

WORD 2000 FR

**Automatisation
(champs et macros)**

1 Table des matières

1	Table des matières.....	2
2	Introduction.....	4
3	Champs - référence	5
3.1	Qu'est-ce qu'un champ ?	5
3.1.1	Les champs résultats	5
3.1.2	Les champs signets	5
3.1.3	Les champs d'action	5
3.2	Des champs sans le savoir	5
3.3	Affichage des champs	5
3.4	Insertion d'un champ	6
3.5	Catégories de champs	6
3.5.1	Automatisation	6
3.5.2	Liaisons et renvois	6
3.5.3	Date et heure.....	6
3.5.4	Équations et formules.....	6
3.5.5	Numérotation	7
3.5.6	Tables et index	7
3.5.7	Résumé.....	7
3.5.8	Publipostage	7
3.5.9	Utilisateur.....	7
3.6	Boîte de dialogue « Insérer un champ »	7
3.7	Les arguments.....	7
3.8	Les commutateurs.....	8
3.9	Commutateurs, exemples	8
3.10	Les commutateurs généraux	8
3.10.1	Verrouillage (\!).....	8
3.10.2	Date-heure (\@).....	8
3.10.3	Numérique (\#)	8
3.10.4	Mise en forme (*)	8
3.10.5	Numériques \#	9
3.10.6	Les dates \@	10
3.10.7	Les heures \@	10
3.10.8	Ajouter un commutateur	11
3.11	Insertion d'un champ au clavier	12
3.12	Actualisation des champs.....	12
3.13	Verrouillage des champs.....	12
3.14	Rompre les liaisons d'un champ	13
3.15	Raccourcis clavier : récapitulatif.....	13
3.16	Les champs formule.....	13
3.16.1	Les opérateurs mathématiques et relationnels	13
3.16.2	Les fonctions	14

3.17	Champs formules dans un tableau.....	14
3.18	Exemples de champs	16
4	Les macros	18
4.1	Enregistrer une macro.....	18
4.1.1	Démarrer l'enregistrement	18
4.1.2	Enregistrement en cours	20
4.1.3	Arrêter l'enregistrement.....	21
4.2	Consulter le code de la macro.....	21
4.3	Gérer ses barres d'outils.....	22
4.3.1	Supprimer ou déplacer des outils.....	22
4.3.2	Rétablir une barre d'outils	22
4.3.3	Ajouter une nouvelle commande	23
4.3.4	Personnaliser un outil	24
5	Notions élémentaires de VBA.....	27
5.1	Objet.....	27
5.2	Propriété.....	27
5.3	Méthode	27
5.4	Collection.....	28
5.5	Variable	28
5.6	Procédure	29
5.7	Fonction	29
5.8	Structures de contrôle.....	29
5.8.1	Les conditions	30
5.8.2	Les boucles	30
5.8.3	Les sélections.....	30
5.8.4	Les sauts.....	31
5.8.5	La gestion des erreurs	31
5.8.6	Les blocs d'instructions par objets	31
5.9	Événement.....	31

2 Introduction

L'automatisation dans Word va vous aider à éviter des cas de répétitions fastidieuses ou va vous aider à la complétion¹ d'un document.

Ce module n'est pas un cours complet car il faudrait des jours et des jours pour faire le tour de tout ce qui est possible.

Il y aura deux parties à ces notes : une première vous apprendra à utiliser des champs, la seconde à enregistrer, corriger et mettre en œuvre une macro.

Vous trouverez sur Internet une foule de renseignements.

Prenez comme point de départ le site <http://faqword.free.fr/index.php?lng=fr>

¹ Action de compléter; résultat de cette action, achèvement, accomplissement. (tlfi)

3 Champs - référence

3.1 Qu'est-ce qu'un champ ?

Un champ permet d'insérer automatiquement des informations relatives au document (telles que son auteur ou le nom de fichier), d'effectuer des calculs, de créer des liaisons et des références à d'autres documents ou éléments, ainsi que d'exécuter d'autres tâches spéciales.

3.1.1 Les champs résultats

Ces champs spécifient des instructions que Word peut utiliser pour déterminer le texte à insérer dans le document.

- Les informations qu'ils génèrent sont appelées les résultats de champ.
- Le document stocke les instructions de champ, et non pas les informations réelles : Word pourra actualiser les résultats du champ si nécessaire.

3.1.2 Les champs signets

Certains champs marquent simplement le texte pour que l'utilisateur (ou un autre champ inséré dans le document) le retrouve plus tard.

3.1.3 Les champs d'action

- Les champs d'action indiquent à Word de réaliser une action spécifique, qui ne place pas de nouveau texte visible dans le document
- Par exemple, le champ **{MACROBUTTON}** place un bouton dans le texte : lorsqu'on cliquera dessus, Word exécutera une macro.

3.2 Des champs sans le savoir

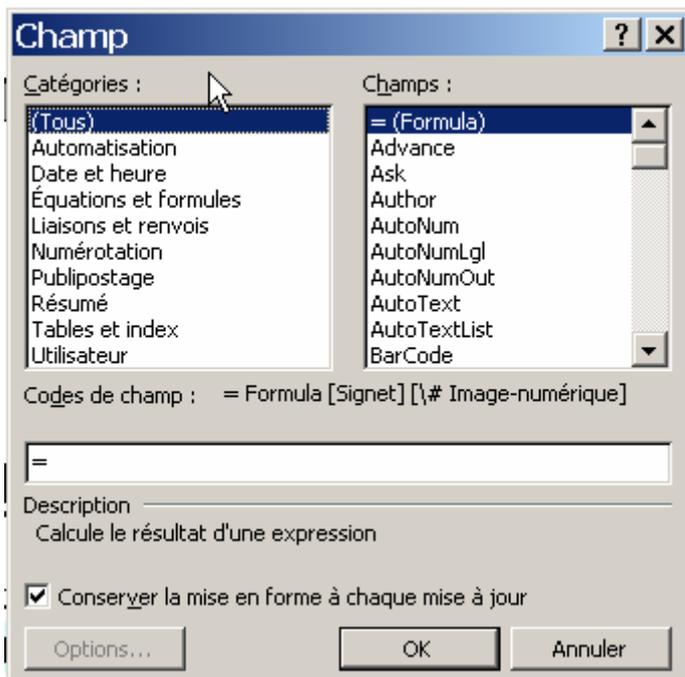
- Lorsque l'on insère la date, l'heure, ou le numéro de page dans le pied de page, Word place en fait un code de champ et non pas du texte.
- Un certain nombre de champs peut en effet être inséré directement à l'aide des boutons de la barre d'outils, ou grâce à des raccourcis clavier.

3.3 Affichage des champs

- Par défaut, les codes des champs ne sont pas visibles dans le document, mais bien les informations créées par les champs.
- Pour afficher ponctuellement un ou plusieurs codes de champ : sélection du champ et **SHIFT + F9**, ou clic droit, « **BASCULER LES CODES CHAMPS** »
- Pour travailler en visualisation de codes champ : Menu Outils, Options, Affichage, cochez « codes de champ ».
- Les codes de champ s'affichent entre accolades (**{ }**).

3.4 Insertion d'un champ

- A l'aide d'une commande existante (INSERTION DATE, TABLE DES MATIÈRES, SIGNET, ETC.)
- À l'aide de la boîte de dialogue « INSERTION, CHAMP »
- « Manuellement », à l'aide du raccourci clavier CTRL+F9



Écran 1, Dialogue "Champ"

3.5 Catégories de champs

3.5.1 Liaisons et renvois

Insère des informations trouvées ailleurs dans le document, à partir d'insertions auto ou encore à partir d'autres fichiers

3.5.2 Date et heure

Champs qui incluent l'heure et la date auxquelles un événement a eu lieu (par ex. le dernier enregistrement ou la dernière impression)

3.5.3 Automatisation

Crée et calcule les résultats des formules, insère des symboles

3.5.4 Équations et formules

Champs de comparaison et de logique - Champs réalisant une action, par ex. exécuter une macro, lancer une impression...)

