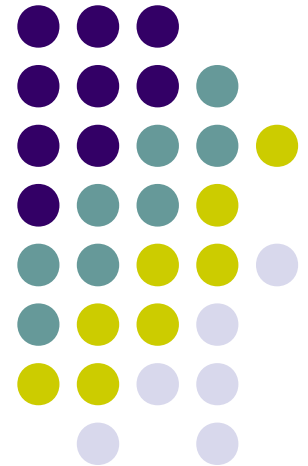
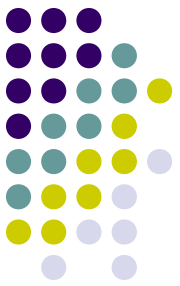


Outils informatiques 2

Claire Hanen
Juliette Arnal

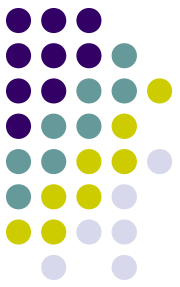




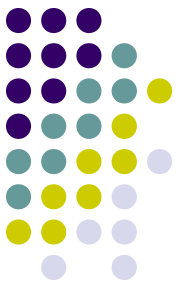
Déroulement du cours

- Cours magistral:
 - Support de travail synthétique sur Internet (page Hanen sur www.u-paris10.fr)
 - Tous les exemples et les explications détaillées sont données en cours.
- Travaux dirigés:
 - Les supports sont à télécharger au même endroit
 - Les TDs se déroulent sur machine
 - Nécessitent Word/Excel 2003 (ou postérieurs)
- Contrôle des connaissances
 - Examen terminal en amphi : 50%
 - Contrôles en TD: interrogation, relevés de TD ou devoirs, contrôle sur machine.50%

Un Langage de programmation

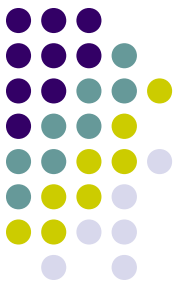


- ***Pour quoi faire ?***
 - Maîtriser les outils informatiques
 - Mieux comprendre la structure de l'information manipulée par un logiciel.
 - Pouvoir développer des petites applications personnalisées.
- ***Environnement de travail choisi***
 - Un document Word ou Excel (2003)
 - Les bibliothèques de programmes et d'informations qui permettent de modifier le document ;
 - Environnement de programmation : débogueur, aide en ligne



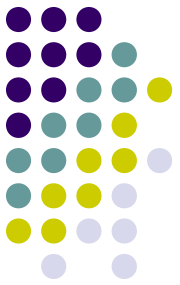
Plan du cours

- Introduction à VBA, macros
- Éléments de base pour démarrer (variables, affectation, entrées-sorties élémentaires)
- Types, opérateurs, expressions, conversion de types
- Structures de contrôle alternatives
- Structures de contrôle répétitives
- Fonctions et procédures, notion de paramètre
- Définir des fonctions personnalisées de calcul, de jeu
- La représentation et la manipulation des données du logiciel Excel
 - Notion d'objet, de collection
 - Feuilles, cellules, fonctions
- Définir des fonctions et macros de plage de cellules

