



A l'attention des professionnels confrontés au travail avec du texte

Bienvenue à la première édition de **OUT OF OFFICE**, une newsletter mensuelle destinée aux traducteurs professionnels, éditeurs, réviseurs, auteurs et quiconque amené à travailler avec des textes à des fins professionnelles.

Je tiens à m'excuser à l'avance pour cet édito un peu long, mais il y a beaucoup à dire dans ce premier numéro. Les futurs éditos seront plus courts.

Cette newsletter aborde exclusivement l'utilisation de Microsoft Office à des fins professionnelles. Elle n'a pas pour objectif d'encenser ou de critiquer cette suite de logiciels, mais plutôt d'examiner nos principaux outils de travail, de comprendre des comportements et de trouver des solutions à des problématiques quotidiennes spécifiques. Une importante partie de cette newsletter est consacrée à VBA (Visual Basic for Applications) et comment écrire des macros afin d'automatiser des tâches répétitives et d'accomplir des tâches routinières le plus rapidement possible, mais d'un point de vue de linguiste et non de programmeur.

OUT OF OFFICE se divise en rubriques traitant chacune un composant différent de MS Office :

- **OH MY WORD !** fournit des outils, des trucs et astuces sur Word® ;
- **ALWAYS EXCEL** offre la même chose pour Excel® ;
- **POWER TO THE POINT** discute de PowerPoint® ;
- **THE OUTLOOK LOOKS GOOD** traite d'Outlook® ;
- **VBA FOR EVER !** propose des solutions et des routines pour automatiser des tâches répétitives ;
- **ANYTHING TO ADD-IN ?** explore les outils utiles disponibles qui nous aident dans nos tâches quotidiennes.

Chaque numéro contient au moins deux de ces rubriques et apporte des solutions réelles à des problèmes ordinaires. Les solutions et conseils prodigués doivent s'appliquer à toutes les versions MS Office® post-2007 de l'environnement Windows®. Par exemple, les raccourcis proposés sont basés sur Windows® (à condition de ne pas les avoir redéfinis vous-même) et ne fonctionnent pas dans un environnement Mac®. Les utilisateurs d'autres environnements peuvent essayer les solutions suggérées et m'indiquer si elles ont fonctionné pour eux en envoyant un email à outoffice.newsletter@gmail.com avec « J'ai essayé cela dans un environnement non-Windows® » dans la ligne Objet. Merci d'inclure les captures d'écran des messages d'erreur que vous voyez afin que je puisse trouver une solution au problème.

Si vous souhaitez qu'une thématique particulière soit expliquée dans cette newsletter, n'hésitez pas à envoyer un email à outoffice.newsletter@gmail.com avec « Comment faire pour... ? » dans la ligne Objet accompagné d'une explication détaillée de ce que vous aimeriez obtenir et si possible des fichiers échantillons.

Cette newsletter ne fait la publicité d'aucun produit ou service et ne vous envoie aucun matériel promotionnel. Une fois par mois, les adhérents reçoivent un fichier Zip contenant la version PDF de cette newsletter accompagnée des fichiers .xslm, .bas, .frm ou .frx liés au contenu de la newsletter.

Cette newsletter peut et doit être considérée comme une partie intégrante de votre **développement professionnel continu**. Aujourd'hui, Internet constitue une source de contenus principalement gratuits. Le seul inconvénient est que vous êtes sans doute obligé de passer beaucoup de temps pour trouver ce que vous cherchez. Rien ne remplacera jamais votre savoir-faire personnel. Vous êtes la personne la mieux placée pour comprendre les problèmes que vous devez résoudre. Par conséquent, au lieu de créer des vidéos et des blogs, cette newsletter a pour but de vous apporter des compétences supplémentaires qui vous aident seul(e) à résoudre certains problèmes rencontrés dans votre travail quotidien. Si toutefois vous n'y parvenez toujours pas, vous pouvez envoyer un email avec « Comment faire pour... ? » dans la ligne Objet.

Le premier numéro est gratuit, mais les numéros ultérieurs seront adressés seulement aux adhérents payants. Pour un modeste forfait de 27 € par an, vous recevrez un numéro par mois. Veuillez exprimer votre intérêt en écrivant à l'adresse ci-dessus avec « S'abonner » dans la ligne Objet en précisant la langue de votre choix (anglais, français, allemand et/ou espagnol). Vous recevrez un Contrat de Confidentialité stipulant clairement que la newsletter ne doit pas être rendue publique et que son contenu ne doit pas tomber dans le domaine public, que ce soit en tout ou partie, sans notre accord écrit préalable. Cependant, vous pouvez utiliser les astuces et trucs suggérés à des fins professionnelles. Le règlement de l'abonnement à la newsletter s'effectue uniquement par Paypal. Tout règlement de votre part confirme votre acceptation du Contrat de Confidentialité et des conséquences en cas de manquement.

La moitié de tous les bénéfices seront reversés à des œuvres caritatives (Solidarités, Translators Without Borders, Water Aid et Médecins Sans Frontières).

Bonne lecture !

L'équipe OUT OF OFFICE

Conventions

Cette mise en forme bleue sera utilisée pour évoquer les éléments du ruban et d'autres options du logiciel.

Les raccourcis sont insérés dans les paragraphes à l'aide de ce format.

Cette mise en forme rouge sera utilisée pour évoquer les combinaisons clés. Si la combinaison vous indique d'appuyer sur deux ou plusieurs touches à la fois, un trait d'union sera inséré entre les touches en question, par exemple **Ctrl-F** pour ouvrir la fonction de base rechercher en opposition à **Alt F** pour accéder à l'onglet **Fichier** sur le ruban. Les abréviations suivantes seront utilisées pour se référer aux touches de votre clavier :

- Alt la touche Alt sur la gauche
- AltGr la touche Alt à droite
- Ctrl n'importe quelle touche de commande
- Maj n'importe quelle touche Majuscule
- AppK la touche menu située à droite de la touche AltGr avec un petit dessin dessus
- Entrée la touche Retour chariot (Entrée)
- Tab la touche Tabulation

À des fins de lisibilité, toutes les lettres du clavier sont indiquées en majuscules, mais si elles DOIVENT être en majuscules, cela est indiqué par une mention de la touche Majuscule.

Les références aux touches numériques (pas les touches de votre clavier numérique) combinées à la touche Majuscule pour obtenir un autre caractère sont effectuées en spécifiant le caractère et l'usage de la touche Majuscule. Par exemple, l'usage de la touche « = » pour mettre la sélection en Exposant sera écrit **Ctrl-Maj+.**

Les références aux caractères Unicode sont effectuées en utilisant **Alt-numéro de combinaison**. Le numéro de combinaison fait référence aux touches du clavier numérique (la touche Verr Num doit être activée).

Ce format de police sera utilisé pour faire référence à tout ce qui est associé au codage, que ce soit VB, XML ou SQL.

Tout le code sera contenu dans les paragraphes qui utilisent ce format.
'Cependant, les commentaires dans le codage commencent par une apostrophe et suivent ce format (police verte)

Les captures d'écran sont extraites de notre version Office 2013 et sont fournies uniquement pour illustrer l'emplacement des éléments sur le ruban. Votre version d'Office peut être différente et/ou votre ruban peut être personnalisé. Les petits cercles ou rectangles verts qui entourent les éléments en question ne font pas partie de la suite Office, mais permettent de situer rapidement le sujet concerné.

Les commentaires et les notes sont présentés dans ce format. Ils ne font pas partie du flux textuel, mais contiennent des remarques importantes sur le sujet concerné.

Dans ce numéro :

OH MY WORD !

- Afficher les marques de mise en forme
- Afficher le contenu du document
- Afficher
- Rechercher et remplacer

VBA FOR EVER !

- Une approche linguistique au VBA
- Notre première Macro
- Créer des profils dans Word®

Dans le prochain numéro

THE OUTLOOK LOOKS GOOD !

- Gérer votre courrier indésirable
- Ajouter un contact
- Créer des règles
- Éditer des règles

VBA FOR EVER !

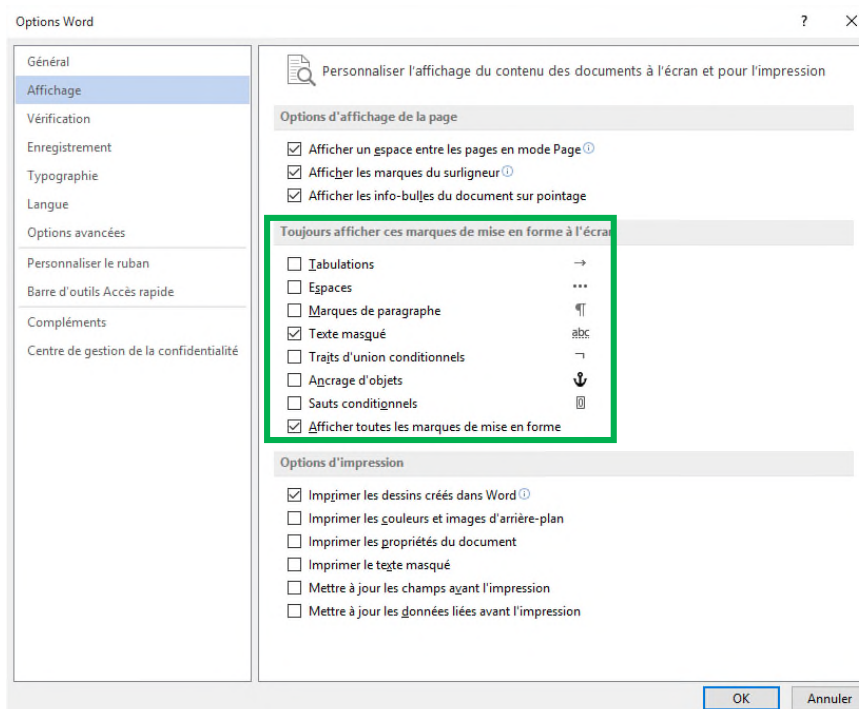
- Liste des pièces jointes (1^{ère} partie)

Dans ce tout premier numéro, commençons par les fondamentaux.