3.5.5 Numérotation

Numérote les paragraphes, les pages et/ou les sections du document

3.5.6 Tables et index

Crée des entrées d'index, construit des index et des tables des matières, des figures et des illustrations.

3.5.7 Résumé

Insère ou stocke les informations sur le document

3.5.8 Publipostage

Insère toutes les informations spécifiques au publipostage

3.5.9 Utilisateur

Stocke les données de l'utilisateur (nom, initiales, adresse...) et les insère dans le document

3.6 Boîte de dialogue « Insérer un champ »

- Word indique la syntaxe à utiliser
 - Quelquefois le nom du champ suffit à lui seul
 - Souvent, les champs nécessitent des instructions supplémentaires :
 - Les arguments
 - Les signets
 - Les expressions
 - Les commutateurs

3.7 Les arguments

Les arguments fournissent à Word des informations nécessaires sur le champ, p.ex. : **{ INCLUDEPICTURE " Nom du fichier " }**

Il suffit de suivre la syntaxe fournie par Word

NB. L'emploi des guillemets est indispensable pour les noms comportant des espaces

Quelquefois, l'argument doit se compléter par un nom de signet ou une expression

- Un nom signet permet au champ de n'agir que sur un bloc de texte marqué par signet
ex. : **{ PAGEREF mon_signet }** : insère le numéro de la page renfermant le signet spécifié

- Une expression permet de créer des champs calcul

3.8 Les commutateurs

- Les commutateurs modifient le comportement d'un champ. Ils commencent tous par \
ex. **{ INCLUDEPICTURE " Nom du fichier " \d }**
 - Les commutateurs généraux modifient le comportement d'un champ, tel que la casse ou le format numérique et peuvent s'appliquer à la plupart des champs.
 - Les commutateurs spéciaux sont propres à un champ.

3.9 Commutateurs, exemples

- Commutateurs généraux
{ INCLUDETEXT « mon fichier.doc" * Upper }
Le commutateur * Upper met le texte du fichier inséré en lettres majuscules. *Commutateurs spéciaux
- **{ REF mon_signet \h }**
Le commutateur \h insère un lien hypertexte sur le renvoi.

3.10 Les commutateurs généraux

3.10.1 Verrouillage (!)

Empêche la mise à jour des champs faisant partie du texte inséré à l'aide d'un champ SIGNET, INCLUDETEXT ou REF.

3.10.2 Date-heure (\@)

Définit la mise en forme des champs dont la valeur est une date ou une heure.

3.10.3 Numérique (\#)

Spécifie l'affichage d'une valeur numérique, par exemple le nombre de chiffres après la virgule et l'utilisation des symboles monétaires.

3.10.4 Mise en forme (*)

Spécifie le format des valeurs numériques, la casse et la mise en forme de caractères ; empêche les changements de mise en forme des valeurs de champ lorsqu'un champ est mis à jour.

Casse

		{ FILENAME *...}
* Lower	Tout en minuscules	word av 09 (automatisation) - notes fr.doc

* Caps	La première lettre de chaque mot en majuscules	Word Av 09 (Automatisation) - Notes FR.Doc
* FirstCap	La première lettre de chaque phrase en majuscules	Word Av 09 (automatisation) - notes FR.doc
* Upper	Tout en majuscules	WORD AV 09 (AUTOMATISATION) - NOTES FR.DOC

Formats numériques

		{ =17 * ...]
* Hex	Nombres hexadécimaux	11
* ALPHABETIC ou * \ alphabetic	Format alphabétique majuscules ou minuscules	Q q
* Arabic	Format chiffres arabes	17
* Cardtext	Texte cardinal	dix-sept
* Dollartext	Texte cardinal. Word insère « et » à la place de la virgule décimale et affiche les deux premières décimales (arrondies au centième) sous forme de chiffres arabes.	dix-sept francs
* Ordinal	Texte ordinal arabe.	17e
* Ordtext	Texte ordinal	dix-septième
* ROMAN ou * roman	Format chiffres romains majuscules ou minuscules	XVII xvii

Formats de caractères

* CHARFORMAT	La mise en forme de la première lettre du type de champ à l'ensemble de la valeur.
* MERGEFORMAT	La mise en forme de la valeur précédente à la nouvelle valeur. Attention garde la même mise en forme sur le même nb de mots. Nb : ce commutateur est inséré par défaut via le menu Insérer Champ (option "conserver la mise en forme à chaque mise à jour")

3.10.5 Numériques \#

, ou .	Spécifie le séparateur décimal
\# "positif;négatif;zéro"	Indique les formats positif, négatif et zéro. Chaque format doit être séparé par un ;
` (accent grave)	Affiche le chiffre de l'élément précédent numéroté à l'aide de la commande Légende ou à l'aide d'un

	champ SEQ. L'identificateur de l'élément tel que « tableau » ou « image » doit figurer entre accents graves (`).
\# x	Tronque les chiffres à gauche s'il est placé à gauche de la décimale, Arrondit le nombre s'il est placé à droite de la décimale.
\# #	Spécifie la façon d'afficher les chiffres dans la valeur. Si à cet endroit la valeur ne comporte pas de chiffre, Word affiche un espace. ex. : \# "# ###,00"
\# 0	Si la valeur ne comprend pas de chiffre à cet endroit, Word affiche un 0.

NB : Si le commutateur comporte des espaces, l'insérer entre " "

3.10.6 Les dates \@

NB : Si le commutateur comporte des espaces, l'insérer entre " "

Affichage du jour

d	Un chiffre sans 0 s'il s'agit des neuf premiers jours du mois.
dd	Un chiffre précédé d'un 0 s'il s'agit des neuf premiers jours du mois.
ddd	Une abréviation de trois lettres
dddd	Le jours en toutes lettres

Affichage du mois :

M	Un chiffre sans 0 s'il s'agit des neuf premiers mois de l'année.
MM	Un chiffre précédé d'un 0 s'il s'agit des neuf premiers mois de l'année
MMM	Une abréviation de trois lettres
MMMM	Le mois en toutes lettres

Affichage de l'année

Yy	Deux chiffres précédés d'un 0 pour les années 01 à 09
Yyyy	Les quatre chiffres

3.10.7 Les heures \@

NB : Si le commutateur comporte des espaces, l'insérer entre " "

Affichage des secondes :

s	Sans 0 pour les valeurs comprises entre 0 et 9
ss	Précédée d'un 0 pour les valeurs comprises entre 0 et 9

Affichage des minutes :

m	Sans 0 pour les valeurs comprises entre 0 et 9
mm	Précédée d'un 0 pour les valeurs comprises entre 0 et 9

Affichage de l'heure

h en minuscule affiche l'heure au format 12 heures.

H en majuscule affiche l'heure au format 24 heures.

h ou H	Sans 0 (zéro) pour les heures comprises entre 0 et 9.
hh ou HH	Précédée d'un 0 pour les heures comprises entre 0 et 9.

Affichage AM et PM dans l'un des quatre formats suivants. Pour utiliser sous Windows les symboles AM et PM, il faut d'abord modifier les Paramètres régionaux du Panneau de configuration Windows.

