

Editions ENI

Excel 2016

Collection
Référence Bureautique

Extrait

Effectuer des calculs sur des données de type date

Après avoir abordé le principe de calcul sur les dates utilisé par Excel, nous vous proposons de découvrir quelques fonctions spécifiques au traitement des dates à travers quelques exemples.

Principe de calcul sur des jours

- ☒ Si le calcul porte sur des jours, procédez comme pour les autres calculs car Excel enregistre les dates sous la forme de nombres séquentiels appelés numéros de série. De ce fait, elles peuvent être ajoutées, soustraites et incluses dans d'autres calculs.
- ☒ Par défaut, sous Windows, Excel utilise le calendrier depuis 1900 (Excel pour Macintosh, le calendrier depuis 1904). Le 1er janvier 1900 correspond donc (pour Excel sous Windows) au numéro de série 1 et le 1er janvier 2005 correspond au nombre 38 353, car 38 353 jours se sont écoulés depuis le 1er janvier 1900.
- ☒ Pour utiliser une fonction spécifique de gestion de dates et d'heures, vous pouvez activer l'onglet **Formules** et cliquer sur le bouton **DateHeure** du groupe **Bibliothèque de fonctions** puis sur la fonction concernée afin d'utiliser l'Assistant.

ANNEE(numéro_de_série)

Convertir un numéro de série en année.

AUJOURDHUI()

Renvoyer le numéro de série de la date du jour.

DATE(année;mois;jour)

Renvoyer le numéro de série d'une date précise.

DATEVAL(date_texte)

Convertir une date représentée sous forme de texte en numéro de série.

FIN.MOIS(date_départ;mois)

Renvoyer le numéro séquentiel de la date du 1er jour du mois précédant ou suivant la date_départ du nombre de mois indiqué.

FRACTION.ANNEE(date_début;date_fin;[base])

Renvoyer la fraction de l'année représentant le nombre de jours entre la date de début et la date de fin.

HEURE(numéro_de_série)

Convertir un numéro de série en heure.

JOUR(numéro_de_série)

Convertir un numéro de série en jour du mois.

Les calculs

JOURS(date_fin;date_début)

Calculer le nombre de jours entre les deux dates.

JOURS360(date_début;date_fin;[méthode])

Calculer le nombre de jours séparant deux dates sur la base d'une année de 360 jours.

JOURSEM(numéro_de_série;[type_retour])

Convertir un numéro de série en jour de la semaine.

MAINTENANT()

Renvoyer le numéro de série de la date et de l'heure du jour.

MINUTE(numéro_de_série)

Convertir un numéro de série en minute.

MOIS(numéro_de_série)

Convertir un numéro de série en mois.

MOIS.DECALER(date_départ;mois)

Renvoyer le numéro séquentiel de la date qui représente une date spécifiée (l'argument date_départ), corrigée en plus ou en moins du nombre de mois indiqué.

NB.JOURS.OUVRES(date_début;date_fin;[jours_fériés])

Renvoyer le nombre de jours ouvrés entiers compris entre deux dates.

NB.JOURS.OUVRES.INTL(date_début;date_fin;[weekend];[jours_fériés])

Renvoyer le nombre de jours ouvrés entiers compris entre deux dates à l'aide de paramètres identifiant les jours du week-end et leur nombre.

NO.SEMAINES(numéro_de_série;[type_retour])

Convertir un numéro de série en numéro de semaine dans l'année.

NO.SEMAINES.ISO(date)

Renvoyer le numéro ISO de la semaine de l'année correspondant à une date donnée.

SECONDE(numéro_de_série)

Convertir un numéro de série en seconde.

SERIE.JOUR.OUVRE(date_début;nb_jours;[jours_fériés])

Renvoyer le numéro de série de la date avant ou après le nombre de jours ouvrés spécifiés.

SERIE.JOUR.OUVRE.INTL(date_départ;nb_jours; [nb_jours_week-end];[jours_fériés])

Renvoyer le numéro de série de la date avant et après un nombre spécifié de jours ouvrés en spécifiant des paramètres qui identifient et dénombrent les jours inclus dans le week-end.