Afficher les marques de mise en forme

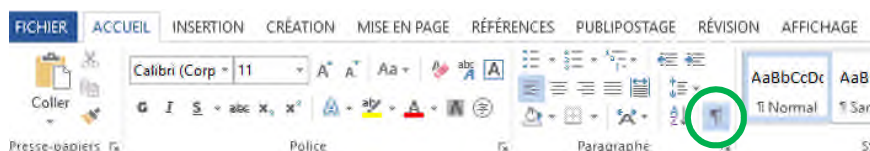
Il est conseillé de voir ce qui apparaît réellement sur votre page, notamment ce qui est non-imprimable et invisible. Certaines personnes peuvent croire que voir des espaces, des marques de tabulation et le reste est un vrai défi, mais c'est important pour tous ceux et celles qui travaillent avec du texte. Souvenez-vous que vous pouvez toujours activer ou désactiver cette fonction selon vos besoins.

Sur le Ruban, allez dans **Fichier > Options** puis dans la section « **Affichage** », sélectionnez les éléments que vous voulez voir.



Dans la section centrale, « **Toujours afficher ces marques de mise en forme à l'écran** », vous pouvez décider des marques que vous voulez voir et, sur la droite, vous voyez la marque qui représentera chaque option sur votre écran.

La plupart des utilisateurs préfèrent utiliser l'icône sur le ruban située sur l'onglet « **Accueil** », dans le groupe « **Paragraphe** » pour activer/désactiver cette option.



Vous remarquerez que cette icône est bleutée lorsqu'elle est activée.

Il existe des limites à utiliser uniquement l'icône du ruban pour afficher ou masquer les marques de mise en forme.

Dans la première capture d'écran, deux options sont sélectionnées : « **Texte masqué** » et « **Afficher toutes les marques de mise en forme** ». L'icône sur le ruban affecte uniquement l'option « **Afficher toutes les marques de mise en forme** »

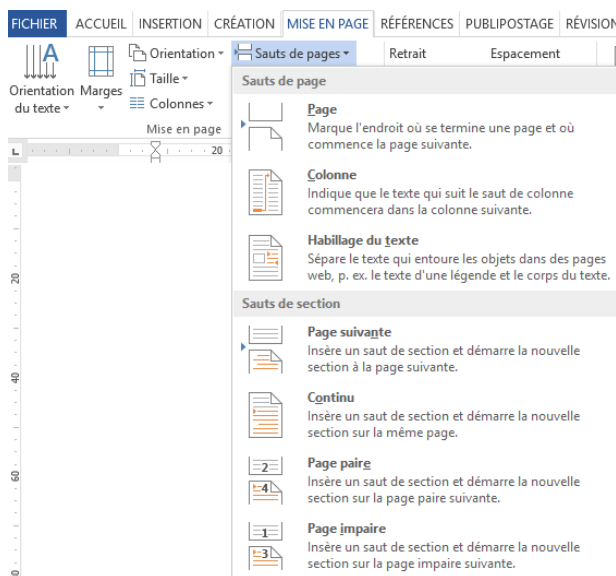
ce qui signifie que toutes autres options que vous avez sélectionnées dans la fenêtre des options ne seront pas affectées. Dans ce cas, l'option « **Texte masqué** » restera visible que l'icône soit activée ou désactivée.

Pourquoi les marques de mise en forme sont-elles utiles ?

Plusieurs raisons expliquent l'utilité de voir les invisibles. Vous pouvez entre autres :

- a) Distinguer les sauts de page des sauts de sections.

Pour insérer un saut de page, sur le ruban, onglet « **Mise en page** », section « **Mise en page** », « **Sauts de page** », la première « **Page** » insèrera un saut de page.



Pour insérer un saut de page, vous pouvez le faire plus rapidement en utilisant le raccourci **Ctrl-Entrée**.

Pour insérer un saut de section, sélectionnez l'une des options inférieures.

Il n'existe pas de raccourcis pour les sauts de section, mais vous pouvez en personnaliser un pour vos besoins.

Autrement, vous pouvez appuyer sur la touche **Alt** pour activer la commande clavier du ruban, puis appuyer sur **G** (pour accéder à l'onglet « **Mise en page** »), puis sur **S** (pour activer l'option « **Sauts de page** » dans la section « **Mise en page** ») et sélectionner **N**, **O**, **E** ou **I** en fonction de vos besoins.

Un avantage du saut de section est que vous pouvez définir, entre autres choses, des nouveaux en-têtes ou pieds de page pour une section de votre document.

- b) Voir les propriétés des sauts de section.

Comme indiqué ci-dessus, il existe quatre types de saut de section lorsque les marques de mise en forme sont affichées, à condition que l'espace après la marque de paragraphe soit suffisant, vous devriez voir le type de saut de section utilisé.

..... Saut de section (page suivante)

Saut de section (continu)

- c) Distinguer les retours à la ligne et les retours forcés à la ligne.

Un retour forcé à la ligne est identifié par ce symbole ↵

Pour insérer un retour forcé à la ligne, utilisez **Maj-Entrée**.

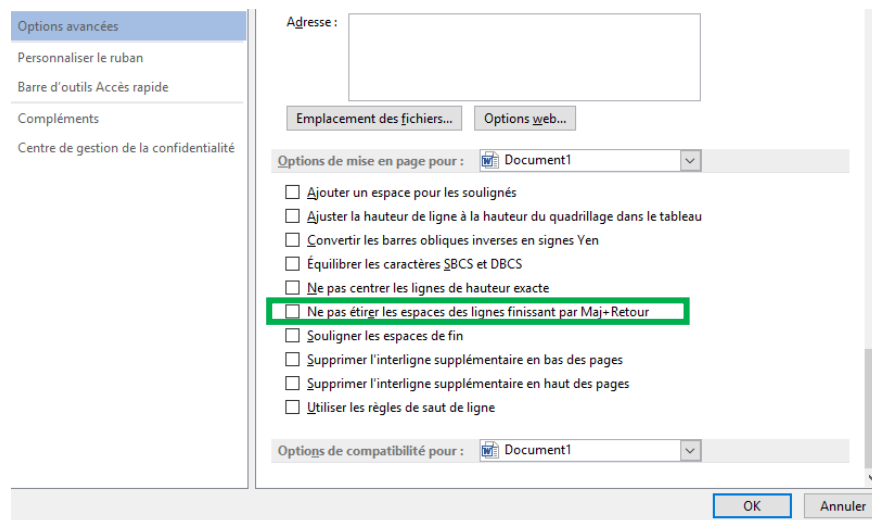
La marque de fin de paragraphe normale ou le retour à la ligne est ¶.

La différence entre un retour forcé à la ligne et un retour à la ligne normal est qu'un paragraphe est défini, entre autres choses, lorsqu'il y a :

- un **Retrait de première ligne** (la position de départ de la première ligne comparée à la marge) ;
- un **Retrait suspendu** (la position de départ des autres lignes du paragraphe) ;
- un **Alignement** (gauche, droit, centré, justifié) ;
- un **Espacement** (avant, après, interlignes) ;

Un retour forcé à la ligne signifie que le paragraphe est le même et, en tant que tel, votre ligne commencera là où le « **Retrait suspendu** » est défini (non pas le « **Retrait de première ligne** »). De plus, vous ne pouvez pas modifier les propriétés de cette partie du paragraphe sans affecter le paragraphe entier.

Si votre paragraphe est défini comme « Justifié », le texte s'étendra jusqu'à la fin de la ligne sauf si vous avez défini autrement ce comportement par défaut dans **Fichier > Options > Options avancées** section « **Options de mise en page** » pour, option « **Ne pas étirer les espaces des lignes finissant par Maj-Entrée** ».



Ceci vise à illustrer que l'usage d'un retour chariot a un effet sur la façon dont vous affichez votre texte dans un paragraphe justifié en fonction de la configuration des options. Vous pouvez constater que le texte n'est pas étendu lorsque l'option est sélectionnée.

Ceci vise à illustrer que l'usage d'un retour chariot a un effet sur la façon dont vous affichez votre texte dans un paragraphe justifié en fonction de la configuration des options. Vous pouvez constater que le texte n'est pas étendu lorsque l'option est sélectionnée.

- d) Voir comment plusieurs onglets ou espaces sont insérés.

Bien qu'il soit possible d'automatiser le processus rechercher/remplacer pour de multiples onglets et espaces rebelles, il peut être utile de voir le contenu du texte et prendre une décision éclairée sur la base du contexte spécifique.

- e) Localiser les points d'ancrage des zones de texte et des formes.

Un point d'ancrage montre le point d'insertion d'un objet qui n'est pas englobé avec le texte. Cet objet est contenu dans une couche différente de la couche d'origine. Pour le rendre visible, l'objet en question doit être sélectionné. Le point d'ancrage est représenté par une petite ancre, en bordure de marge gauche, à côté du paragraphe où vous avez effectué l'insertion.



Cela ne signifie pas que l'objet sera près de l'ancre (il a pu être déplacé). Cependant, en supprimant ce paragraphe, vous supprimerez également le(s) objet(s) associé(s).

Le point d'ancrage peut être sélectionné avec la souris et glissé vers un autre endroit sans déplacer l'objet.

Si vous sélectionnez plusieurs objets, les points d'ancrage ne seront pas visibles.

Si vous sélectionnez un paragraphe entier et qu'un seul objet lui est associé, le point d'ancrage deviendra visible. Toutefois, si plusieurs points d'ancrage sont positionnés au même point, ils ne seront pas visibles.

Il est donc conseillé de localiser tous les objets contenus dans le document et de s'assurer que leurs points d'ancrage ne sont pas associés à un paragraphe que vous pourriez supprimer. Ici, le mot « Objet » fait référence à des dessins, des formes, des zones de texte et tout autre élément qui peut être inséré et défini autrement que « [Aligné sur le texte](#) ».

- f) Éditer le texte masqué.

Le texte masqué est représenté par une petite ligne en pointillé.