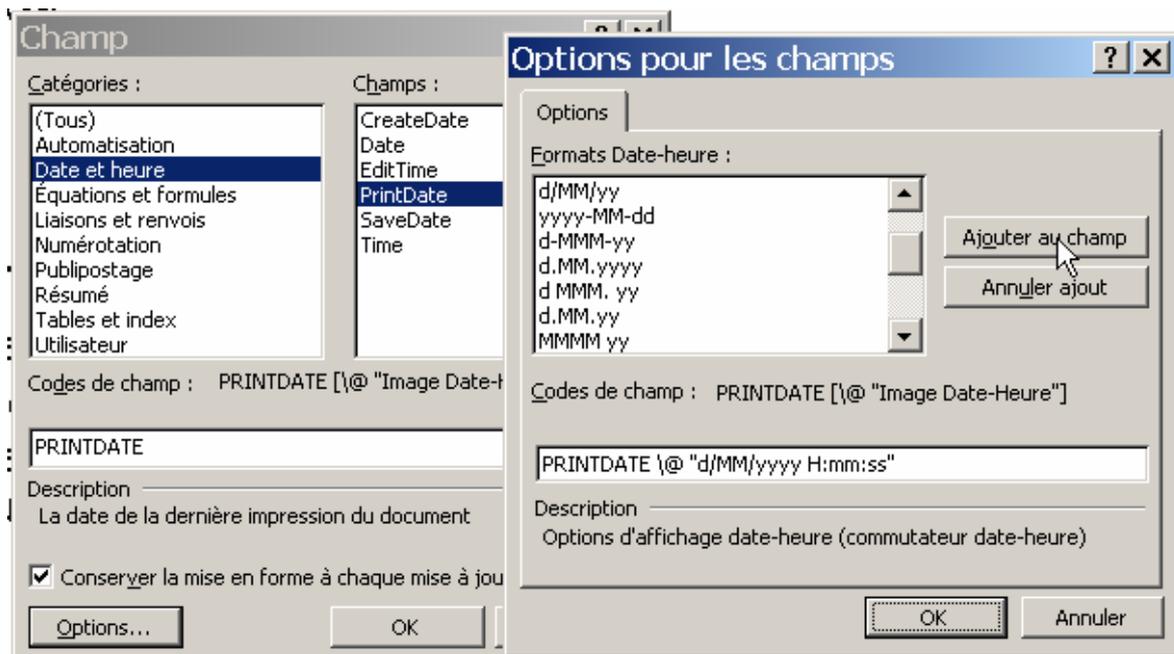
- am/pm
- AM/PM
- a/p
- A/P

3.10.8 Ajouter un commutateur

À l'aide de la boîte **INSERTION**, **CHAMP**

Cliquer sur le bouton **OPTION**, sélectionner éventuellement l'onglet, et choisir les commutateurs dans la liste déroulante.

On peut ajouter autant de commutateurs que nécessaire.



Écran 2, Dialogues Insertion - Champ – Options

Résultat : 25/12/2008 17:52:00

3.11 Insertion d'un champ au clavier

- Pour insérer un champ sans passer par le menu Insertion, Champ :
 - Ctrl + F9 permet d'entrer les accolades { }
 - Attention : ces accolades ne peuvent être saisies au clavier
 - Saisir le champ entre les accolades
- Cette méthode est conseillée uniquement pour les utilisateurs ayant une bonne connaissance des champs et dans certains cas où il n'est pas possible d'utiliser le menu Insertion (dans une étiquette de publipostage par exemple).

3.12 Actualisation des champs

- Les champs ne se mettent pas à jour automatiquement
- Ils peuvent s'actualiser :
 - À l'impression ou aperçu avant impression si l'option est cochée dans le menu Outils, Options, Impression
 - À la demande : sélectionner le champ ou tout le document, puis F9, ou le menu contextuel (clic droit) « METTRE À JOUR LES CHAMPS »

3.13 Verrouillage des champs

- Pour empêcher l'actualisation temporaire d'un champ, on peut le verrouiller :
 - Placer le point d'insertion sur le champ
 - Appuyer sur CTRL + F11

- Pour déverrouiller un champ :
 - Placer le point d'insertion sur le champ
 - Appuyer sur **SHIFT + CTRL + F11**

3.14 Rompre les liaisons d'un champ

Pour ne plus jamais actualiser le champ, on peut remplacer les codes de champ par son résultat le plus récent : la liaison sera rompue. Pour rompre une liaison :

CTRL + SHIFT + F9

Nb : Cette commande ne touche pas les champs signets.

3.15 Raccourcis clavier : récapitulatif

F9	Actualiser un champ
SHIFT + F9	Afficher/Masquer un code de champ
SHIFT + F11	Atteindre le champ précédent
F11	Atteindre le champ suivant
CTRL + SHIFT + F11	Déverrouiller un champ
CTRL + F9	Insérer un champ
CTRL + SHIFT + F9	Rompre la liaison d'un champ
CTRL + F11	Verrouiller un champ

3.16 Les champs formule

Les champs formule calculent à l'aide d'une formule mathématique. Ces champs débutent toujours par le signe =

Ces champs utilisent :

- Les opérateurs mathématiques et relationnels
- Les fonctions
 - Les fonctions utilisent des arguments, avec une syntaxe similaire à celle d'Excel : les arguments sont entre parenthèses, et séparés par des ; ou :
 - Les arguments peuvent être des nombres, des formules ou des noms de signet

3.16.1 Les opérateurs mathématiques et relationnels

- + Addition
- - Soustraction
- * Multiplication
- / Division
- % Pourcentage
- ^ Puissances et racines
- = Égal à
- < Inférieur à
- <= Inférieur ou égal à
- > Supérieur à
- >= Supérieur ou égal à
- <> Différent de

3.16.2 Les fonctions

ABS(x)	Valeur positive d'un nombre ou d'une formule
AND(x;y)	Valeur 1 si les expressions logiques x et y sont toutes deux vraies ou valeur 0 (zéro) si l'une ou l'autre expression est fausse.
AVERAGE()	Moyenne d'une liste de valeurs.
DEFINED(x)	Valeur 1 (vraie) si l'expression x est valide ou valeur 0 (faux) si l'expression ne peut pas être calculée.
FALSE	0 (zéro).
INT(x)	Nombres situés à gauche du séparateur décimal dans la valeur ou la formule x.
MAX()	Plus grande valeur dans une liste.
MIN()	Plus petite valeur dans une liste
MOD(x;y)	Reste de la division de la valeur x par la valeur y un nombre entier de fois.
NB()	Nombre d'éléments dans une liste.
NOT(x)	Valeur 0 (zéro) (faux) si l'expression logique x est vraie ou valeur 1 (vrai) si l'expression est fausse.
OR(x;y)	Valeur 1 (vrai) si l'une et/ou l'autre expression logique x et y sont vraies ou valeur 0 (zéro) (faux) si les deux expressions sont fausses.
PRODUCT()	Résultat de la multiplication d'une liste de valeurs. Par exemple, la fonction { = PRODUCT (1,3,7,9) } renvoie la valeur 189.
ROUNDI(x;y)	Valeur de x arrondi au nombre de décimales y spécifié ; x peut être un nombre ou le résultat d'une formule.
SIGNE(x)	Valeur 1 si x est une valeur positive ou valeur -1 si x est une valeur négative.
SUM()	Somme d'une liste de valeurs ou de formules.
TRUE	1

3.17 Champs formules dans un tableau

- Se positionner sur la cellule où l'on souhaite afficher le résultat.
- Menu **TABLEAU, FORMULE**.
- Dans la zone **INSÉRER LA FONCTION**, cliquer sur une fonction. Pour additionner des nombres, par exemple, cliquez sur SUM.
- Pour faire référence au contenu d'une cellule de tableau, taper les références de cellule entre parenthèses dans la formule.
- Dans la zone **FORMAT**, sélectionner ou taper le format des nombres.
- Création de référence à des cellules de tableau
 - Pour faire référence aux cellules de tableau sous la forme A1, A2, B1, B2, etc., la lettre représente une colonne et le numéro représentant une ligne. Les références de cellule dans Word sont toujours des références absolues, contrairement à Excel. Par exemple, la référence à une cellule sous la forme A1 dans Word correspond à la référence \$A\$1 dans Excel.

Word 2000 – Automatisation (champs et macros)

- Utiliser le point-virgule (;) pour séparer les références à différentes cellules et le signe deux points (:) pour séparer la première cellule de la dernière cellule d'une plage déterminée.
- Pour faire référence à toute une ligne ou toute une colonne (permet au calcul d'inclure automatiquement toutes les cellules de la ligne ou de la colonne si l'on décide d'ajouter d'autres cellules ultérieurement) :
- On peut faire référence aux cellules du dessus, du dessous, de gauche, de droite : (ABOVE) (BELOW) (LEFT) (RIGHT), ex : **{=SUM(ABOVE)}**
- Hors tableau, On peut utiliser une plage qui comprend uniquement la lettre ou le numéro qu'elle représente, par ex, 1:1 pour faire référence à la première ligne d'un tableau.
- Hors tableau, on peut faire un calcul sur un tableau marqué d'un signet.