Calculs avancés

TEMPS(heure;minute;seconde)

Renvoyer le numéro de série d'une heure précise.

TEMPSVAL(heure_texte)

Convertir une heure représentée sous forme de texte en numéro de série.

Combiner du texte avec une date

- ☐ Pour combiner dans une cellule le texte et la date contenus dans différentes cellules, vous pouvez utiliser la fonction **TEXTE** dont la syntaxe est :

=TEXTE(valeur;format_texte)

L'argument **valeur** représente une valeur numérique ou une formule dont le résultat est une valeur numérique ou bien encore une référence à une cellule contenant une valeur numérique.

L'argument **format_texte** représente un format de nombre sous forme de texte défini dans la zone **Catégorie** de la boîte de dialogue **Format de cellule**.

Voici un exemple d'utilisation :

| | A | B | C | D | E |
|---|---|---------------------|--------------------------|------------|--|
| 1 | | Noms | Date de naissance | Age | Jour et mois de naissance |
| 2 | | Edmond BOSAPIN | 21/04/1978 | 37 ans | =\"né(e) le \"&TEXTE(C2;\"jj mmmm\")\" |
| 3 | | Juste COMILFO | 17/05/1968 | 47 ans | né(e) le 17 mai |
| 4 | | Honoré DE VOUVOIR | 15/06/1991 | 24 ans | né(e) le 15 juin |
| 5 | | Aude JAVEL | 12/05/1956 | 59 ans | né(e) le 12 mai |
| 6 | | Dimitri LE COURRIER | 03/07/1987 | 28 ans | né(e) le 03 juillet |
| 7 | | Edith MOITOU | 12/04/1970 | 45 ans | né(e) le 12 avril |

Calculer la différence entre deux dates (fonction DATEDIF)

DATEDIF est l'une des fonctions "masquées" de l'application Excel ; elle n'apparaît pas dans l'Assistant fonction ni dans l'aide en ligne. Les fonctions masquées ont été introduites dans Excel pour des raisons de compatibilité avec d'autres tableurs, elles fonctionnent parfaitement mais ne font pas partie des fonctions "officielles" d'Excel.

Cette fonction s'avère très pratique dans le cas, par exemple, où vous souhaitez calculer l'ancienneté d'un employé en années et en mois. La syntaxe de la fonction **DATEDIF** est **DATEDIF(Date_début;Date_fin;Type)**.

Les calculs

L'argument **Type** représente la durée calculée et peut prendre les valeurs suivantes :

- "y" pour calculer la différence absolue en années.
- "M" pour calculer la différence absolue en mois.
- "d" pour calculer la différence absolue en jour.
- "ym" pour calculer la différence en mois si les deux dates sont dans la même année.
- "yd" pour calculer la différence en jours si les deux dates sont dans la même année.
- "md" pour calculer la différence en jours si les deux dates sont dans le même mois.

Voici un exemple d'utilisation :

| | A | B | C |
|---|---|------------------------|---------------------|
| 1 | | Date de début | 30/11/2013 |
| 2 | | Date de fin | 01/12/2015 |
| 4 | | Nombre de jours | =DATEDIF(C1;C2;"d") |
| 5 | | Nombre de mois | =DATEDIF(C1;C2;"m") |
| 6 | | Nombre d'années | =DATEDIF(C1;C2;"y") |
| 7 | | | |

Voici un autre exemple qui permet de calculer l'âge d'une personne en fonction de la date du jour (fonction=**Aujourd'hui()**) :

| | A | B | C | D | E |
|---|---|--------------------------|-------------------------|--|---|
| 1 | | Date de naissance | 25/02/1964 | FORMULES | |
| 3 | | Age en années | 51 ans | =DATEDIF(C1;AUJOURDHUI();"y")&" ans" | |
| 4 | | Nombre de mois | 51 ans et 4 mois | =DATEDIF(C1;AUJOURDHUI();"y")&" ans et "&DATEDIF(C1;AUJOURDHUI();"ym")&" mois" | |
| 5 | | | | | |

Calculer le nombre de jours ouvrés ou non entre deux dates

Excel sait calculer le nombre de jours ouvrés (du lundi au vendredi) qui séparent deux dates à l'aide de la fonction **NB.JOURS.OUVRES** dont la syntaxe est :
=NB.JOURS.OUVRES(jour_début;jour_fin)