Ce texte est visible ¶

Mais ce texte ne l'est pas ¶

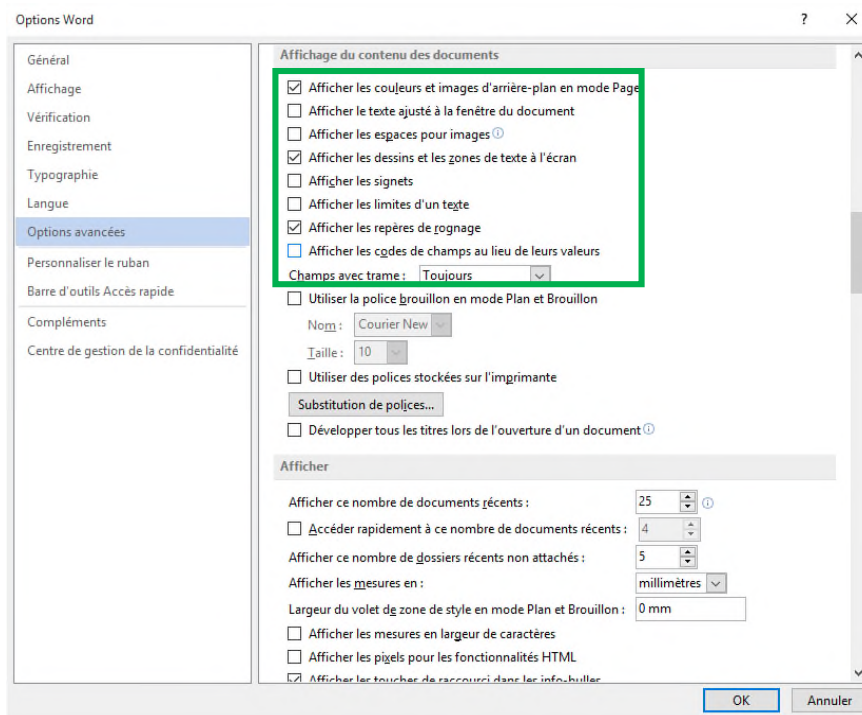
La seule façon de voir et d'éditer le texte masqué est d'utiliser les marques de mise en page afficher/masquer, ce qui est très utile pour vérifier si le document contient des champs masqués tels que des entrées d'index ou des champs TC. Nous aborderons les champs dans un prochain numéro.

Heureusement, Word® offre plusieurs options pour souligner le texte avec différents types de lignes en pointillé, mais rien ne correspond aux fines lignes en pointillé du texte masqué ; par conséquent, la confusion est peu probable.

Afficher le contenu du document

Certaines marques ne sont pas des marques de mise en forme, mais vous pouvez avoir besoin de les voir. La capacité de voir les zones de texte, les signets, les habillages de texte, les limites de texte et le contenu des champs et les trames.

Pour se faire, allez à **Fichier > Options > Options avancées** dans la section « **Affichage du contenu des documents** ».



Pourquoi est-il utile de voir les éléments du contenu d'un document ?

- a) Obtenir un rendu précis du contenu de la page avant l'impression.

L'option « **Afficher les couleurs et images d'arrière-plan en mode Page** » permet à l'utilisateur d'avoir un réel rendu sur l'écran du document sans avoir à demander un « **Aperçu avant impression** ». Cependant, comme indiqué dans le nom de l'option, ceci s'applique seulement au mode « **Page** ».

- b) Préserver l'ajustement du texte en mode Plan

L'option « **Afficher le texte ajusté à la fenêtre du document** » permet de conserver l'ajustement de la page d'impression pendant que vous travaillez en mode « **Plan** ».

- c) Ignorer les images

L'option « **Afficher les espaces pour images** » affiche des zones vides à la place des images. Cette option peut être utile lorsque vous travaillez en mode « **Page** » sur un document où vous n'avez pas besoin de voir les images ou si les images/illustrations ne sont pas prêtes à l'insertion.

L'infobulle Microsoft® indique que cette option permet d'améliorer le défilement du document.

- d) Voir le texte contenu dans des zones de texte

Cette option s'applique également au mode « **Page** ». L'option « **Afficher les dessins et les zones de texte à l'écran** » vous permet de visualiser des objets (zones de texte et formes) qui sont sur un plan différent du texte (différente couche). Les autres modes tels que « **Plan** » ou « **Web** » n'autorisent pas les couches multiples et ne les affichent pas.

e) Voir l'emplacement des signets

Les signets sont extrêmement utiles même si peu de personnes les utilisent au maximum de leur capacité. L'option « **Afficher les signets** » indiquera l'emplacement du signet dans votre texte non seulement en mode « **Page** », mais dans tous les modes sauf en « **Mode Lecture** ». La visibilité des signets permet de garantir qu'ils ne seront pas supprimés pendant la révision ou le travail sur le document.

Ils sont représentés dans le texte par ce symbole qui ne peut pas être sélectionné :



f) Voir les limites du texte

L'option « **Afficher les limites d'un texte** » affichera des lignes en pointillé en forme de rectangles autour de chaque paragraphe, d'une marge à l'autre marge et autour des formes et des zones de texte. Encore une fois, ceci s'applique uniquement au mode « **Page** » et vous permet de voir l'espace avant et après chaque paragraphe.



g) Afficher les champs

Les champs forment du contenu automatisé qui peut provenir du document même, tel qu'un numéro de page ou la table des matières, ou de données externes, telles que d'autres documents Word® ou d'autres applications MS Office®. Dans les deux cas, il est toujours préférable de les faire ressortir (à condition qu'ils ne soient pas masqués dans des champs tels que des Entrées d'index). L'option « **Champs avec trame** » vous permet de spécifier si les champs sont ombrés ou non.

h) Voir la source des champs

Compte tenu de la grande variété de champs, vous avez parfois besoin de visualiser la fonctionnalité du champ et ce qui lui est associé, quel que soit son contenu.

<code>{TOC\o-1-5\h\z}</code>	Aucune entrée de table des matières n'a été trouvée.
<code>{AUTOTEXT-"Binomial-Theorem"-*MERGEFORMAT}</code>	$(x + a)^n = \sum_{k=0}^n \binom{n}{k} x^k a^{n-k}$
<code>{KEYWORDS-"test,demonstration,OutOfOffice"-*MERGEFORMAT}</code>	test,demonstration,OutOfOffice

Selon le type de travail que vous devez réaliser sur un document donné, vous pouvez avoir besoin de modifier ces options assez régulièrement. Dans ce numéro, une fois que vous aurez appris les fondamentaux de VBA, vous apprendrez comment écrire une macro pour définir précisément les options que vous souhaitez appliquer au travail concerné.

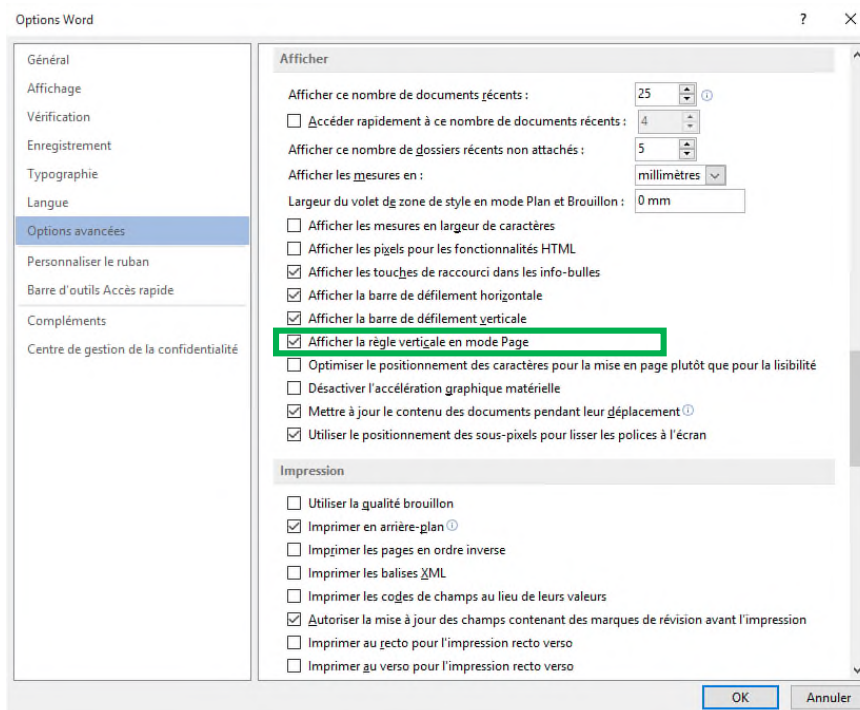
Afficher

Ce groupe situé dans l'onglet « **Afficher** » du ruban contient trois éléments essentiels lorsqu'on travaille sur des documents.

1. Règles

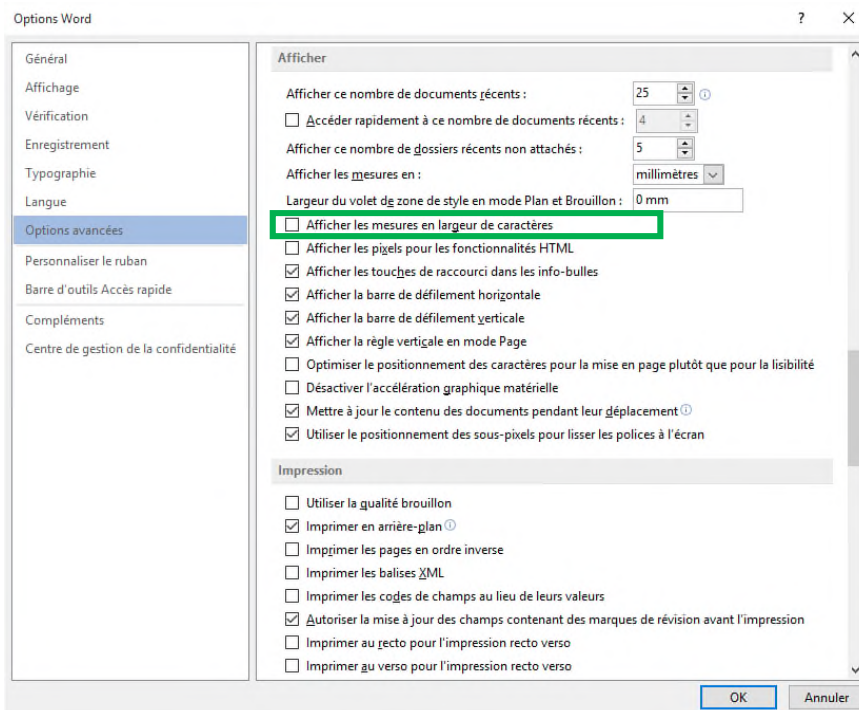
La première option dans **Affichage > Afficher** est « **Règle** ». Même si le mot est au singulier, deux règles sont créées simultanément, une règle horizontale et une règle verticale. Cependant, le mode d'affichage de la règle verticale n'est pas paramétré ici, mais dans **Fichier > Options > Options avancées** dans la section « **Afficher** »

en cochant l'option « **Afficher la règle verticale en mode Page** ». Ceci signifie que vous ne disposez pas d'un moyen d'afficher uniquement la règle verticale et que cette règle est disponible uniquement en mode « **Page** ».