RÉGIONS	BELGES	ÉTRANGERS	TOTAL
Le Littoral	59,80%	10,00%	
Les villes d'art	3,20%	11,10%	
l'Ardenne	7,80%	1,80%	
les autres régions	4,80%	1,50%	
Le royaume			

¶

Formule

Formule :
=SUM(ABOVE)*100

Format :
0,00%

Insérer la fonction : Insérer le signet :

OK Annuler

Écran 3, Dialogue "Formule"

RÉGIONS	BELGES	ÉTRANGERS	TOTAL
Le Littoral	59,80%	10,00%	69,80%
Les villes d'art	3,20%	11,10%	14,30%
l'Ardenne	7,80%	1,80%	9,60%
les autres régions	4,80%	1,50%	6,30%
Le royaume	75,60%	24,40%	100,00%

Écran 4, tableau calculé

3.18 Exemples de champs

Quoi	Comment	Exemple
Utilisé en entête ou en pied de page : Insère le texte à partir du paragraphe de style Titre 1 en cours	{ STYLeref "Titre 1" }	Champs - référence
Insère la date de création du document au format : dimanche 8 octobre 06	{ CREATEDATE \@ "dddd d MMMM yy" }	jeudi 5 octobre 06
Insère l'heure actuelle au format : 10:15:22 pm	{ TIME \@ "HH:mm:ss am/pm" }	17:52:39
Insère le nombre de caractères du fichier	{ NUMCHARS }	29652
Insère le nom du fichier, ainsi que son chemin	{ FILENAME \p }	D:\Materiel Peda\Teatcher Chambre\Word\Word automatisation\word Av 09 (automatisation) - notes FR.doc
Valeur nulle (format de nombre)	{ = 125*0 \# "00,00;(00,00);valeur nulle" }	valeur nulle
Déplace de 6 points vers le bas le reste du texte de la ligne	{ ADVANCE \d 6 }	Début ligne fin ligne
Déplace de 30 points vers la gauche le reste du texte de la ligne (entraîne une superposition)	{ ADVANCE \I 30 }	Début ligne fin ligne
Crée une intégrale :	{ EQ \I(5;2;3) }	$\int_2^5 3$
Crée une fraction	{ EQ \F(7;15) }	$\frac{7}{15}$

Word 2000 - Automatisation (champs et macros)

Quoi	Comment	Exemple
Surimprime chaque élément sur les précédents : O et /	{ EQ \O(O;/) }	∅
cent trente-septième	{ = 137 * ordtext }	cent trente-septième
Calcule la différence entre le nombre de caractères du fichier et le nombre de caractères sans les espaces, pour obtenir le nombre d'espaces	{={DOCPROPERTY "CharactersWithSpaces" } - {DOCPROPERTY "Characters" }}	5484
Calcul	{= 100 * 15 \# "# ###,000" }	1 500,000

4 Les macros

Macro est le diminutif de macro-commande qui désigne à l'origine une commande regroupant plusieurs commandes. C'est un programme simple. Les macros sont très utiles pour exécuter de façon répétitive les mêmes suites d'opérations. Par la suite on a pris l'habitude d'appeler macro tout programme introduit dans un logiciel pour en étendre les commandes.

Word est un traitement de texte programmable. Les commandes et fonctions déclenchées au clavier ou à la souris par l'utilisateur peuvent pratiquement toutes être exécutées de façon automatisée par une macro.

Pour programmer Word on utilise le langage VBA (Visual Basic pour Applications) un langage très proche de Visual-Basic et commun à un grand nombre d'applications en particulier celles du pack Office comme Excel ou Access.

Mais il n'est pas nécessaire de connaître le langage (c'est cependant utile), car un enregistreur de macros va nous aider.

4.1 Enregistrer une macro

Supposons que nous devons régulièrement coller du texte, via le menu ÉDITION, COLLAGE SPÉCIAL, TEXTE SANS MISE EN FORME.

La manipulation est fastidieuse lorsqu'elle doit être répétée souvent et il n'y a pas de raccourci-clavier existant, ni d'outil dédié.

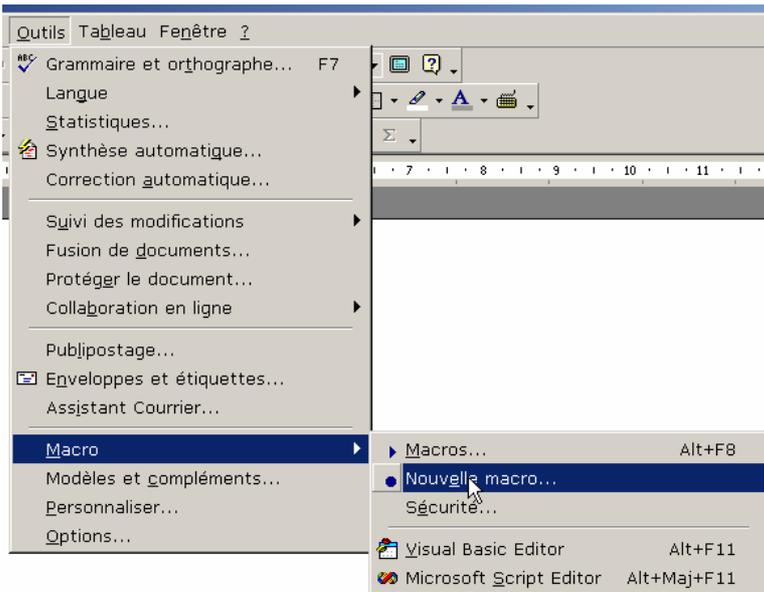
Nous voudrions en faire un raccourci-clavier et le rendre disponible dans tous les nouveaux documents.

Pour éviter tout problème lors de l'enregistrement, faites une répétition « à blanc » de la procédure

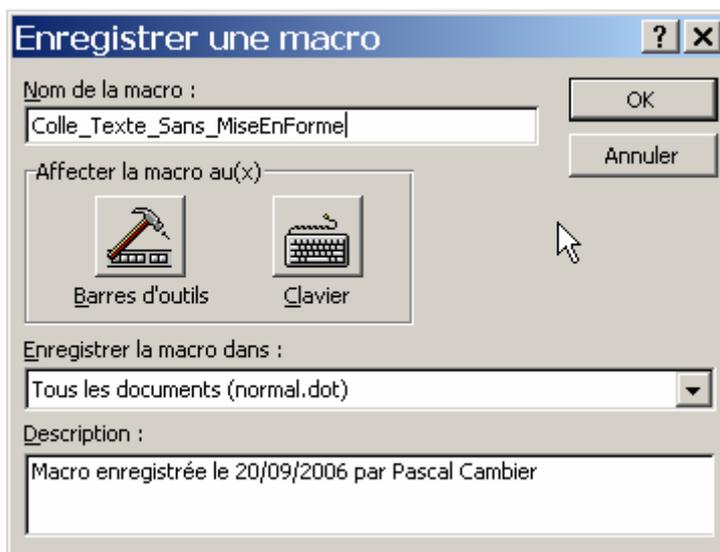
4.1.1 Démarrer l'enregistrement

- Faites-le via le menu OUTILS – MACRO – NOUVELLE MACRO

Word 2000 – Automatisation (champs et macros)