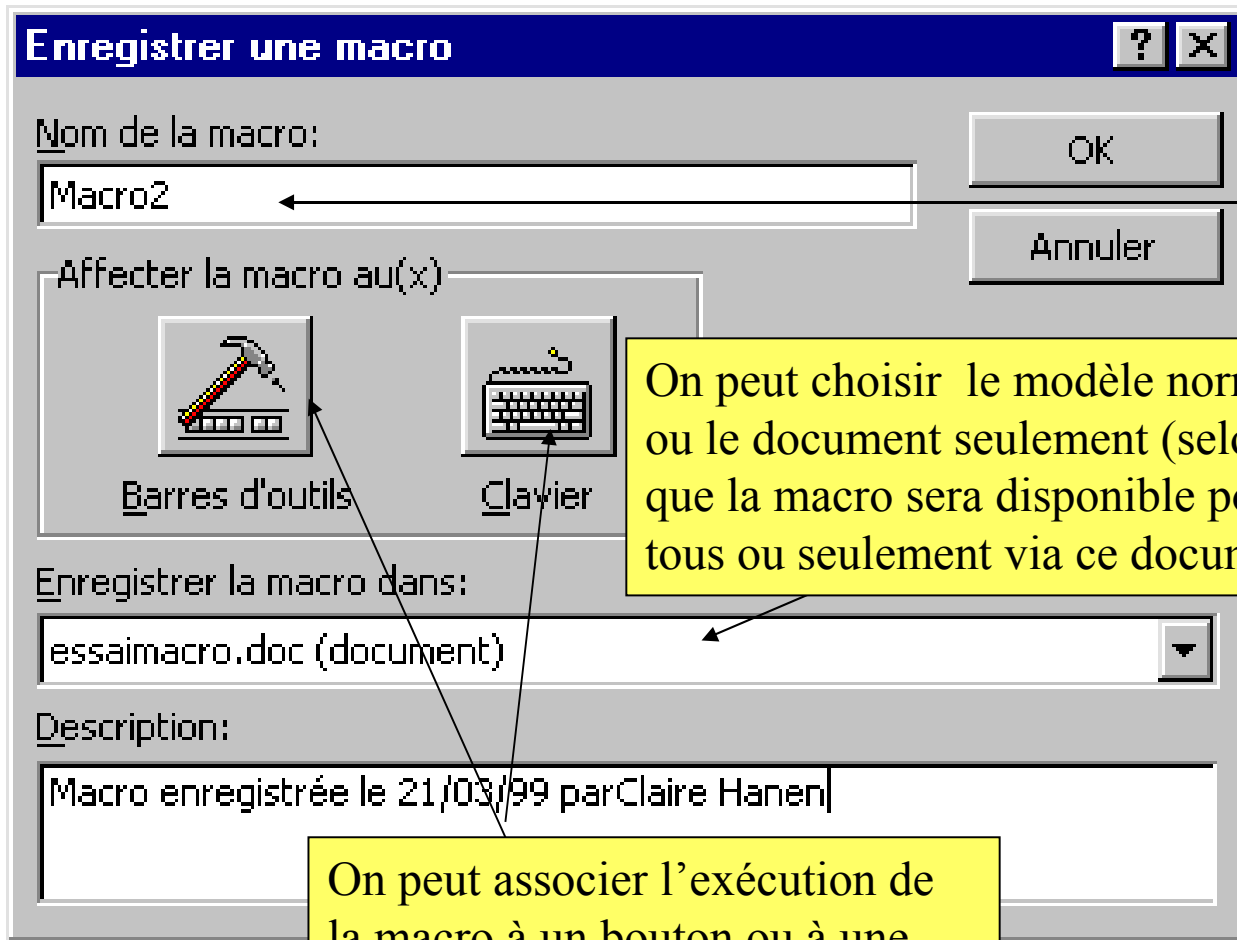


Les Macros

- **Définition :**
 - Une macro est la mémorisation d'une suite de commandes d'un logiciel.
 - Une macro correspond à un sous-programme qui peut être écrit dans un langage de programmation et qui est exécutable à la demande de l'utilisateur.
 - Ce sous-programme possède un **nom** défini par l'utilisateur.
 - Il peut être associé à un bouton de la barre d'outils ou à une touche du clavier.
 - Dans tous les cas il est exécutable par le **menu Outils-Macro**
- **Enregistrer une macro**
 - Pourquoi faire ? : automatiser des tâches répétitives.
 - Exemple : enregistrer un document sur la clé usb et sur le disque dur.