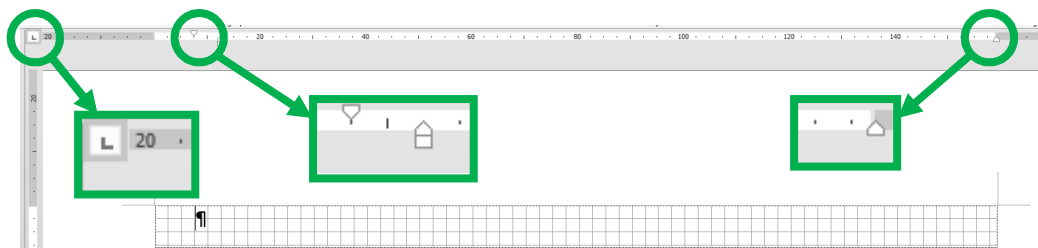
Les numéros que la ou les règles afficheront se basent sur l'unité de mesure que vous avez définie dans vos options. La fréquence de numérotation dépend également de votre unité de mesure. Par exemple, vous avez décidé d'utiliser des pouces, vos règles afficheront chaque unité avec des marqueurs à chaque 1/8ème de pouce alors que si vous avez opté pour des centimètres, vos marqueurs se trouveront à chaque quart de centimètre.

Il y a une exception. Si dans **Fichier > Options > Options avancées** dans la section « **Afficher** » vous avez coché l'option « **Afficher les mesures en largeur de caractères** », le chiffre indiquera le nombre de caractères. La majorité des polices utilise différentes largeurs de caractères, mais certaines polices, comme Courier, sont à chasse fixe.



L'option « **Afficher les mesures en largeur de caractères** » n'est disponible que si vous travaillez avec des langues asiatiques telles que le japonais qui utilisent des largeurs standard de caractères.

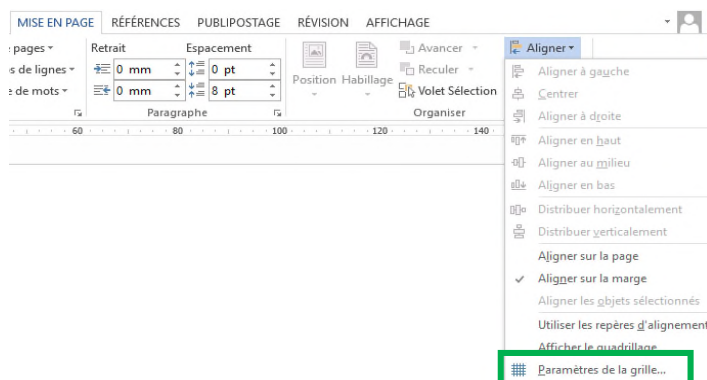
En plus de chiffrer la distance entre les marges gauche et droite et les marges supérieure et inférieure, les règles fournissent également d'autres informations extrêmement utiles. Lorsque votre curseur est placé dans n'importe quel paragraphe, la règle vous indique également l'emplacement du Retrait de la première ligne et du Retrait suspendu (tous les deux formant le Retrait à gauche), le Retrait à droite et toute marque de tabulation définie. L'extrême gauche de la règle contient également un outil très utile qui vous permet de définir le type de marque de tabulation que vous voulez utiliser.



2. Quadrillages

La seconde option dans le groupe « **Afficher** » sur l'onglet « **Affichage** » vous permet de voir les « **Quadrillages** ». Cet outil est utile pour vérifier que tout est correctement aligné dans votre document. Cependant, cela peut occasionner une fatigue oculaire et vous devez effectuer une configuration correcte.

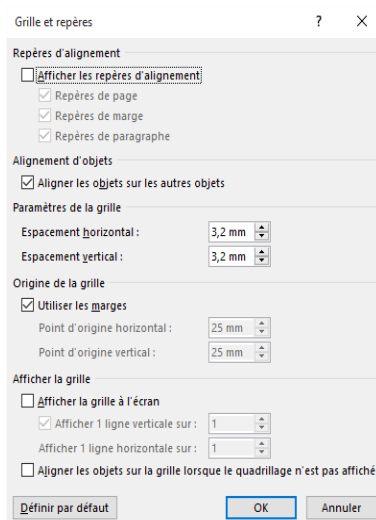
Pour la configuration, allez à l'onglet « **Mise en page** » sur le ruban, dans le groupe « **Organiser** », cliquez sur l'option « **Aligner** » et sélectionnez « **Paramètres de la grille** » tout à la fin de la liste des options.



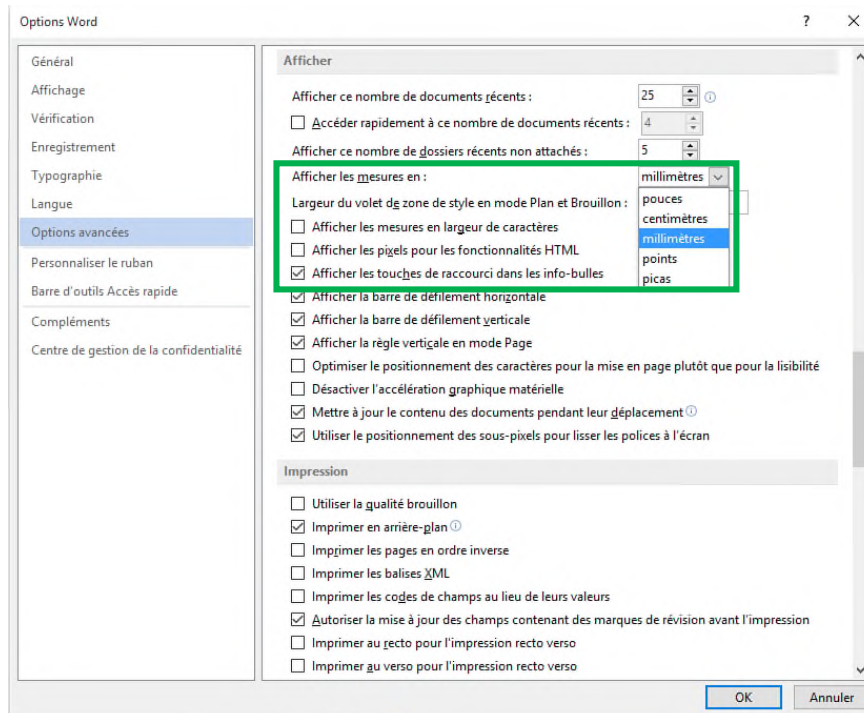
Une option consiste à utiliser la touche Alt et de procéder à la combinaison de touches suivante pour accéder aux paramètres de la grille sans utiliser la souris : **Alt P A A G** (l'absence du trait d'union entre les combinaisons de touche signifie que vous n'avez pas besoin de maintenir la touche Alt enfoncée)

Cette action permet d'ouvrir la boîte de dialogue suivante afin de définir précisément votre grille (espace entre les lignes verticales, espace entre les lignes horizontales, affichage ou non de la grille et les lignes à afficher, point d'origine de la grille, etc.). Si vous activez la visibilité de la grille, votre page de travail ressemblera à une page de papier quadrillé.

Si vous préférez travailler sur un arrière-plan avec des lignes, décochez simplement l'option « **Afficher 1 ligne verticale sur** » dans la section « **Afficher la grille** » sous « **Afficher la grille à l'écran** » (qui doit être coché).



Sur la capture d'écran ci-dessus, vous pouvez voir que les mesures sont en millimètres. Vous pouvez définir l'unité que vous voulez utiliser pour tous vos besoins dans **Fichier > Options > Options avancées** dans la section « **Afficher** ».



Quelle que soit l'unité que vous choisissez, même pour les options par défaut qui utilisent d'autres mesures, vous pouvez toujours spécifier l'unité que vous voulez et Word® la changera automatiquement. Par exemple, si vous ne savez pas du tout ce qu'un point représente, vous pouvez définir l'espacement du paragraphe à « 0,5 cm ». Vous devez intégrer l'unité de mesure abrégée utilisée par Word®, mais Word® la convertira en « 14.2 pt » dès que vous appuierez sur la touche « OK ».

3. *Volet de navigation*

Le Volet de navigation vous permet de voir la structure du document. Pour l'activer, sur le ruban, allez à l'onglet **Affichage**, groupe « **Afficher** » et cochez « **Volet de navigation** ».

Vous pouvez accéder au volet de navigation en faisant un **Ctrl-F** pour lancer la fonction de recherche de base. Cette action ouvre le volet de navigation de l'onglet « **Résultats** ». Cliquez sur l'onglet « **Titres** » (ouverture d'onglet par défaut depuis le ruban) ou « **Pages** ».

L'onglet « **Titres** » affichera la structure du document (plan) sur la base des niveaux de chaque paragraphe (de 1 à 9). Tout ce qui se trouve au « Niveau du corps » n'apparaîtra pas. Cet outil est pratique pour naviguer rapidement vers le niveau du paragraphe en question sans faire défiler le document.

Cette vue est particulièrement utile pour savoir à quoi ressemble votre table des matières. Cependant, cette vue affiche tous les paragraphes jusqu'au niveau 9, vous préférez la plupart du temps n'afficher que les trois ou quatre premiers niveaux de votre table des matières. Il n'existe pas de méthode pour diminuer le nombre de niveaux de paragraphe affichés dans cet onglet.

Vous pouvez aussi utiliser cette vue de manière très efficace pour élaborer votre « **liste multiniveaux** ». Nous aborderons les listes multiniveaux avec numérotation dans un prochain numéro.

L'onglet « **Pages** » propose un aperçu des pages et vous permet de naviguer rapidement à travers les pages. Malheureusement, vous ne pouvez ajuster que la largeur du volet, mais pas la taille de l'aperçu de la page, ce qui rend difficile la lecture du texte.

Rechercher et remplacer

Comme indiqué plus haut, utilisez le raccourci **Ctrl-F** pour ouvrir le volet de navigation sur l'onglet « **Résultats** ». Cette fonction de recherche de base vous permet de chercher simplement une chaîne spécifique dans le texte.

Si vous voulez accéder à des options de recherche plus avancées, vous devez lancer la boîte de dialogue « **Rechercher et remplacer** ».

Ctrl-H ouvre la boîte de dialogue « **Rechercher et remplacer** » sur l'onglet « **Remplacer** ».