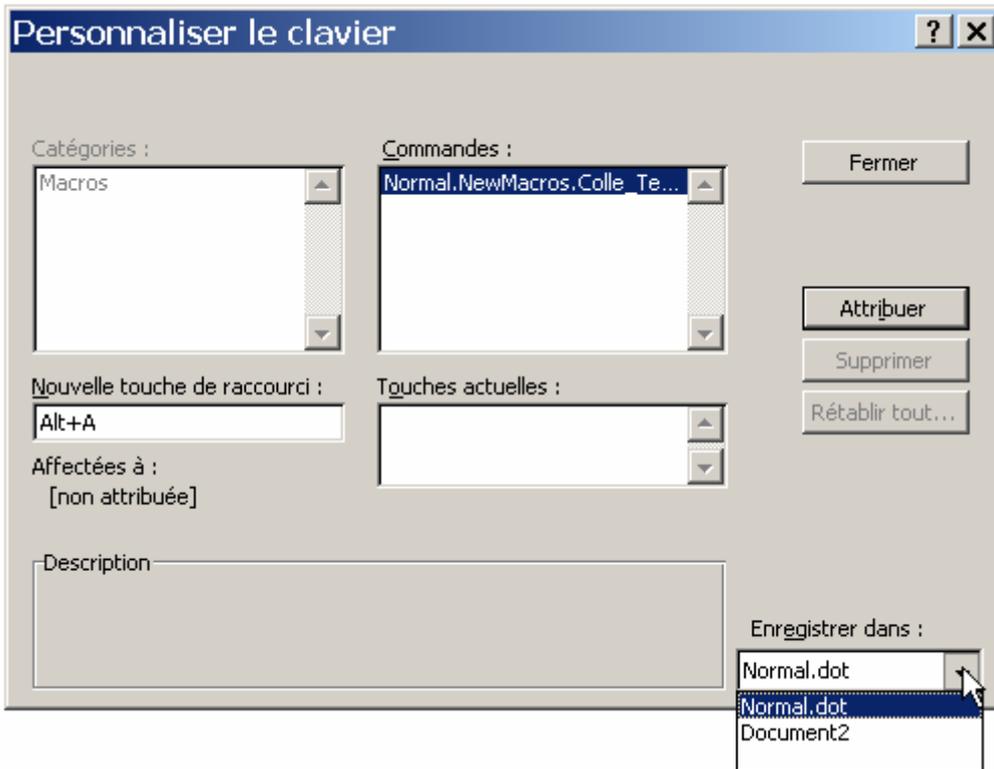


Écran 5, Menu outils - Macro - Nouvelle macro



Écran 6, Dialogue Enregistrer une macro

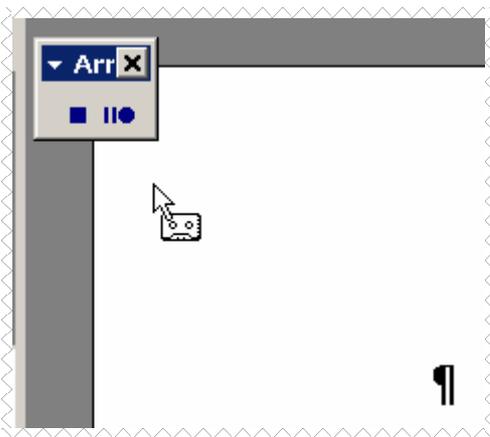
- Tapez un nom de macro explicite sans espace. Utilisez le caractère de soulignement (_) ou des majuscules pour le rendre lisible.
- Cliquez ensuite sur le bouton **CLAVIER** pour lui attribuer un raccourci



Écran 7, Dialogue "Personnaliser le clavier"

- Dans la zone NOUVELLE TOUCHE DE RACCOURCI, cherchez une combinaison (ou séquence) qui n'est pas attribuée ou que vous n'utiliserez jamais.
- Cliquez sur ATTRIBUER pour affecter le raccourci à la macro
- Vérifiez que ENREGISTRER DANS est bien NORMAL.DOT, mettant ainsi la macro disponible pour tout nouveau document.
- Cliquez sur FERMER pour démarrer l'enregistreur

4.1.2 Enregistrement en cours

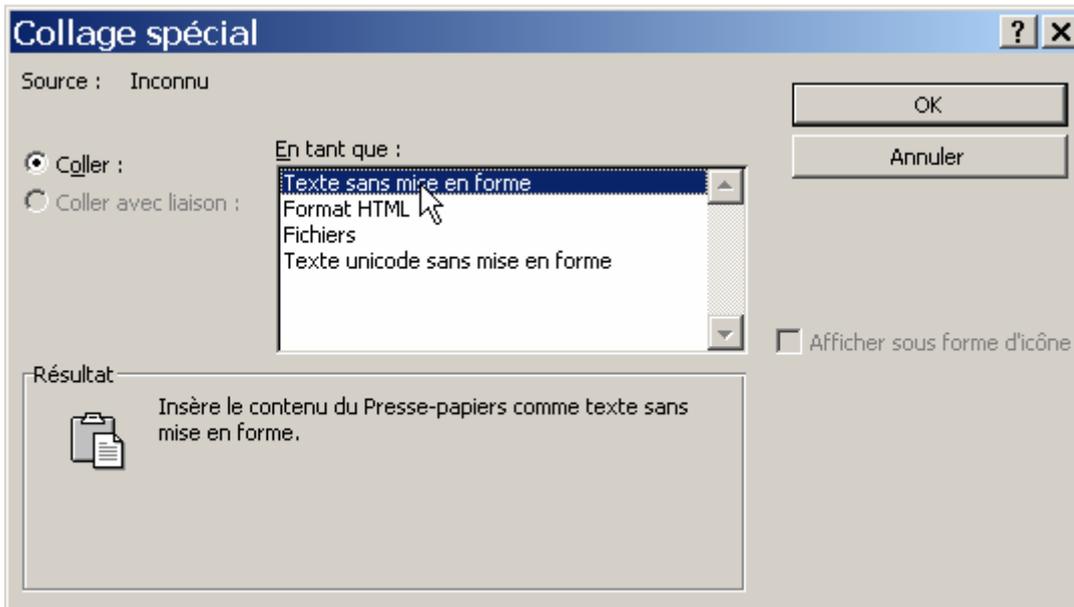


Écran 8, Enregistrement de macro en cours

- Dès ce moment là, l'enregistreur est en route et enregistre TOUT ce que vous faites.

Word 2000 – Automatisation (champs et macros)

- Le pointeur est accompagné d'une cassette audio, symbolisant l'enregistrement et une barre d'outil spéciale pour marquer une pause ou arrêter la macro.
- Faites toutes vos manipulations, donc ici le menu **ÉDITION**, **COLLAGE SPÉCIAL**, **TEXTE SANS MISE EN FORME**.



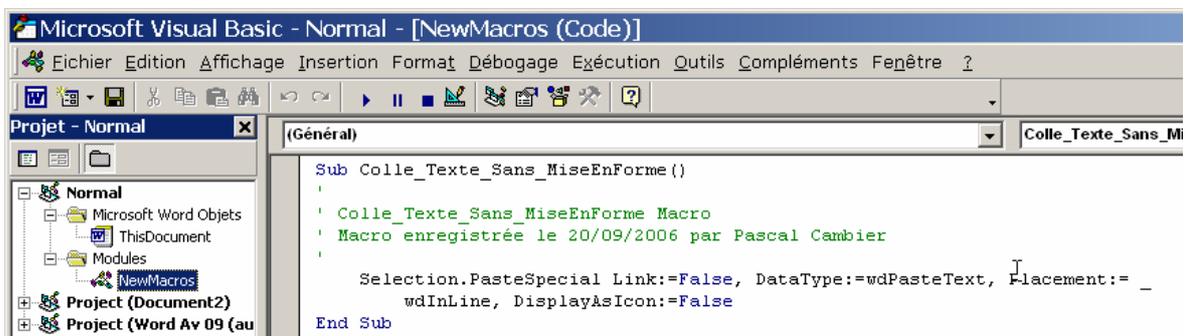
Écran 9, Dialogue "Collage spécial"

4.1.3 Arrêter l'enregistrement

En utilisant l'outil ad-hoc ou le menu **OUTILS – MACRO – ARRÊTER L'ENREGISTREMENT**

4.2 Consulter le code de la macro

Vous pouvez consulter (et corriger) le code de la macro à l'aide du raccourci ALT+F11 qui ouvre l'éditeur VBA



Écran 10, Éditeur VBA

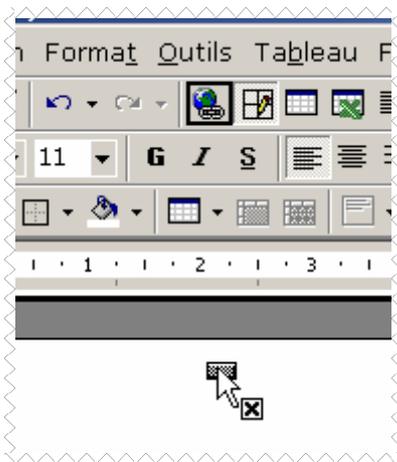
4.3 Gérer ses barres d'outils

Il est très facile de personnaliser une barre d'outils, en ajoutant ou supprimant des outils existants ou en créant de nouveaux outils.