Voici un exemple d'utilisation :

| | A | B | C |
|---|-------------------------------|---|--------------------------------|
| 1 | Date de début | | 30/11/2013 |
| 2 | Date de fin | | 01/12/2015 |
| 4 | Nombre de jours ouvrés | | 522 |
| 5 | Formule de calcul | | =NB.JOURS.OUVRES(C1;C2) |
| 6 | | | |

Pour que cette fonction puisse tenir compte des jours fériés, vous devez y ajouter un troisième argument faisant référence à un jour férié ou à une plage de dates fériées. La syntaxe de cette fonction est alors :
=NB.JOURS.OUVRES(jour_début;jour_fin;jours_fériés)

Sur cet exemple, les jours fériés ont été calculés dans la plage de cellules B3 à B15.

| Année | | 2015 | | Nombre de jours chômés | | | | 9 | |
|---|-----------------|----------|------------------|------------------------|-----------|------------|------------------------|-------------------|-----------|
| | | | | <i>(hors week-end)</i> | | | | | |
| Jours fériés | | | | Mois | Début | Fin | Nombre de jours ouvrés | Nbre de dimanches | |
| 1 | Premier de l'an | jeudi | 01 janvier 2015 | 1 | janvier | 01/01/2015 | 31/01/2015 | 21 | 4 |
| 2 | Pâques | dimanche | 05 avril 2015 | 2 | février | 01/02/2015 | 28/02/2015 | 20 | 4 |
| 3 | Lundi de Pâques | lundi | 06 avril 2015 | 3 | mars | 01/03/2015 | 31/03/2015 | 22 | 5 |
| 4 | Fête du travail | vendredi | 01 mai 2015 | 4 | avril | 01/04/2015 | 30/04/2015 | 21 | 4 |
| 5 | Victoire 1945 | vendredi | 08 mai 2015 | 5 | mai | 01/05/2015 | 31/05/2015 | 17 | 5 |
| 6 | Ascension | jeudi | 14 mai 2015 | 6 | juin | 01/06/2015 | 30/06/2015 | 22 | 4 |
| 7 | Pentecôte | lundi | 25 mai 2015 | 7 | juillet | 01/07/2015 | 31/07/2015 | 22 | 4 |
| 8 | Fête nat. | mardi | 14 juillet 2015 | 8 | août | 01/08/2015 | 31/08/2015 | 21 | 5 |
| 9 | Assomption | samedi | 15 août 2015 | 9 | septembre | 01/09/2015 | 30/09/2015 | 22 | 4 |
| 10 | Toussaint | dimanche | 01 novembre 2015 | 10 | octobre | 01/10/2015 | 31/10/2015 | 22 | 4 |
| 11 | Armistice | mercredi | 11 novembre 2015 | 11 | novembre | 01/11/2015 | 30/11/2015 | 20 | 5 |
| 12 | Noël | vendredi | 25 décembre 2015 | 12 | décembre | 01/12/2015 | 31/12/2015 | 22 | 4 |
| L'année 2015 n'est pas une année BISSEXTILE | | | | | | | Total | 252 | 52 |

✎ Pour calculer le nombre de jours entre deux dates (jours fériés, chômés... inclus), vous pouvez utiliser la fonction **JOURS** dont la syntaxe est **JOURS(date_fin;date_début)**.

Les calculs

Calculer la date située après un nombre de jours ouvrés donné

La fonction **SERIE.JOUR.OUVRE** vous permet de calculer une date correspondant à une date (date de début) plus ou moins le nombre de jours ouvrés spécifié. Les jours ouvrés excluent les samedi et dimanche ainsi que toutes les dates identifiées comme étant des jours chômés.

La syntaxe de cette fonction est la suivante :

=SERIE.JOUR.OUVRE(date_début;nb_jours;jours_fériés) :

date_début Représente la date de début.

nb_jours Représente le nombre de jours ouvrés avant ou après la date de début. Un nombre de jours positif donne une date future, à l'inverse, un nombre de jours négatif donne une date passée.

jours_fériés Représente une liste de dates à exclure du calendrier des jours de travail (jours fériés, congés, absence...). Cet argument est facultatif.