Ctrl-B ouvre la même boîte de dialogue, mais sur l'onglet « **Atteindre** ».

Autrement, sur l'onglet « **Résultats** » du volet de navigation, à droite du champ de recherche après la loupe, vous pouvez cliquer sur la petite flèche pour afficher le menu et sélectionner l'une des quatre premières options disponibles : « **Options...** », « **Recherche avancée...** », « **Remplacer...** » et « **Atteindre...** ».

Cliquez sur « **Options...** » pour ouvrir une autre fenêtre qui vous permet de définir des options de recherche. Les trois dernières options ouvrent une boîte de dialogue « **Rechercher et remplacer** » directement sur l'onglet de votre choix.

Si vous avez utilisé l'un ou l'autre raccourci indiqué ci-dessus, vous pouvez cliquer sur l'onglet « **Rechercher** ».

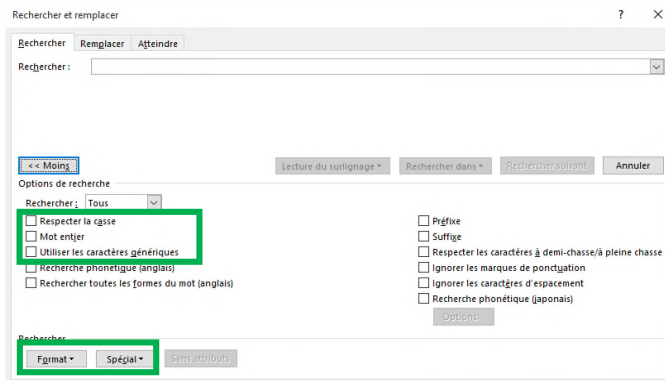
Si vous n'aimez pas utiliser la souris, un **Alt-R** vous amènera à l'onglet désiré.

Dans la plupart des boîtes de dialogue, les différentes composantes ont des légendes avec une lettre soulignée. Pour accéder à la fonction désirée, il vous suffit d'utiliser la combinaison **Alt + la lettre soulignée** pour activer cette fonction.



L'onglet de base « **Rechercher** » remplira la même fonction que la fonction Rechercher du volet de navigation. Cependant, vous pouvez cliquer sur « **Plus >>** » pour accéder à plus de fonctionnalités.

Là encore, si vous n'aimez pas utiliser la souris, vous pouvez développer la boîte de dialogue pour afficher les options de recherche en appuyant sur la combinaison **Alt-S**.



Les options disponibles dépendent de votre langue, mais les options de base sont disponibles pour toutes les langues :

- Rechercher
- Respecter la casse
- Mot entier
- Utiliser les caractères génériques

Même si les trois premières options semblent évidentes, les caractères génériques sont un autre problème. Nous les aborderons dans un prochain numéro. Pour l'instant, disons simplement qu'ils vous permettent de trouver, par exemple, toutes les occurrences de « tond » et « tend » dans un texte et de vérifier si vous n'avez pas inversé de lettres par inattention.

Sous les options, vous disposez de deux boutons qui vous permettent de rechercher un « **Format** » spécifique et un caractère « **Spécial** ». Le contenu de l'option « **Spécial** » varie si vous avez coché ou non la case des caractères génériques.

Les linguistes considèrent que la programmation s'adresse aux personnes qui aiment les maths. Une autre façon de percevoir le LANGAGE de programmation est simplement qu'il s'agit d'une langue avec des règles grammaticales destinées uniquement à la machine et pour lui dire ce qu'elle doit faire. Dans ce numéro, je présenterai les règles grammaticales fondamentales d'un point de vue linguistique. Le vocabulaire utilisé ne ressemblera pas à la terminologie geek standard des langages orientés objet (qui sera insérée entre parenthèses après le terme), mais à celui d'un linguiste.

La suite MS Office comprend un très puissant outil dissimulé en arrière-plan qui permet à l'utilisateur de créer des groupes d'actions spécifiques au sein d'un programme de la suite Office ou entre différents programmes. Vous pouvez même utiliser ce langage pour créer des programmes maison autonomes si vous utilisez Visual Basic 2010 ou toute version disponible au lieu de Visual Basic Editor intégré dans l'un des logiciels de MS Office.

Le VBE

Toutes les commandes que vous voulez que votre ordinateur exécute se feront à partir du VBE (Visual Basic Editor).

Pour accéder au VBE à partir d'un logiciel MS Office, appuyez sur **Alt-F11**.

Vous écrirez vos séquences d'instructions sur cette interface.

En haut à gauche, vous verrez la fenêtre du projet qui vous permet de sélectionner le fichier qui recevra vos instructions. Par exemple, si vous avez essayé le raccourci indiqué plus haut dans Word®, vous devriez voir au minimum « Normal » et le document que vous avez ouvert dans cette section.

Les instructions (ou macro-instructions) que vous écrirez dans « Normal » seront disponibles dans tous les documents basés sur votre modèle normal. Si vous décidez d'écrire vos instructions dans « Project (OutOfOffice_0000) », elles ne seront accessibles que lorsque ce fichier spécifique sera ouvert.

VBA

1. Composants

VBA signifie Visual Basics for Applications. Chaque logiciel individuel s'appelle une application (nous vivons dans un monde d'applis !). Dans chaque application, vous avez des composants spécifiques appelés objets.

Pour faire un comparatif avec notre monde phénoménal, prenons l'exemple de l'application « Arbre ». Un « Arbre » se compose d'éléments : feuilles, tronc, branches, bourgeons, etc. Ils forment les « objets » de l'application « arbre ».

Dans le VBA pour Word®, par exemple, certains objets sont représentés par « documents », « paragraphs », « tables », etc.

2. Collections et éléments

Ces composants peuvent être génériques (collection) ou spécifiques (éléments individuels).

Pour revenir à notre application « Arbre », nous pouvons avoir des « feuilles » ou une « feuille », des « branches » ou une « branche » selon qu'il est fait référence à tous les éléments de ce groupe ou à un élément spécifique.

Dans VBA pour Word®, par exemple, vous aurez « documents », « paragraphs », « tables », etc.

3. Hiérarchie

Les composants font partie d'une hiérarchie dans laquelle un composant comprend un autre composant de manière intrinsèque.

Dans notre application « Arbre », chaque arbre a un tronc, chaque tronc a des branches et chaque branche a des feuilles.

Dans VBA pour Word®, par exemple, vous ne pouvez pas avoir de cellules sans un tableau et vous ne pouvez pas avoir un tableau sans un document.

4. Adjectifs et verbes

Les éléments ont des adjectifs (propriétés) :

Dans notre application « Arbre », les feuilles peuvent avoir une forme et une couleur.

Dans VBA pour Word®, les lignes du tableau peuvent avoir des couleurs.

et des verbes (méthodes) :

Dans notre application « Arbre », vous pouvez compter et ajouter le nombre de feuilles sur une branche donnée.

Dans VBA pour Word®, vous pouvez compter, supprimer ou ajouter un paragraphe ou une section à un document.

5. Syntaxe

La syntaxe VBA regroupe des objets et leurs propriétés et les méthodes d'utilisation d'un point pour séparer des éléments.

Dans notre application « Arbre », pour spécifier que la 25^{ème} feuille de la 1^{ère} branche est verte, nous écrivons :

```
tronc.branches(1).feuilles(25).couleur = Vert
```

Dans VBA pour Word®, pour enregistrer un document et le fermer, nous écrivons :

```
Document.Close Save:=True
```

Dans l'exemple précédent, la méthode de « fermeture » comprend des options. Les options d'une méthode s'ajoutent après la méthode. Dans notre exemple, l'option « Enregistrer ». Nous aurions pu écrire :

```
Document.Save  
Document.Close
```

Cependant, plus vous écrivez de lignes d'instructions et plus l'ordinateur aura de difficulté et de lenteur à les exécuter.

6. Lexique

Chaque application possède son propre lexique, mais vous pouvez aussi définir le vôtre. VBA base son lexique sur l'anglais, quelle que soit votre langue de travail. Ces nouveaux termes (Variables) peuvent changer de valeur. Avant de pouvoir utiliser ces nouveaux termes, il est conseillé d'informer l'ordinateur de leur registre. Cela se fait en général en écrivant une phrase (énoncé) commençant par le mot clé « Dim », suivi du nom de votre propre terme (il ne doit pas contenir d'espace et ne pas être un terme utilisé par VBA à d'autres fins), suivi de « As » et son registre (le type de variable qu'il peut contenir) :

```
Dim Voici_un_chiffre As Integer  
Dim Voici_une_chaine_de_caractères_alphanumériques As String
```

Informez l'ordinateur des termes que vous voulez utiliser et leurs registres permet d'économiser de la mémoire et vous aide à rédiger des instructions plus lisibles.

Les termes que vous définissez peuvent être des valeurs attribuées :

```
Voici_un_chiffre = 25
Voici_une_chaine_de_caractères_alphanumériques = "J'espère que mon
explication est claire".
```

Ici, le signe = est équivalent à « est ».

Vous pouvez également utiliser vos propres termes pour clarifier ce que vous dites et vous référer à certains objets de l'application.

Par exemple, si vous voulez utiliser un document situé sur `c:\mon ordinateur\mon dossier\mon sous-dossier\mon sous-sous-dossier\le fichier pour mon exemple.docx`, vous ne voulez pas avoir à écrire le chemin entier à chaque fois que vous allez en parler. Vous pouvez donc écrire :

```
Dim MonDoc As Document
Set MonDoc = Documents.Open ("c:\mon ordinateur\mon dossier\mon sous-
dossier\mon sous-sous-dossier\le fichier pour mon exemple.docx")
```

Ensuite, vous devrez mentionner « `MonDoc` » et l'ordinateur saura à quoi vous faites référence. Vous remarquerez que cette fois il est nécessaire de commencer la phrase par le mot clé « `Set` ». « `Set` » doit être utilisé pour faire référence aux objets de l'application.

7. Où écrivez-vous vos instructions ?

Dans le VBE, selon le type d'instructions, vous écrivez soit sur la page de code du document, soit dans un dossier séparé.