4.3.1 Supprimer ou déplacer des outils

Il se peut fort bien qu'il y ait l'un ou l'autre outil que vous n'utilisez jamais, ou l'un ou l'autre qui ne soit pas à la place qu'il mérite.

Maintenez la touche ALT, prenez l'outil avec la souris et déplacez-le ailleurs ou hors barres pour le supprimer.

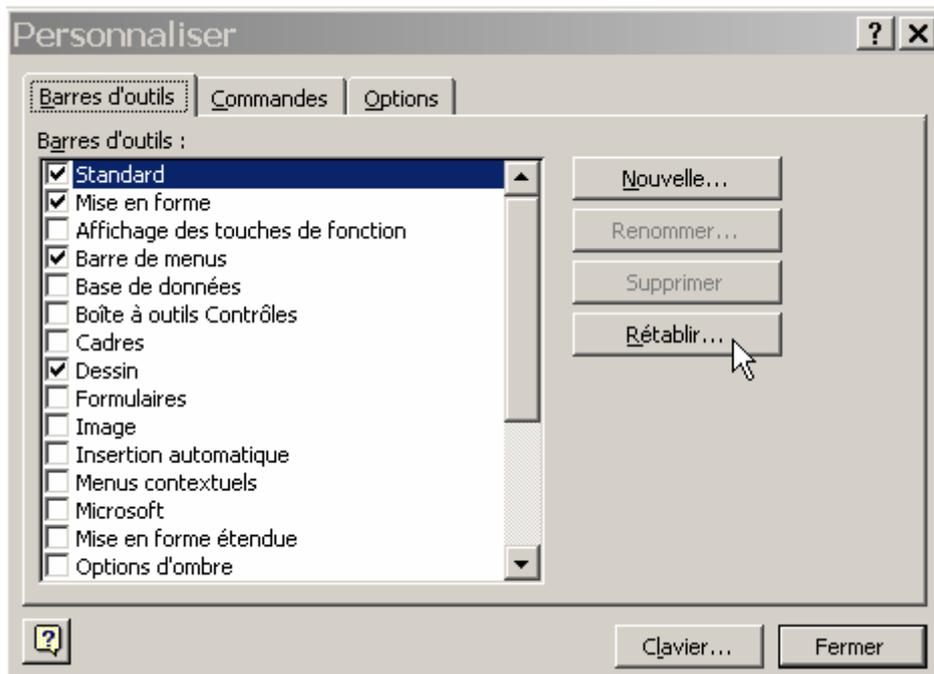


Écran 11, Outil en suppression

4.3.2 Rétablir une barre d'outils

Il est tellement facile de supprimer un outil de sa barre, qu'il faut bien vite apprendre à rétablir.

- Utilisez le menu **OUTILS – PERSONNALISER...** ou le menu **AFFICHAGE – BARRE D'OUTILS – PERSONNALISER...** ou encore le menu contextuel sur une barre d'outils et choisir **PERSONNALISER...**
- Sur l'onglet **BARRE D'OUTILS**, sélectionner la barre à corriger et cliquer sur le bouton **RÉTABLIR**. Valider l'avertissement et le tour est joué.



Écran 12, Dialogue "Personnaliser" - Barres d'outils

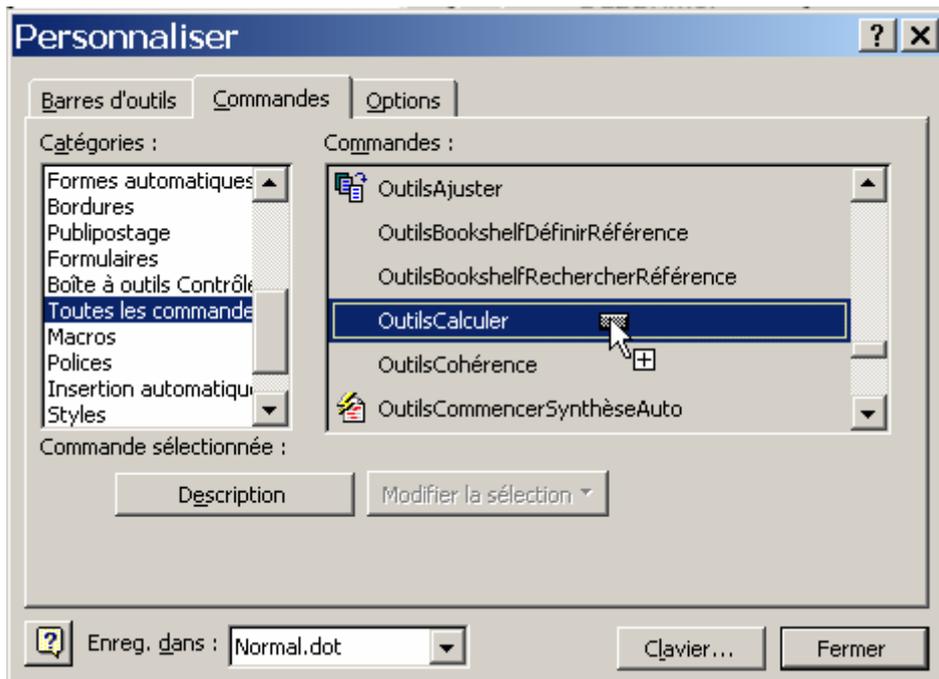
4.3.3 Ajouter une nouvelle commande

On va vous faire découvrir un outil sous-utilisé et pourtant bien pratique.

1. Ouvrez le dialogue précédent et choisissez l'onglet **COMMANDES**.
2. Dans la partie **CATÉGORIES**, choisissez **TOUTES LES COMMANDES**
3. Dans la partie **COMMANDES**, faites dérouler jusqu'à **OUTILSCALCULER**
4. À l'aide de la souris, prenez cette ligne (comme à l'Écran 13, Dialogue "Personnaliser" – Commandes ci-dessous) et glissez-là jusqu'à l'endroit désiré sur une barre d'outils.

À quoi peut donc servir cet outil ?

- *Écrivez dans votre texte une opération mathématique telle que « cela fera donc, TVA comprise $1325 * 1,186$ »*
- *Sélectionnez le calcul ;*
- *Cliquez sur l'outil **OUTILS CALCULER***
- *Observez tout de suite votre barre d'état (en bas à gauche) : vous avez le résultat du calcul.*
- *Et si vous choisissez **COLLER**, vous aurez le résultat : 1571,45*