Enregistrement et exécution

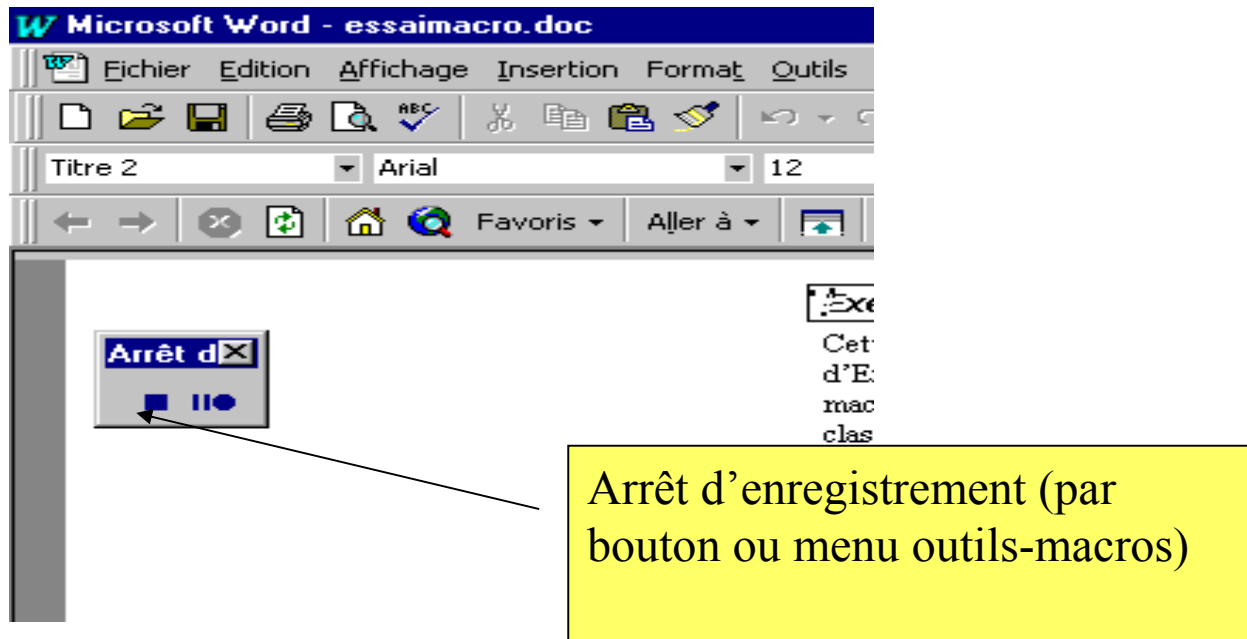
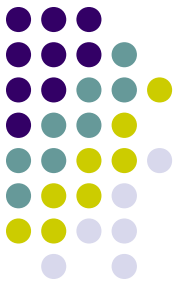


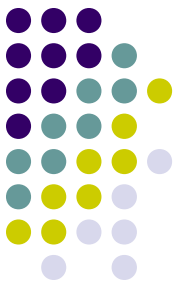
On choisit le nom de la macro

On peut choisir le modèle normal ou le document seulement (selon que la macro sera disponible pour tous ou seulement via ce document)

On peut associer l'exécution de la macro à un bouton ou à une touche

Arrêt d'enregistrement





Un exemple sous Word

- Enregistrer un document dans deux dossiers successivement –Mes documents et transfert.

Nom de la macro :
enregistrerspecial

Affecter la macro au(x) :

Barres d'outils Clavier

Enregistrer la macro dans :
Document2 (document)

Description :
Macro enregistrée le 03/02/2010 par claire hanen

OK Annuler

Sub enregistrerspecial()

```
'  
' enregistrerspecial Macro  
' Macro enregistrée le 03/02/2010 par claire hanen  
'
```

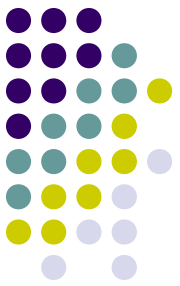
```
ActiveDocument.SaveAs FileName:="essai.doc", FileFormat:=wdFormatDocument, _  
LockComments:=False, Password:="", AddToRecentFiles:=True, WritePassword _  
:= "", ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _  
SaveNativePictureFormat:=False, SaveFormsData:=False, SaveAsAOCELetter:= _  
False
```

```
ChangeFileOpenDirectory "C:\Users\claire\Documents\transfert\"
```

```
ActiveDocument.SaveAs FileName:="essai.doc", FileFormat:=wdFormatDocument, _  
LockComments:=False, Password:="", AddToRecentFiles:=True, WritePassword _  
:= "", ReadOnlyRecommended:=False, EmbedTrueTypeFonts:=False, _  
SaveNativePictureFormat:=False, SaveFormsData:=False, SaveAsAOCELetter:= _  
False
```

```
End Sub
```





