l'en-tête d'une fonction prédéfinie (aide en ligne), les paramètres facultatifs sont figurés entre crochets [ ]
- Dans l'en-tête d'une fonction définie par le programmeur, les paramètres facultatifs sont précédés du mot Optional
  - Exemple f(x as double [,y as integer] [,z as integer]) as double
  - Function f(x as double, optional y as integer, optional z as integer) as double
- Un paramètre facultatif peut être omis (il a en général une valeur par défaut).
- Problème pour la syntaxe de l'appel: si on liste des paramètres réels et que certains sont omis et pas d'autres, comment l'ordinateur peut-il reconnaître à quel paramètre formel est associé un paramètre réel?
  - Exemple f(3,2): 2 doit-il se substituer à y ou à z dans le calcul?
- Une autre syntaxe d'appel pour traiter ce cas: la syntaxe par paramètres nommés

Exemple: la fonction InputBox

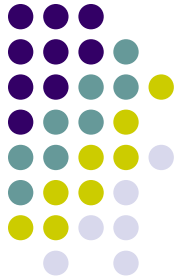


- En-tête (trouvée dans l'aide en ligne)
  - Function InputBox(prompt As string[, title As string] [, default As string] [, Xpos As integer] [,Ypos As integer] [, helpfile, context])
- prompt Expression de chaîne affichée comme message dans la boîte de dialogue
- title Facultatif. Expression de chaîne affichée dans la barre de titre de la boîte de dialogue.
- default Facultatif. Expression de chaîne affichée par défaut dans la zone de texte en l'absence de toute autre valeur
- Xpos: Facultatif. Expression de type entier désignant la position (en points) de la boite de dialogue sur l'axe horizontal
- Ypos: Facultatif. Expression de type entier désignant la position (en points) de la boite de dialogue sur l'axe vertical



- x = InputBox(prompt:="entrez un entier", Title:="test d'inputBox", Xpos:=600, Ypos:=7000)

Syntaxe par paramètres nommés



- La liste de paramètres réels est constituée:
  - De chaque nom de paramètre formel choisi suivi du signe :=, puis de l'expression donnant le paramètre réel.
  - Les paramètres sont séparés par des virgules.
  - Les paramètres obligatoires doivent évidemment être présents
  - L'ordre des paramètres n'a pas d'importance
  - Les parenthèses peuvent être omises
- Exemples
  - x = InputBox prompt:="entrez un entier", Title:="test d'inputBox", Xpos:=600, Ypos:=7000
  - x= InputBox Default:=0 , Prompt:="entrez un entier"

La fonction MsgBox complète

- En fait, MsgBox est une fonction (et pas une macro) prédéfinie.
- Dans sa version complète elle permet de gérer l'interaction avec l'utilisateur au travers de boutons.
- Paramètres principaux :
- *Prompt* : Message affiché (seul paramètre obligatoire)
- *Title* :Facultatif titre de la boite de dialogue, Xpos, Ypos (comme pour InputBox
- *Buttons* : Identification des boutons à afficher
- Le type de la fonction est Byte (valeur du bouton cliqué par l'utilisateur)

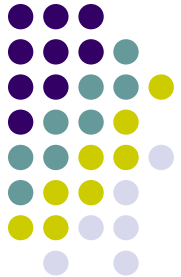
Boutons affichés	**Constante prédéfinie**	**valeur**
OK	vbOKOnly	0
OK/Annuler	vbOKCancel	1
Oui,Non, Annuler	vbYesNoCancel	3
Oui, Non	vbYesNo	4
Réessayer, Annuler	vbRetryCancel	5

Bouton choisi	**Constante renvoyée**	**valeur**
OK	VbOK	1
Annuler	VbCancel	2
Réessayer	VbRetry	4
Oui	VbYes	6
Non	VbNo	7

Exemple 1

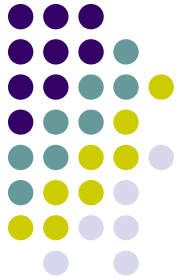
- Sub testmsgbox()
- Dim valeur As String
- Dim reponse As Byte
- reponse = MsgBox prompt:="voulez-vous jouer?", Title:="test de MsgBox", Buttons:=vbYesNo
- If reponse = vbYes Then
- MsgBox prompt:="tant pis !", Title:="le jeu est fini!"
- Else
- MsgBox prompt:="tant mieux!", Title:="le jeu va commencer"
- End If
- End Sub

Exemple 2: utilisation dans le jeu de l'addition



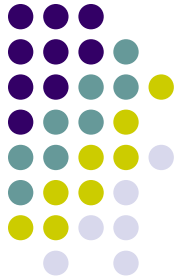
- Do
- x = Int(Rnd() * 100) + 1
- y = Int(Rnd() * 100) + 1
- nbadd = nbadd + 1
- nb = 0
- score = 5
- z = InputBox(prompt:="combien font " & x & " et " & y & " ?", Title:="jeu de l'addition")
- Do Until (nb = 5) Or (z = x + y)
- score = score - 1
- z = InputBox(prompt:="Erreur, encore un essai: combien font " & x & " et " & y & " ?", Title:="jeu de l'addition")
- nb = nb + 1
- Loop
- If (z = x + y) Then
- MsgBox prompt:="Bravo!", Title:="jeu de l'addition"
- Else
- MsgBox prompt:="La bonne réponse était: " & x + y, Title:="jeu de l'addition"
- End If
- scoreglobal = scoreglobal + score
- Loop Until (MsgBox(prompt:="voulez-vous une autre addition?", Title:="jeu de l'addition", Buttons:=vbYesNo) = vbNo)

Ecrire soi même des fonctions: pour quoi faire?