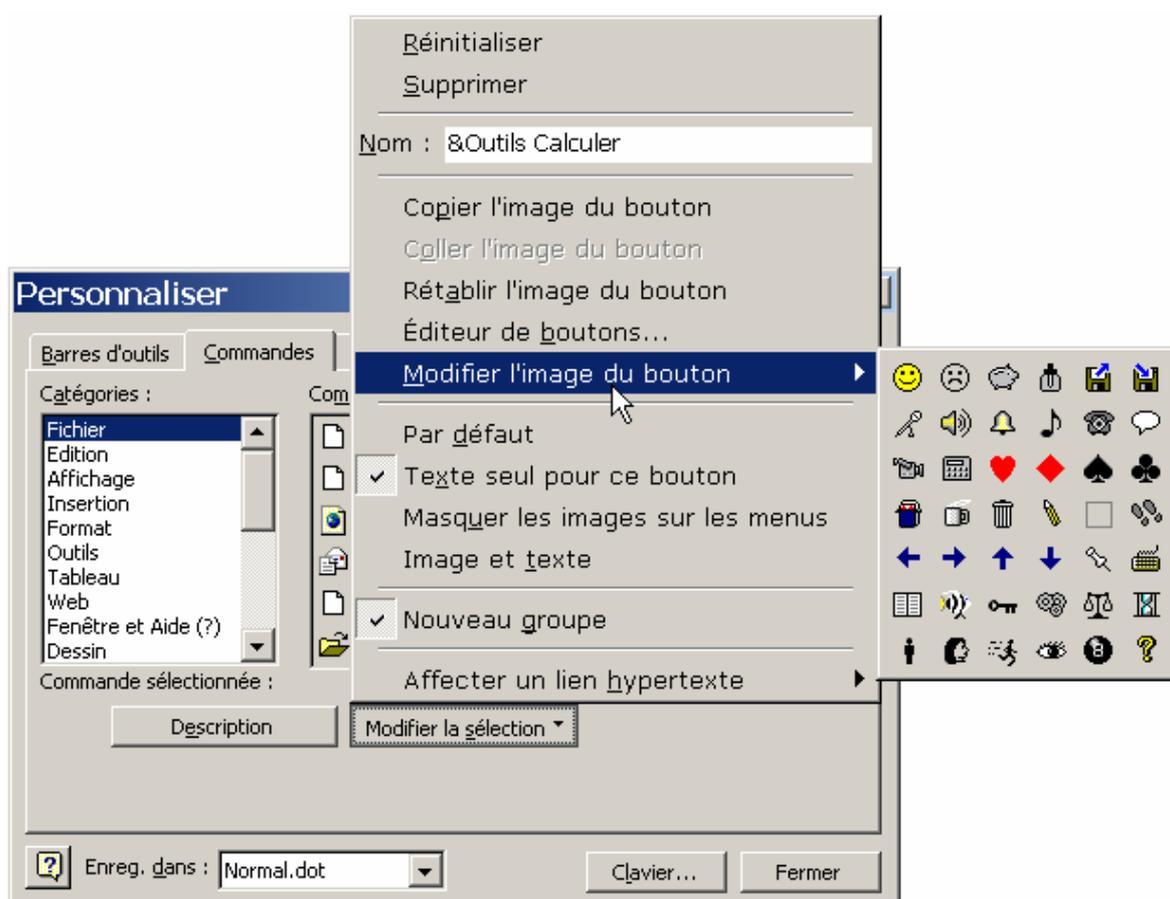
Écran 13, Dialogue "Personnaliser" – Commandes

4.3.4 Personnaliser un outil

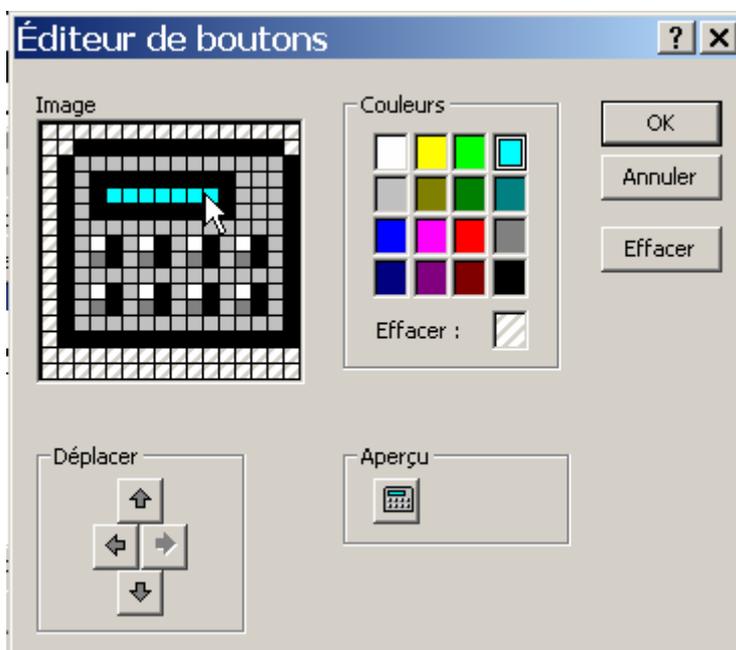
1. Ouvrez le dialogue Personnaliser – commandes,
2. sélectionnez dans la barre, l'outil que vous voulez personnaliser, par exemple ici l'outil Outils Calculer,
3. Dans le dialogue, cliquez sur le bouton modifier la sélection
4. Choisissez modifier l'image du bouton
5. Choisissez par exemple l'image de la calculatrice
6. Dans le dialogue, cliquez sur le bouton modifier la sélection
7. Choisissez Par défaut.

C'est fait.

- Vous pouvez aussi modifier l'image en passant par l'éditeur de bouton dans le même menu.



Écran 14, Dialogue "Personnaliser" - Modifier l'image du bouton



Écran 15, L'éditeur de boutons

4.3.5 Affecter une macro à un outil

La procédure est exactement identique à celle décrite au point 4.3.3 ci-dessus, sinon que :

2. Dans la partie **CATÉGORIES**, choisissez **MACROS**
3. Dans la partie **COMMANDES**, choisissez la macro que vous voulez affecter. Continuez ensuite comme au point 4.3.4 ci-dessus.

5 Notions élémentaires de VBA

VBA reprend à l'identique toutes les caractéristiques de Visual Basic, un langage simple *orienté objet*. La syntaxe est élémentaire, les contraintes limitées ce qui en fait un langage idéal pour de petites applications. Les premières notions à comprendre sont peu nombreuses :

5.1 Objet

Le langage manipule des objets : un document Word est un objet, un paragraphe est un objet, mais aussi un style, un dictionnaire, un bouton, un menu ou même l'application Word elle-même. Un objet se caractérise par sa Classe. L'objet **ActiveDocument** est le 'document actif', celui que voit l'utilisateur, c'est un objet de la classe Document. Dès qu'un objet existe – on parle d'instance – il hérite de toutes les caractéristiques de sa classe que sont principalement ses propriétés et ses méthodes.

5.2 Propriété

Une propriété est une donnée interne propre à un objet. Ainsi tous les objets de la classe Document possèdent un nom **Name**, des paragraphes **Paragraphs**, des mots **Words**, des caractères **Characters** etc. Une propriété peut être elle-même un objet ou une collection (voir 5.4 ci-dessous) d'objets. On accède à la valeur d'une propriété d'un objet particulier en écrivant **Objet.propriété** avec un point comme délimiteur : **ActiveDocument.Name** est le nom du document actif. Pour lire ou modifier une propriété il faut décliner toute la suite des objets et propriétés qui sont ses parents.

l'instruction suivante :

ActiveDocument.Paragraphs(1).Range.Font.Color = wdColorBlue met en bleu tout le texte du premier paragraphe du document actif.

5.3 Méthode

Une méthode est une commande propre à un objet. Tous les objets de la classe Document peuvent recevoir l'ordre enregistrer **Save**, imprimer **PrintOut**, vérifier l'orthographe **CheckSpelling** etc. Comme une propriété, une méthode s'écrit avec son objet et un point : **ActiveDocument.Save** enregistre le document actif. Une méthode comporte souvent une série d'arguments qui précisent les modalités de la commande. Certains arguments peuvent être facultatifs. Les arguments sont passés à la méthode soit entre parenthèses soit nommément avec l'opérateur d'affectation := .

Les instructions

- **ActiveDocument.SaveAs ("Essai.doc")** et
- **ActiveDocument.SaveAs FileName:="Essai.doc"**

commandant d'enregistrer le document actif sous le nom Essai.doc sont équivalentes.

5.4 Collection

Une collection est un objet qui regroupe des objets d'une même classe. L'objet Documents désigne la collection des documents ouverts dans l'application. Il ne faut pas confondre collection et classe ni collection et objets: les propriétés et les méthodes de Documents ne sont pas celles d'un document. Exemple : **Documents.Count** est la propriété du nombre de documents ouverts dans la collection Documents, et **Documents.Add** est la méthode pour ouvrir un nouveau document. Les objets d'une collection sont accessibles quant à eux comme propriétés particulières de la collection. Exemple : s'il n'y a qu'un document ouvert nommé Essai.doc **Documents.Item(1)** ou **Documents(1)** ou Documents("Essai.doc") désignent de façon équivalente le document Essai.doc. Dans le cas des collections les objets éléments sont référencés par indice ou nommément entre parenthèses.