Voici un exemple d'utilisation : nous souhaitons trouver la date d'échéance d'un travail devant commencer le 01 mai 2015 et qui est d'une durée de 40 jours ouvrés.

| | A | B | C | D | E |
|---|---|--------------------------|---------------------------------------|---------------------|-------------------|
| 1 | | | | | |
| 2 | | Date de début | 01/05/2015 | Jours fériés | 01/05/2015 |
| 3 | | Nombre de jours | 40 | | 08/05/2015 |
| 4 | | Échéance | 01/07/2015 | | 14/05/2015 |
| 5 | | Formule de calcul | =SERIE.JOUR.OUVRE(C1;C2;E1:E4) | | 25/05/2015 |
| 6 | | | | | |

*Le format **Date** a été appliqué à la cellule C3 car, par défaut, Excel affiche le résultat sous forme de numéro de série.*

Si cette formule vous renvoie un message d'erreur, en voici sa signification :

#VALEUR! Un argument n'est pas une date valide.

#NOMBRE! La date de début plus le nombre de jours ne donne pas une date valide.

Editions ENI

Apprenez le langage VBA et devenez un expert sur Excel

Collection
Solutions Business

Extrait

A. Formulaire de gestion des ventes

1. Description de l'exemple

a. Présentation de l'exemple

Nous sommes l'entreprise **SacEni**, un distributeur qui commercialise des sacs de sports, un sac en taille L, un autre en taille XL. L'activité débute et l'entreprise souhaite se doter d'un simple fichier pour suivre ses ventes et son stock.

L'outil que vous allez mettre en place va permettre au vendeur de cette petite entreprise de créer, stocker et télécharger des factures.

L'outil va se présenter comme un formulaire à remplir par le vendeur. Il sera accessible à partir d'un fichier Excel et du bouton **Accéder à l'outil de gestion des ventes** qui sera positionné sur la feuille d'ouverture du classeur.

Voici l'outil une fois qu'il sera finalisé :

Logiciel de gestion des ventes

Quantité Produit

5 Sac taille XL

Ajouter à la ligne

4*Sac taille L=80€
5*Sac taille XL=150€

Supprimer la ou les lignes sélectionnées

Total HT 230 € Remise 10 % pro

Total après
remise 230 € TVA 46 €

Total TTC 276 €

Sauvegarder et imprimer la commande

b. Présentation du fichier

Pour réaliser cet exemple, vous allez utiliser le fichier Excel **Enoncé_3-A.xlsm**. Le format XLSM signifie que c'est un fichier Excel qui prend en charge les macros (permettant l'utilisation de VBA, souvent désactivée par défaut).

Ce fichier contient trois feuilles Excel (appelées *sheets* en Visual Basic) :

Feuille Accueil

Cette feuille est destinée à contenir uniquement le bouton **Accéder à l'outil de gestion des ventes**. Il permettra l'accès à l'outil de gestion des ventes.

Initialement cette feuille est vide.

Feuille Produits

La feuille **Produits** recense les produits vendus par la société. Elle contient trois colonnes :

| | Colonne A | Colonne B | Colonne C |
|---------|----------------|----------------|---------------------|
| Ligne 1 | Nom du produit | Prix hors taxe | Quantité disponible |
| Ligne 2 | Sac taille L | 20 € | 90 |
| Ligne 3 | Sac taille XL | 30 € | 50 |

Feuille Factures

Cette feuille contiendra les factures créées avec l'outil de gestion des ventes. Elles seront référencées par numéro et vous trouverez également la date/heure de l'édition, ainsi que le montant.

| | Colonne A | Colonne B | Colonne C |
|---------|-------------------|---------------|-----------------------|
| Ligne 1 | Numéro de facture | Date et heure | Montant de la facture |

c. Fonctionnalités

L'objectif est donc d'avoir un outil qui permet de gérer les ventes des sacs de l'entreprise. Voici la retranscription des besoins sous forme d'exigences.

Exigences métiers

Les exigences métiers correspondent à la description des fonctionnalités globales de l'application, c'est le niveau de détail le plus faible :

- ▶ Créer une facture ;
- ▶ Tracer la facture ;
- ▶ Mettre à jour les stocks de produits.