- Enrichir les fonctions prédéfinies du logiciel
- Pouvoir rendre plus lisible des programmes, en nommant un calcul fait fréquemment
- Pouvoir réutiliser les fonctions dans d'autres programmes.

Syntaxe de déclaration d'une fonction

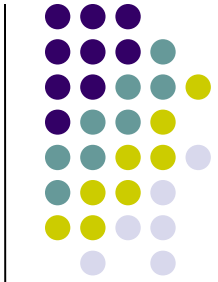


- Function nomfonc(liste de paramètres formels typés) As typedelafonction
- Instructions
- End Function
- Les instructions peuvent être de n'importe quelle sorte. Les paramètres formels sont utilisé comme des variables.
- Il est nécessaire que les instructions comportent au moins une fois une instruction :
  - nomfonc=expression
- C'est cette instruction qui permet de donner une valeur à la fonction.
- L'appel à la fonction dans une macro peut ensuite être faite à l'aide de l'une des deux syntaxes d'appel (listés ou nommés) au choix.

Somme des carrés des entiers

- On souhaiterait définir une fonction qui possède un seul paramètre (un nombre entier) et qui calcule la somme des carrés des entiers de 1 à ce paramètre.
- On donne un nom à la fonction: somcar
- On définit son en-tête:
- Function somcar(max As integer) As integer
- (le résultat est un entier, et on nomme le paramètre)
- ATTENTION: dans cette démarche, la fonction ne doit pas faire d'entrée-sortie (de dialogue avec l'utilisateur). Elle est juste là pour nommer un calcul

Code de somcar



- Function somcar(max as integer) as integer
- Dim n as integer, s as integer
- s=0
- For n=1 to max
- s=s+n*n
- Next n
- somcar=s
- End Function

Appel de la fonction somcar

- Pour utiliser la fonction somcar, on peut l'appeler depuis une macro:

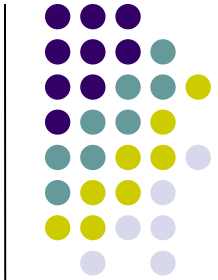
```
Sub testsomcar()
Dim e as integer
e=InputBox Title:="test", prompt:="entrez un entier"
e=somcar(e)
MsgBox(Title:="test", prompt:= e)
End Sub
```

- On peut aussi l'employer dans la feuille Excel contenant la macro:
- =SOMCAR(B5) par exemple

Nombre de combinaisons

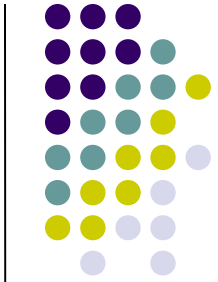
- Programme vu la fois précédente:
- Sub Cnp()
- Dim n As Integer
- Dim p As Integer
- Dim i As Integer
- Dim res As Double
- n = InputBox("entrez un entier")
- p = InputBox("entrez un entier inférieur à " & n)
- res = 1
- For i = 0 To p - 1
- res = res * (n - i) / (p - i)
- Next
- MsgBox ("Le nb de tirages de " & p & " éléments parmi " & n & " est : " & res)
- End Sub

Extraire le calcul pour en faire une fonction.



- Function combi(n as Integer, p as integer) as double
- Dim i As Integer
- Dim res As Double
- res = 1
- For i = 0 To p - 1
- res = res * (n - i) / (p - i)
- Next
- combi=res
- End Function

Modification de la macro d'affichage



- Sub Cnp()
- Dim n As Integer
- Dim p As Integer
- n = InputBox("entrez un entier")
- p = InputBox("entrez un entier inférieur à " & n)
- MsgBox ("Le nb de tirages de " & p & " éléments parmi " & n & " est : " & combi(n, p))
- End Sub

Résolution triangle de Pascal avec utilisation de la fonction

```
Sub Pascal()
Dim max As Integer, lig As Integer, col As Integer
Dim triangle As String
max = InputBox("Entrez le nombre de lignes du triangle")
triangle = ""
For lig = 1 To max  ‘ calcul de la ligne lig
    For col = 0 To lig  ' calcul de C(lig,col) pour le lig et le col fixé
        triangle = triangle & " " & combi(lig,col)  ‘ On l'ajoute au triangle
    Next col
triangle = triangle & vbCr ‘On passe à la ligne dans la chaine triangle
Next lig
MsgBox (triangle)
End Sub
```