Si vous voulez écrire sur la page du document, dans la fenêtre **projet**, sélectionnez le document et double-cliquez dessus ou appuyez sur **F7**. La grande fenêtre grise sur la droite où vous écrivez devient blanche et le haut de la fenêtre ressemblera à cela :

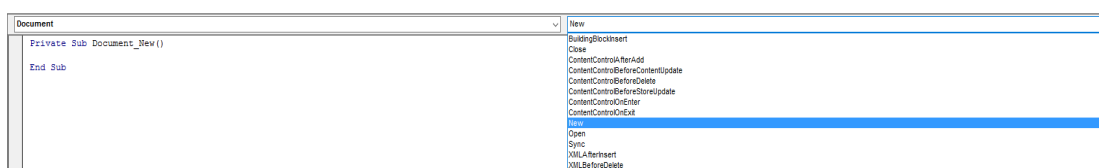


Cette section vous permet d'associer vos instructions aux actions (événements) qui prennent place dans votre document.

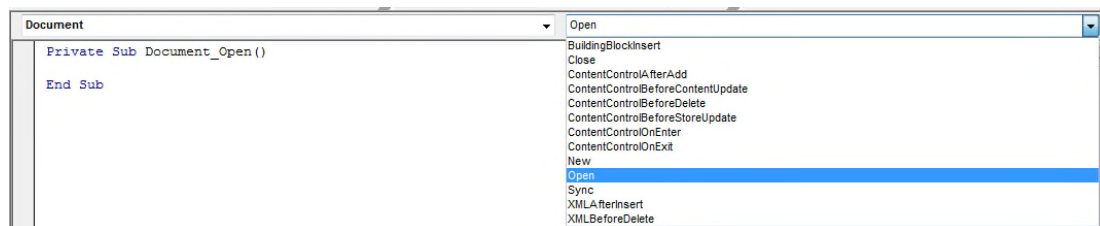
Par exemple, si vous utilisez le bouton déroulant à droite de **(Général)**, vous verrez **Document**. Sélectionnez « **Document** » pour changer immédiatement la partie droite de **(Déclarations)** à « **New** ». Et vous noterez l'apparition de texte juste au-dessous.



Si vous regardez maintenant le menu déroulant, vous disposez d'une liste d'actions/événements que vous pouvez utiliser pour exécuter vos instructions.



Par exemple, vous souhaitez recevoir un email dès que votre document est ouvert vous pouvez sélectionner :



et écrire vos instructions entre les deux lignes automatiquement générées.

Vous écrirez le plus souvent vos instructions dans un dossier séparé appelé un « **Module** » ou dans un « **UserForm** » ou un « **Module de classe** ».

- Un **Module** est simplement un dossier contenant un code que vous pouvez appeler depuis votre application.

Depuis votre application (et non pas le VBE), vous pouvez appeler la liste des macros disponibles en appuyant sur **Alt-F8**.

- Un **UserForm** est simplement une interface visuelle sur laquelle vous placerez des boutons et des objets et spécifierez le code pour le comportement de chaque objet. La plupart des interfaces utilisateur dans le logiciel sont simplement des userforms.

Par exemple dans Word®, appeler les boîtes de dialogue format Police ou Paragraphe signifie que vous appelez un userform.

- Un **Module de classe** vous permet de créer votre propre classe à savoir un mot défini avec des propriétés et des méthodes spécifiques.

Par exemple, vous voulez créer votre propre application dans laquelle vous avez une classe appelée employés qui a sa propre méthode comme compter, ajouter, supprimer et ses propres propriétés comme nom, adresse, etc.

8. Règles grammaticales

Comme dans toutes les langues, des règles grammaticales strictes doivent être observées.

La première règle est que vous devez écrire vos instructions entre deux lignes, l'une marquant le début et l'autre indiquant la fin. Dans notre capture d'écran précédente, l'ordinateur écrit automatiquement ces lignes :

```
Private Sub Document_Open()  
End Sub
```

Toutes les instructions que vous donnez doivent être contenues entre ces deux types de lignes.

Tout ce qui figure entre ces deux énoncés s'appelle une routine. Les routines peuvent être des **Sub** ou des **Functions**.

Un **Sub** peut être **Private** (Privé) ou **Public**. **Private** signifie que la routine ne peut être appelée que depuis le module où elle se trouve tandis que **Public** indique que cette routine peut être appelée depuis un autre module et qu'elle figurera dans la liste des macros disponibles de votre application.

Le nom d'un **Sub** ou d'une **Function** doit être unique et ne peut pas contenir d'espaces.

Une **Function** est faite pour renvoyer une valeur unique.

9. *Amusons-nous un peu*

Maintenant que vous connaissez les bases, amusons-nous un peu et écrivons notre première macro ensemble dans Word®.

Nous allons changer la couleur d'arrière-plan d'un document ouvert et modifier la couleur de chaque mot. Le premier mot aura la même taille de police et la couleur bleue, le second mot augmentera la taille de police de 2 points et sera blanc, et le troisième mot augmentera la taille de police de 1 point et sera rouge.

Comment faut-il faire ?

D'abord, nous ouvrons le VBE (**Alt-F11**)

Nous sélectionnons « **Normal** »

Puis, nous insérons un module : **Insertion>Module**

Dans la section vide à droite, nous écrivons notre code en commençant par donner un nom à notre action (nommer la routine).

Appelons-la « **Coloriage** »

```
Public Sub Coloriage()
```

Vous pouvez voir qu'en appuyant sur « Entrée », la ligne

```
End Sub
```

s'ajoute automatiquement et le curseur se place juste en dessous de la première ligne.

Il est inutile d'inclure les parenthèses après le nom de la routine. Le VBE le fait automatiquement. Nous aborderons la fonction de ces parenthèses dans un prochain numéro.

Avant d'écrire quoi que ce soit, essayons de nous organiser et de définir exactement ce que nous voulons que la macro exécute.

- Changer la couleur d'arrière-plan en mauve clair.
- Passer en revue chaque mot contenu dans le document.
 - o Si c'est un 1, la couleur est le bleu
 - o Si c'est un 2, la couleur est le blanc et la police est +2
 - o Si c'est un 3, la couleur est le rouge et la police est +1
- Ignorons toute police supérieure à 25 points.

Ensuite, quelles variables nous faut-il ?

- **MonDoc** pour le document d'application
- **CompteurMots** pour parcourir tous les mots contenus dans le document
- **MiniCompteur** pour compter de 1 à 3

Par conséquent, les premières lignes de notre code sont :

```
Dim MonDoc As Document
Dim CompteurMots As Long
Dim MiniCompteur As Integer
```

`Long` est la valeur numérique d'un nombre non décimal qui peut être extrêmement élevé, alors que `Integer`, qui représente également une catégorie de nombre non décimal, est limité. Nous ne connaissons pas le nombre de mots dans le document, mais s'il est trop grand et si nous utilisons `Integer` pour notre `CompteurMots`, ceci générera une erreur si la valeur dépasse la limite supérieure du nombre entier.

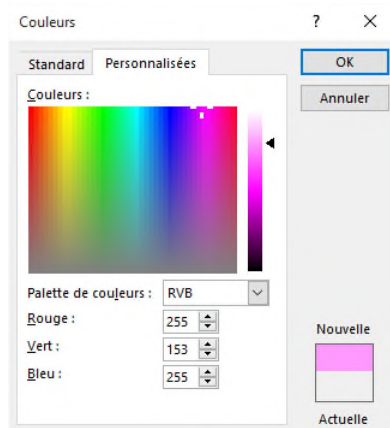
Nous devons donc définir certaines de nos valeurs

```
Set MonDoc = ActiveDocument
MiniCompteur = 1
```

Le référencement dans VBA est essentiel. Nous avons défini `MonDoc` comme `ActiveDocument` ou `ThisDocument`. Comme nous écrivons dans un module contenu dans le modèle `normal.dotm`, `ThisDocument` s'applique à `normal.dotm`, alors que `ActiveDocument` se réfère au document Word® actuellement actif utilisé pour appeler la procédure. Notre référence permet de modifier UNIQUEMENT le document sur lequel nous travaillons. Si nous avons sélectionné `ThisDocument`, les modifications s'appliqueraient à notre modèle `normal.dotm` et non à notre document de travail.

Nous devons donc changer la couleur de notre arrière-plan

- Remplir (`Fill`) la couleur (`ForeColor`) de l'arrière-plan (`Background`) de `MonDoc` par un mauve



```
MonDoc.Background.Fill.ForeColor = RGB(255, 153, 255)
```

Notez que pendant votre saisie, le VBE vous propose la liste des « requêtes » acceptables. Vous pouvez utiliser les flèches montantes et descendantes pour surligner votre choix et utiliser le bouton « **Tab** » (Tabulation) pour insérer l'élément choisi. Dans notre exemple, « `MonDoc` » n'était pas défini, l'interface ne pouvait rien proposer. À présent, l'interface sait que « `MonDoc` » est un « `Document` » et propose l'une des propriétés et des méthodes potentielles associées à un « `Document` ».

- Le rendre visible

```
MonDoc.Background.Fill.Visible = msoTrue
```

Pour ces 2 dernières lignes, nous constatons que les deux commencent par

```
MonDoc.Background.Fill
```

En tant que linguistes, nous n'aimons pas les répétitions et nous allons préciser ces deux lignes comme suit :

```
With MonDoc.Background.Fill
    .ForeColor = RGB(255, 153, 255)
    .Visible = msoTrue
End With
```

Notez que tous les éléments se référant à l'élément « With » (ici « MonDoc.Background.Fill ») commencent par un point, et que pendant la saisie, le VBE vous propose la liste des « requêtes » acceptables. Vous pouvez utiliser les flèches montantes et descendantes pour surligner votre choix et utiliser le bouton « Tab » (Tabulation) pour insérer l'élément choisi.

- Faire une boucle de tous les mots contenus dans le document

```
For CompteurMots = 1 To MonDoc.Words.Count
Next
```

C'est dans cette boucle For-Next que nous allons définir la couleur et la taille de la police à appliquer.

Une boucle For-Next commence à un nombre (ici 1) et augmente à chaque fois qu'elle atteint l'instruction Next [suivant] (ici incrément de 1 puisqu'aucun « Step » [Pas] n'est défini après le nombre de fin) jusqu'à ce qu'elle atteigne le dernier nombre (ici le nombre de mots comptés dans le document).