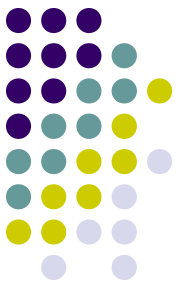
Effet de la macro:

- A chaque fois qu'elle est exécutée, les deux commandes enregistrées le sont également (le document est donc enregistré deux fois)
- Les instructions correspondent à la description en VBA des commandes du logiciel (ici Word).
- Lorsque la macro doit être exécutée, elle est compilée (traduite en langage machine), chargée en mémoire et la machine l'exécute.



Éléments de base

- **Tout programme est une suite d'instructions, avec une syntaxe précise et dépourvue de toute ambiguïté.**
- **1 ligne = 1 instruction**
 - **Si une instruction ne tient pas sur une ligne, il faut mettre en fin de ligne un blanc souligné (underscore) et passer à la ligne suivante.**
- **Instructions de déclaration**
 - **Il s'agit d'indications pour le compilateur, lui permettant de traduire les instructions de calcul et leur enchaînement en langage machine.**
 - **Exemple: sub macro1() (indique le début d'un programme)**
- **Instructions de calcul**
 - **Il s'agit d'instructions demandant l'exécution d'un calcul ou d'un programme déclaré au préalable (par une instruction de déclaration) et modifiant l'état de la mémoire.**
 - **Les instructions sont composées à partir de mots du langage (dits mots réservés), de noms définis par le programmeur à l'aide des instructions de déclaration et des noms définis par le concepteur du logiciel associés aux commandes du logiciel.**



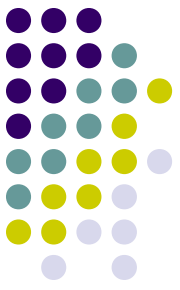
Rôles:

- Dans la suite, on distinguera:
 - Le programmeur : celui qui écrit les macros à l'aide du langage VBA et de ses extensions/logiciel utilisé.
 - L'utilisateur: celui qui, à partir d'un document exécute les macros et inter-agit avec elles.
- Dans le cadre des TDs, vous avez les deux rôles à la fois, mais en entreprise, les rôles sont souvent différenciés.

Concepts de base pour démarrer

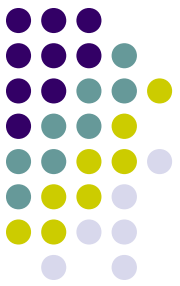


- Variables (définition, utilisation)
- Affectation
- Entrées- sorties



Variables

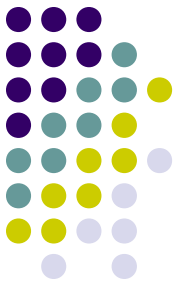
- Le programmeur utilise sciemment la mémoire au travers de la notion de variable:
- Une variable est un mot mémoire auquel le programmeur donne un **nom** et un **type**.
- Le **nom** sert ensuite à désigner cet emplacement et/ou la valeur qu'il contient
- Le **type** sert à indiquer à la machine quel codage employer pour les valeurs qui seront rangées dans la variable.



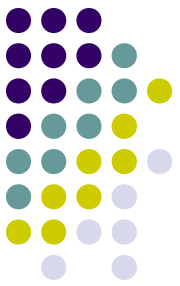
Exemple

- **Déclaration** d'une variable
 - Dim x As Integer
 - Dans l'emplacement appelé x, on pourra ranger un entier.
- Utilisation d'une variable pour ranger une valeur (c'est l'**affectation**)
 - $x=3$
 - Ranger la valeur 3 dans la case mémoire x.