5.5 Variable

Une variable est un élément de programmation désignant et gardant une donnée en mémoire. Une variable possède un nom, une portée et une valeur. Le nom est un identificateur unique. La portée précise dans quelle partie du code ce nom est reconnu et par conséquent la variable utilisable en tant que telle. La valeur dépend du type de données de la variable: une variable peut être de type nombre (octet, booléen vrai-faux, nombre entier court ou long, nombre décimal etc.), de type chaîne de caractères, de type date, de type variant qui est un type alphanumérique flexible par défaut, d'un type objet, ou encore d'un type lui-même défini par l'utilisateur. Une instruction de déclaration **Dim**, **Private**, **Public** ou **Static** donne naissance à une variable en lui attribuant à la fois un nom, une portée et un type de donnée.

Public Client As String déclare une variable publique (accessible dans tous les modules) nommée Client et de type chaîne de caractères

Dim Signature As Paragraph déclare une variable-objet nommée Signature de portée locale représentant un objet de la classe **Paragraph**.

Une variable est vide lors de sa déclaration. Une valeur lui est donnée par une instruction d'affectation avec le signe =

Client = "Monsieur Durand"

Set Signature = ActiveDocument.Paragraphs.Last

Le mot-clé réservé Set est obligatoire pour affecter une variable-objet.

VBA permet une déclaration implicite des variables non objets au moment de leur première affectation. Les instructions de déclaration deviennent obligatoires si l'instruction **OPTION EXPLICIT** est placée en tête d'un module

De nature voisine, une Constante est aussi une donnée nommée, contenant une valeur d'un type de donnée non objet, de portée locale ou générale mais non modifiable contrairement à une variable. Elle est déclarée par l'instruction **Const**. Les constantes sont utilisées pour améliorer la lisibilité du code

Public Const Bleu As Long = 16711680 déclare une constante globale de type entier long représentant la couleur bleue. C'est la même valeur que la

constante prédéfinie **wdColorBlue**. VBA contient un grand nombre de constantes prédéfinies.

Quand un programme « ne marche pas » comme on s’y attendait, neuf fois sur dix c’est une erreur de programmation dans le nom, la portée ou le type de données d’une variable. Les outils d’aide à la programmation ont pour charge essentielle de surveiller les variables.

5.6 Procédure

Une procédure est une entité élémentaire de programmation commandant l’exécution d’une tâche définie. Toute procédure comporte un début et une fin. Un programme est généralement composé d’une procédure principale qui exécute d’autres procédures plus simples. On dit qu’une procédure en appelle une autre : le contrôle d’exécution est passé de la procédure appelante à la procédure appelée puis une fois cette dernière terminée, de nouveau à la procédure appelante au point quitté. En VBA les procédures sont regroupées dans un module. Dans Word, un module vba peut être stocké dans un modèle de document (.dot) ou dans un document (.doc)

Le type le plus général de procédure est la procédure **Sub** (comme sous-programme) entre **Sub NomDeProcédure()** et **End Sub** ; les parenthèses à la fin du nom de la procédure servent à préciser le nombre, les noms et les types des données à passer en arguments lors d’un appel à cette procédure. Les parenthèses sont obligatoires même si la procédure s’exécute sans arguments. L’instruction **Sub** peut être précédée de l’instruction **Public** (par défaut) ou **Private** pour préciser la portée de la procédure : accessible dans toute la portée du projet ou dans son module seulement.

Pour qu’une procédure **Sub** soit une macro Word, il faut qu’elle soit publique et qu’elle s’exécute sans argument. Toutes les autres procédures seront invisibles dans la boîte de dialogue Macros **ALT+F8** ou **OUTILS - MACRO - MACROS**

5.7 Fonction

Une fonction est un type particulier de procédure qui renvoie une valeur. Le langage VBA dispose déjà d’un grand nombre de fonctions préprogrammées. Ce sont notamment les fonctions de calcul numérique, les fonctions sur les chaînes de caractères, les fonctions de conversions entre types de données etc.

Une fonction définie entre **Function NomDeFonction(arguments)** et **End Function** s’utilise exactement comme une fonction préprogrammée.

5.8 Structures de contrôle

Les structures de contrôle sont les instructions du langage qui contrôlent l’exécution d’un programme. Elles rompent l’exécution séquentielle linéaire des autres instructions. Tout langage de programmation dispose d’un lot plus ou moins riche de structures de contrôle : tests, boucles, sauts, gestions des

erreurs etc. C'est grâce à ses structures de contrôle qu'un langage peut traiter un algorithme de résolution d'un problème complexe.

Une macro VBA obtenue par l'enregistreur de macros est une suite d'instructions sans structure de contrôle. Ce n'est qu'à partir de l'utilisation des structures de contrôle qu'on peut réellement parler de programmation

Les principales structures de contrôle VBA sont :

5.8.1 Les conditions

<pre>If <condition> Then Instructions End If</pre>	<pre>If <condition> Then Instructions Else Autres Instructions End If</pre>	<pre>If <condition1> Then Instructions1 ElseIf <condition2> Then Instructions2 ElseIf <conditions3> Then Instructions3 Else Autres Instructions End If</pre>
--	---	--

5.8.2 Les boucles

<pre>For Index = x To y [Step Z] Instructions Next Index</pre>	<pre>For Each obj In collection Instructions Next obj</pre>	
<pre>Do While <condition> Instructions Loop</pre>	<pre>Do Until <condition> Instructions Loop</pre>	<pre>While <condition> Instructions Wend</pre>
<pre>Do Instructions Loop While <condition></pre>	<pre>Do Instructions Loop Until <condition></pre>	<pre>Do Until While <condition1> Instructions Loop Until While <condition2></pre>

Les instructions Exit For ou Exit Do permettent de sortir d'une boucle For...Next ou d'une boucle Do...Loop

5.8.3 Les sélections

```
Select Case Expression
    Case valeur1
        Instructions1
    Case valeur2, valeur3
        Instructions2
    Case valeur4 To valeur5
        Instructions3
```

```
Case Is > valeur6
    Instructions4
Case Else
    Autres Instructions
End Select
```

5.8.4 Les sauts

<pre>GoTo Etiquette ... Etiquette: Instructions</pre>	<pre>GoSub Etiquette ... Etiquette: Instructions Return</pre>	<pre>On Expression GoTo E1, E2 On Expression GoSub E1, E2 ... E1: Instructions Return E2: Instructions Return</pre>
---	---	---

5.8.5 La gestion des erreurs

<pre>On Error Resume Next</pre>	<pre>On Error GoTo Etiquette ... Etiquette: Instructions Resume [Next Etiq2]</pre>	<pre>On Error GoTo 0</pre>
---------------------------------	--	----------------------------

5.8.6 Les blocs d'instructions par objets

```
With objet
    Instructions liées à l'objet
        .propriété
        .méthode
End With
```

5.9 Événement

Un événement est une action ou un message auquel un objet est susceptible de répondre en déclenchant une procédure. L'appui sur une touche, l'action de cliquer sur un bouton, la fermeture d'un document sont des événements. Le modèle objet Word dispose d'assez peu d'événements pouvant être interceptés en VBA. La programmation événementielle concerne surtout les contrôles : boutons, zones de texte, cases à cocher, listes déroulantes et autres contrôles ActiveX qui peuvent être incorporés directement dans un document Word ou dans une UserForm (une fenêtre ou une feuille complémentaire hors document).

6 Index des illustrations

Écran 1, Dialogue "Champ"	6
Écran 2, Dialogues Insertion - Champ – Options.....	12
Écran 3, Dialogue "Formule"	15
Écran 4, tableau calculé	15
Écran 5, Menu outils - Macro - Nouvelle macro.....	19
Écran 6, Dialogue Enregistrer une macro	19
Écran 7, Dialogue "Personnaliser le clavier"	20
Écran 8, Enregistrement de macro en cours.....	20
Écran 9, Dialogue "Collage spécial"	21
Écran 10, éditeur VBA.....	21
Écran 11, outil en suppression.....	22
Écran 12, Dialogue "Personnaliser" - Barres d'outils	23
Écran 13, Dialogue "Personnaliser" – Commandes	24
Écran 14, Dialogue "Personnaliser" - Modifier l'image du bouton	25
Écran 15, L'éditeur de boutons	25