Pour notre exercice, nous voulons que notre MiniCompteur soit de 1 à 3, donc avant Next, nous devons :

- Ajouter 1 au MiniCompteur
- Vérifier que la valeur du MiniCompteur ne dépasse pas le 3, sinon changer la valeur du MiniCompteur à 1

```
MiniCompteur = MiniCompteur + 1
If MiniCompteur > 3 Then MiniCompteur = 1
```

Avant cela (parce que notre MiniCompteur commence à 1), nous devons choisir des actions en fonction de la valeur de notre MiniCompteur. Nous allons utiliser la fonction Select Case.

```
Select Case MiniCompteur
Case 1
Case 2
Case 3
End Select
```

Pour chaque Case (Cas), nous examinerons le numéro du mot, le nombre donné par notre CompteurMots nous chercherons la couleur de la police et sa taille.

```
MonDoc.Words(CompteurMots).Font.ColorIndex
```

Et

```
MonDoc.Words(CompteurMots).Font.Size
```

Nous devons aussi préciser les conditions pour augmenter ou diminuer la taille de la police. À cet effet, nous utiliserons la structure If-Then.

```
If MonDoc.Words(CompteurMots).Font.Size < 25 then
End if
```

Notre code final ressemblera à cela :

```
Public Sub Coloriage()
Dim MonDoc As Document
Dim CompteurMots As Long
Dim MiniCompteur As Integer

Set MonDoc = ActiveDocument

MiniCompteur = 1
```

```

With MonDoc.Background.Fill
    .ForeColor = RGB(255, 153, 255)
    .Visible = msoTrue
End With

For CompteurMots = 1 To MonDoc.Words.Count
    If MonDoc.Words(CompteurMots).Police.Taille < 25 Then
        Select Case MiniCompteur
            Case 1
                MonDoc.Words(CompteurMots).Font.ColorIndex = wdDarkBlue
            Case 2
                MonDoc.Words(CompteurMots).Font.ColorIndex = wdWhite
                MonDoc.Words(CompteurMots).Font.Size =
MonDoc.Words(CompteurMots).Font.Size + 2
            Case 3
                MonDoc.Words(CompteurMots).Font.ColorIndex = wdRed
                MonDoc.Words(CompteurMots).Font.Size =
MonDoc.Words(CompteurMots).Font.Size + 1
            End Select
            MiniCompteur = MiniCounter + 1
            If MiniCompteur > 3 Then MiniCounter = 1
        End If
    Next
End Sub

```

Notez que toutes les lignes qui semblent être sur plusieurs lignes telles que :

```

MonDoc.Words(CompteurMots).Font.Size = MonDoc.Words(CompteurMots).Font.Size
+ 2

```

ne le sont pas.

De plus, l'indentation utilisée n'a d'autre finalité que de faciliter la lecture du code.

La norme est d'ajouter un caractère de soulignement (.) à la fin de la ligne pour montrer que la ligne continue. Autrefois, les lignes étaient numérotées. Dans notre contexte, si vous avez le moindre doute, copiez et collez la ligne dans le VBE et si la ligne ne contient pas de retour à la ligne, elle n'occupera qu'une seule ligne.

Vous pouvez facilement remarquer ces lignes, car la deuxième ligne et les lignes suivantes de ces paragraphes ne sont pas mises en retrait.

De plus, le retrait n'a pas d'autre utilité que de simplifier la lecture du code.

Vous pouvez voir que notre code comporte beaucoup de `MonDoc.Words(CompteurMots).Font`, nous pourrions le simplifier de cette façon :

```

Public Sub Coloriage()

    Dim MonDoc As Document
    Dim CompteurMots As Long
    Dim MiniCompteur As Integer

    Set MonDoc = ActiveDocument

    MiniCompteur = 1

    With MonDoc.Background.Fill
        .ForeColor = RGB(255, 153, 255)
        .Visible = msoTrue
    End With

    For CompteurMots = 1 To MonDoc.Words.Count
        With MonDoc.Words(CompteurMots).Font
            If .Size < 25 Then
                Select Case MiniCompteur
                    Case 1

```



```

        .ColorIndex = wdDarkBlue
    Case 2
        .ColorIndex = wdWhite
        .Size = .Size + 2
    Case 3
        .ColorIndex = wdRed
        .Size = .Size + 1
    End Select
    MiniCompteur = MiniCounter + 1
    If MiniCompteur > 3 Then MiniCounter = 1
End If
End With
Next
End Sub

```

Ce code nous est aujourd'hui on ne peut plus clair, mais il ne l'est peut-être pas pour une autre personne ou ne le sera plus pour nous dans quelques semaines ou mois. Par conséquent, il est vivement conseillé d'ajouter des commentaires dans votre code pour vous aider à mémoriser vos intentions. Vous pouvez ajouter un commentaire sur n'importe quelle ligne, qu'elle soit vide ou à la fin, en commençant par une apostrophe. Voici notre routine finalisée :

```

'*****
' Routine pour colorer un texte en 3 couleurs différentes *
'*****
Public Sub Coloriage()

    'Ceci définit MonDoc dans un document Word@
    Dim MonDoc As Document
    'Ceci définit le CompteurMots en tant que grand nombre naturel
    Dim CompteurMots As Long
    'MiniCompteur est un petit nombre utilisé pour compter jusqu'à 3
    Dim MiniCompteur As Integer

    'MonDoc est défini comme le document utilisé pour appeler la macro
    Set MonDoc = ActiveDocument

    'Ceci lance le MiniCompteur
    MiniCompteur = 1

    'Changer la couleur d'arrière-plan du document
    With MonDoc.Background.Fill
        .ForeColor = RGB(255, 153, 255) 'mauve clair
        .Visible = msoTrue 'le rendre visible
    End With

    'Cette boucle passe en revue tous les mots contenus dans le document
    For CompteurMots = 1 To MonDoc.Words.Count
        With MonDoc.Words(CompteurMots).Font
            'Ceci se produit si la taille de la police est inférieure à 25
points
            If .Size < 25 Then
                Select Case MiniCompteur
                    Case 1 'si le MiniCompteur égal 1
                        'change la couleur en bleu foncé
                        .ColorIndex = wdDarkBlue
                    Case 2 'si le MiniCompteur égal 2
                        'change la couleur en blanc
                        .ColorIndex = wdWhite
                        'augmente la taille de la police de 2 points
                        .Size = .Size + 2
                    Case 3 'si le MiniCompteur égal 3
                        'change la couleur en rouge
                        .ColorIndex = wdRed
                        'augmente la taille de la police de 1 point
                        .Size = .Size + 1
                End Select
                'Incrémentation du MiniCompteur
                MiniCompteur = MiniCounter + 1
                'Vérifie si le MiniCompteur est supérieur à 3
                'Si c'est le cas, réinitialise le MiniCompteur à 1
                If MiniCompteur > 3 Then MiniCounter = 1
            End If
        End With
    Next
End Sub

```

```
End If
End With
Next
End Sub
```

À présent, vous pouvez ouvrir l'un de vos documents dans Word® et depuis ce document, appuyez sur **Alt-F8**, sélectionnez **Coloriage** et observez le résultat qui s'applique à ce document spécifique.

10. Ce n'est pas ce à quoi je m'attendais !

Vous remarquerez que ce n'est pas exactement ce à quoi vous vous attendiez. La raison en est simple. La signification de « **words** » (mots) dans le VBA n'est pas exactement la même que pour nous linguistes. Dans le VBA pour Word®, un WORD est un mot tel que les linguistes le comprennent, suivi d'une espace s'il n'y a pas de ponctuation ou de marque de mise en forme, mais chaque marque de ponctuation est également considérée comme un mot, comme le trait d'union, les marques de paragraphe, etc. Vous pouvez aussi remarquer que la numérotation automatique prend la couleur de la marque du précédent paragraphe. Le meilleur moyen de s'en rendre compte est d'afficher les marques de mise en forme dans votre texte.

Si le document que vous testez contient des en-têtes et des pieds de page ou des zones de texte, vous remarquerez que la macro ne les a pas modifiés. Ce sont également des mots, non ? Oui en effet, mais ces éléments ne se trouvent pas dans la même ligne de « l'histoire ». Dans **OH MY WORD !**, il a été fait mention de couches. Chaque couche contient une histoire différente. Comme nous n'avons pas indiqué à notre macro de parcourir toutes les histoires, elle a seulement pris en compte l'histoire par défaut (le texte principal).

Par ailleurs, pourquoi avoir limité la taille de la police à 25 ? Le but consistait à empêcher tout message d'erreur pour les utilisateurs qui voudraient tester la macro sur des fichiers comportant des tableaux. Chaque cellule du tableau dans Word® contient un caractère « fin de cellule » qui n'a pas de taille de police (pas vraiment exact, car la taille de la police est 9999 et ne peut pas être incrémentée et il s'agit seulement d'un moyen de ne pas attribuer une taille de police à des éléments) et que le VBA pour Word® considère comme un mot. Sans cela (et de manière entièrement arbitraire puisque nous aurions pu décider de définir la limite supérieure de police à 9997) notre macro aurait rencontré un obstacle impossible : augmenter la taille de la police au-dessus de la limite maximale définie. En d'autres termes, une erreur avec l'épouvantable fenêtre d'erreur et en général un numéro d'erreur obscure (qui est souvent le moment auquel une personne saine d'esprit clique sur la touche « Fin » et s'en va) vous demandant quoi faire.

Ce type de macro inutile est en réalité très utile pour saisir le sens exact des termes utilisés dans le VBA et faire faire à vos macros précisément ce que vous voulez qu'elles exécutent.

Dans notre exemple, nous pouvons la réécrire pour ajouter des exceptions, par exemple :

- Si une marque de ponctuation suit le mot, alors appliquer les mêmes changements et augmenter en conséquence la valeur de notre **CompteurMots**.
- Si le mot est une parenthèse ouverte, alors inclure le mot suivant de sorte qu'il intègre les mêmes changements, et vérifier qu'il est suivi d'une parenthèse fermée.
- Etc.

11. Conclusion

Le VBA forme un langage puissant que nous pouvons utiliser pour indiquer à l'ordinateur d'exécuter des ensembles d'instructions variées afin de ne pas avoir à les réaliser manuellement.

Le VBA étant un langage, nous devons respecter ses règles grammaticales et comprendre parfaitement la définition précise de chaque mot. Ceci devrait être assez facile pour nous linguistes.

Le langage informatique du VBA nous conduit également à être extrêmement précis dans notre façon d'écrire nos instructions pour éviter les erreurs, les pièges et les comportements inattendus. Nous devons garantir la logique de nos instructions. En tant que linguistes, ceci peut se révéler légèrement plus compliqué.