Déclaration d'une variable locale

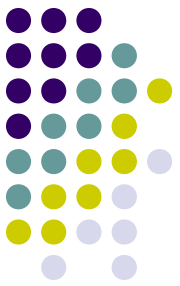


- Syntaxe:
 - Dim nomvar As typevar
- Effet de la déclaration:
 - Réservation d'un espace mémoire de taille correspondant au type typevar.
 - Cet espace n'est utilisable que dans la macro dans laquelle a été faite la déclaration.



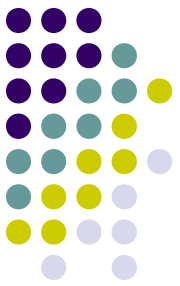
Affectation

- **Syntaxe:**
 - Nomvar = expression
 - Nomvar est un nom de variable déclarée
 - Expression décrit un calcul ou appelle une fonction à partir d'opérandes et d'arguments constants ou variables.
- **Effet:**
 - L'ordinateur calcule la valeur de l'expression, puis range cette valeur dans l'espace mémoire associé à la variable.



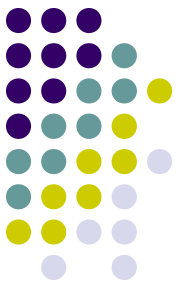
Exemple

- **Sub** essai()
- **Dim** a **As** Integer
- **Dim** b **As** Integer
- $a=15$ met la valeur entière 15 dans case a.
- $b=a+2$ met la valeur entière 17 dans case b.
- $b=b*2 +a+10$ met la valeur entière 59 dans la case b
- **End Sub**



Types de base

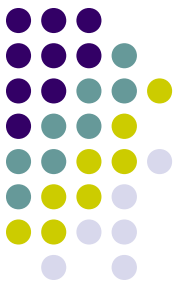
- **Byte** : nombre entier de 0 à 255
- **Integer** : nombre entier de -32768 à 32767
- **Boolean** : True ou False
- **String** : chaîne de caractères
- **Double** : nombre décimal en double précision
- **Variant** : lorsqu'il peut y avoir plusieurs types
- Il y a d'autres types qui ne seront pas utilisés dans le cadre du cours.



Sortie d'information: MsgBox

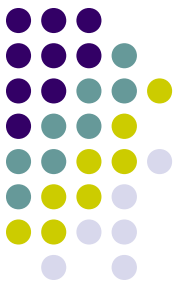
- MsgBox permet à une macro d'afficher, à l'utilisateur une information sous forme d'une boîte de dialogue.
- Syntaxe:
 - MsgBox(« message à afficher »)
- Effet:





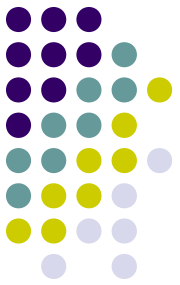
Exemple

- **Sub** essai()
- **Dim** a **As** Integer
- **Dim** b **As** Integer
- $a=15$ met la valeur entière 15 dans case a.
- $b=a+2$ met la valeur entière 17 dans case b.
- $b=b*2 +a +10$ met la valeur entière 59 dans la case b
- **MsgBox(b)** affiche la valeur de la variable b
- **End Sub**



Entrée d'information

- La fonction `InputDialog` permet à l'utilisateur de fournir des données à une macro.
- Syntaxe:
 - `nomvar=inputBox(« message »)`
- Effet:
 - Une boîte de dialogue est affichée avec le message dans la zone grise.
 - L'utilisateur entre une donnée dans la zone blanche
 - La valeur rentrée par l'utilisateur est le résultat de la fonction (on dit que la fonction **retourne** cette valeur)
 - Cette valeur est rangée dans la variable (affectation).



Exemple

- Sub double()
- Dim x As Integer
- x=InputBox(" entrez un entier ")
- $x=2*x$ calcule le double de la valeur entrée
- MsgBox(x) affiche cette valeur
- End Sub