Dans notre exemple de code, nous avons décidé de l'écrire d'une façon, mais nous aurions pu opter pour bien d'autres façons. Par exemple, au lieu d'utiliser une structure `Select Case` pour notre `MiniCompteur`, nous aurions pu utiliser la construction `If-Then`. Nous aurions pu créer des routines distinctes pour chaque action de modification de la taille de police et de la couleur et les appeler depuis notre principale routine. Le VBA est un langage et vous pouvez profiter d'un peu de latitude au sein de sa structure. Certaines structures de codage sont meilleures que d'autres (exécution plus rapide, moins de mémoire utilisée). De mon point de vue, dans la mesure où le groupe d'instructions ne contient aucune erreur, l'exécution est toujours plus rapide que la saisie manuelle et le temps passé à l'écriture de la macro est rentable (en d'autres termes, si le temps passé à développer les instructions est compensé par le temps que la macro épargnera), alors l'automatisation de la tâche en vaut la peine.

12. *Un peu plus utile*

Comme nous l'avons vu dans la première partie de la newsletter, notre travail implique souvent de modifier les options en fonction du travail en cours. Cette tâche peut être fastidieuse, répétitive et vous pouvez parfois oublier d'activer ou de désactiver une option. Il est donc conseillé d'écrire quelques profils dans lesquels vous définissez exactement l'état de vos options, ce qui vous permet ensuite de les appeler en quelques clics.

Voici la liste des options couvertes dans ce numéro et comment elles sont appelées dans le VBA :

Fichier>Options	Nom complet du VBA	État
Onglet Affichage section Toujours afficher ces marques de mise en forme à l'écran		
Tabulations	ActiveWindow.View.ShowTabs	
Espaces	ActiveWindow.View.ShowSpaces	
Marques de paragraphe	ActiveWindow.View.ShowParagraphs	
Texte masqué	ActiveWindow.View.ShowHiddenText	True ou False
Traits d'union conditionnels	ActiveWindow.View.ShowHyphens	True ou False
Ancrage d'objets	ActiveWindow.View.ShowObjectAnchors	
Sauts conditionnels	ActiveWindow.View.ShowOptionalBreaks	
Afficher toutes les marques de mise en forme	ActiveWindow.View.ShowAll	
Onglet Options avancées section Options de mise en page pour		
Ne pas étirer les espaces des lignes finissant par Mj+Retour	ActiveDocument.Compatibility(wdExpandShiftReturn)	True ou False
Onglet Options avancées section Affichage du contenu des documents		
Afficher les couleurs et images d'arrière-plan en mode Page	ActiveWindow.View.DisplayBackgrounds	
Afficher le texte ajusté à la fenêtre du document	ActiveWindow.View.WrapToWindow	
Afficher les espaces pour images	ActiveWindow.View.ShowPicturePlaceHolders	True ou False
Afficher les dessins et les zones de texte à l'écran	ActiveWindow.View.ShowDrawings	
Afficher les signets	ActiveWindow.View.ShowBookmarks	
Afficher les limites d'un texte	ActiveWindow.View.ShowTextBoundaries	
Afficher les repères de rognage	ActiveWindow.View.FieldShading	wdFieldShadingNever wdFieldShadingWhenSelected ou wdFieldShadingAlways
Afficher les codes de champs au lieu de leurs valeurs	ActiveWindow.View.ShowFieldCodes	True ou False
Onglet Options avancées section Afficher		
Afficher la règle verticale en mode Page	ActiveWindow.DisplayVerticalRuler	True ou False
Afficher les mesures en :	Options.MeasurementUnit	wdInches wdCentimeters wdMillimeters wdPicas ou wdPoints
Afficher les mesures en largeur de caractères	Options.UseCharacterUnit	True ou False

Ruban onglet	Nom complet du VBA	État
Ruban onglet Affichage groupe Afficher		
Règle	ActiveWindow.ActivePane.DisplayRulers	True ou False
Quadrillages	Options.DisplayGridLines	True ou False
Volet de navigation	ActiveWindow.DocumentMap	
Ruban onglet	Nom complet du VBA	État
Ruban onglet Mise en page groupe Organiser Option		
Aligner sélection Paramètres de la grille...		
Afficher les repères d'alignement	Options.DisplayAlignmentGuides	
Repères de page	Options.PageAlignmentGuides	
Repères de marge	Options.MarginAlignmentGuides	True ou False
Repères de paragraphe	Options.ParagraphAlignmentGuides	True ou False
Alignement d'objets		
Aligner les objets sur les autres objets	ActiveDocument.SnapToShapes	
Paramètres de la grille		
Espacement horizontal	ActiveDocument.GridDistanceHorizontal	MillimetersToPoints(VALEUR) CentimetersToPoints(VALEUR) InchesToPoints(VALEUR) PicasToPoints(VALEUR) ou VALEUR [l'unité par défaut est le point]
Espacement vertical	ActiveDocument.GridDistanceVertical	MillimetersToPoints(VALEUR) CentimetersToPoints(VALEUR) InchesToPoints(VALEUR) PicasToPoints(VALEUR) ou VALEUR [l'unité par défaut est le point]
Origine de la grille		
Utiliser les marges	ActiveDocument.GridOriginFromMargin	True ou False
Point d'origine horizontal	ActiveDocument.GridOriginHorizontal	MillimetersToPoints(VALEUR) CentimetersToPoints(VALEUR) InchesToPoints(VALEUR) PicasToPoints(VALEUR) ou VALEUR [l'unité par défaut est le point]
Point d'origine vertical	ActiveDocument.GridOriginVertical	MillimetersToPoints(VALEUR) CentimetersToPoints(VALEUR) InchesToPoints(VALEUR) PicasToPoints(VALEUR) ou VALEUR [l'unité par défaut est le point]
Afficher la grille		
Afficher la grille à l'écran		
Afficher 1 ligne verticale sur (cocher l'option)	ActiveDocument.GridSpaceBetweenHorizontalLines	0
Afficher 1 ligne verticale sur	ActiveDocument.GridSpaceBetweenHorizontalLines	VALEUR [entier]
Afficher 1 ligne horizontale sur	ActiveDocument.GridSpaceBetweenVerticalLines	
Aligner les objets sur la grille lorsque le quadrillage n'est pas affiché	ActiveDocument.SnapToGrid	True ou False

Maintenant, vous pouvez facilement créer une macro pour définir des profils de travail spécifiques. Voici un exemple.

Ce que je veux

Comment demander à l'ordinateur d'exécuter une macro avec VBA

```
Public Sub Profil_EditerTexte()
```

Je veux voir

- Marques de tabulation
- Espaces
- Marques de paragraphe
- Texte masqué
- Ancrage d'objets
- Signets
- Contenu de champ (et non des codes)

```
With ActiveWindow.View
    .ShowTabs = True
    .ShowSpaces = True
    .ShowParagraphs = True
    .ShowHiddenText = True
    .ShowObjectAnchors = True
    .ShowBookmarks = True
    .ShowFieldCodes = False
```

Mais pas

- Sauts conditionnels

```
.ShowOptionalBreaks = False
End with
```

Je veux voir aussi

- La règle verticale
- La ou les règles
- L'explorateur de document

```
With ActiveWindow
    .DisplayVerticalRuler = True
    .ActivePane.DisplayRulers = True
    .DocumentMap = True
End with
```

Et je veux mes mesures en pouces

```
Options.MeasurementUnit = wdInches
End sub
```

Maintenant, écrivez ce code dans un module (un nouveau module ou après la dernière routine), appuyez sur **Alt-F8** à partir d'un document et sélectionnez `Profile_TextEditing` pour activer immédiatement toutes ces fonctions.

Nous espérons que vous avez apprécié ce numéro et qu'il vous aura appris des informations utiles. Nous souhaitons également vous revoir au prochain numéro dans lequel nous aborderons Outlook® et comment écrire une macro qui énumère tous les fichiers que vous pouvez joindre à votre email.

L'Équipe Out of Office



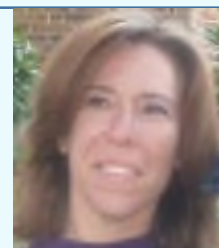
Hacène Dramchini
MCIL, MNWTN, MIAPTI
Auteur principal
Traducteur technique anglais > français passionné de TI,
Hacène est en charge du contenu de cette newsletter.

Lucy Brooks, CL, FCIL, MITI
Rédacteur anglais
Traductrice compétente français, allemand et espagnol et
Directrice/Fondatrice d'[eCPD webinars](#), Lucy a une passion pour
le partage des connaissances.



Angelika Körber, MATICOM, MCBTIP, MCiOLGS,
traductrice assermentée
Localisateur allemand
Traductrice technique anglais et français vers l'allemand, et
journaliste, Geli a un esprit curieux et vise toujours la
perfection.

Muriel Louchart, Traductrice assermentée
Localisateur français
Traductrice compétente du domaine juridique et de la
localisation de logiciels anglais > français, Muriel assure la
localisation de ce périodique mensuel pour le français
européen.



Patricia Torres, MCONALTI, MATA et traductrice
assermentée
Localisateur espagnol
Traductrice anglais et italien > espagnol et professeur
universitaire en traductologie, Patricia adore la technologie.

CLAUSE DE NON-RESPONSABILITE:

L'équipe décline expressément toute responsabilité, toute perte ou tout risque lié à une conséquence directe ou indirecte de l'utilisation du contenu de cette newsletter. Cette newsletter est fournie « telle quelle » sans déclarations ou garanties, explicites ou implicites, y compris mais sans s'y limiter, les garanties implicites de commerciabilité, d'adaptation à une fin particulière et de non-violation. L'utilisation du contenu de cette newsletter implique que vous renoncez à tous droits ou à toutes réclamations contre l'auteur ou contre l'un des éditeurs de la newsletter. L'information contenue dans cette newsletter est fournie uniquement à titre d'information générale et peut ne pas répondre à toutes les questions pertinentes d'ordre commercial ou juridique, en conséquence aucune promesse ou garantie n'est faite concernant l'exactitude ou l'exhaustivité des informations.

DROITS D'AUTEUR & MARQUES COMMERCIALES

Out Of Office est une publication indépendante et n'est pas affiliée, ni n'a été autorisée, parrainée ou autrement approuvée par Microsoft Corporation. Microsoft, MS Office et tous les logiciels de la suite MS Office sont soit des marques déposées ou des marques de Microsoft Corporation aux États-Unis et/ou dans d'autres pays.

NOUS CONTACTER

outoffice.newsletter@gmail.com