Retranscription de ces exigences métiers en fonctionnalités

Les fonctionnalités permettent de détailler les exigences métiers en fonctionnalités qui correspondent aux actions utilisateurs et aux traitements du système. Cette liste doit être exhaustive afin de permettre de réaliser ces fonctionnalités sous forme d'application.

- ▶ Ajouter une ligne de commande
 - ▶ Choisir la quantité ;
 - ▶ Choisir le produit ;
 - ▶ Valider la ligne de commande ;
- ▶ Afficher la commande.
- ▶ Supprimer une ligne de commande.

- ▶ Faire le total et permettre l'application d'une remise de 10 %.
- ▶ Mettre à jour les stocks.

2. Notions de cours

Cet exemple contient de nombreux nouveaux concepts liés à la programmation avec le langage Visual Basic... Quelques repères sont donc utiles pour pouvoir démarrer sereinement.

a. Concept de programmation

Voici une description simplifiée de quelques notions de base de programmation permettant une meilleure approche des exemples proposés.

Objet et classe

Un **objet** est une entité informatique, il peut être de toute forme et chaque objet est unique. Il est caractérisé selon son type.

La **classe** correspond à la définition de l'objet, elle servira de canevas pour la création de nouveaux objets. Par conséquent, tous les objets d'une même classe auront les mêmes propriétés, ils se différencieront par les valeurs de leurs propriétés.

Exemple :

| Classe | Objets |
|--|---|
| Cell (cellule d'une feuille Excel) | <ul style="list-style-type: none"> ▶ Cells("A1") : Cellule A1 de la feuille en cours ; ▶ Cells("C4") : Cellule C4 de la feuille en cours. |
| Sheet (Feuille de classeur) | <ul style="list-style-type: none"> ▶ Sheets(0) : première feuille du classeur en cours ; ▶ Sheets(1) : deuxième feuille du classeur en cours. |
| Textbox (zone de texte saisissable dans un formulaire) | <ul style="list-style-type: none"> ▶ Textbox1 : zone de texte saisissable nommée Textbox1 par l'utilisateur ▶ Textbox2 : zone de texte saisissable nommée Textbox2 par l'utilisateur. |

Propriétés

Une **propriété** correspond à un attribut d'une classe. Lorsqu'un objet est créé, il a donc des valeurs assignées à ses propriétés.

Exemple : La cellule d'une feuille comporte de nombreuses propriétés, comme par exemple la valeur : `Cells("A1").value`

Méthode

Une **méthode** correspond à une action qui peut être réalisée par un objet. Par exemple, l'objet feuille de calcul (`Sheets`) propose une méthode `Add` qui permet d'ajouter une feuille.

Exemple : `Sheets.Add`

Collections

Une **collection** est une liste d'objets d'une même classe. Par exemple, la collection `Sheets` correspond à l'ensemble des feuilles. En Visual Basic, les collections sont des objets à part entière avec leurs propres méthodes et propriétés.

Variables

Une **variable** est une entité informatique qui permet de stocker des informations au sein de l'application, elle se déclare de la manière suivante :

- ▶ `Dim` : permet de définir la variable (`Public` pour une variable publique) ;
- ▶ `Nom_variable` : permet de donner un nom à la variable ;
- ▶ `As TypeVariable` : permet de typer la variable.

Exemple :

```
Dim MaVariable As String
```

Cela signifie que la variable `MaVariable` est déclarée en tant que chaîne de caractères.

Les variables sont :

- ▶ **Publiques** : elles sont accessibles sur l'ensemble de l'application. Elles sont déclarées en dehors de toute procédure de code ;
- ▶ **Privées** : elles sont accessibles uniquement dans la procédure où elles sont déclarées (sur une procédure donnée).

Les variables sont typées principalement pour les trois motifs suivants :

- ▶ Cela permet d'avoir des méthodes (voir précédemment) adaptées à la variable : une addition de chaînes de caractères correspond à la concaténation alors qu'une addition de nombres correspond à la somme des valeurs :

| Opérations | Valeur de la variable chaîne de caractères | Valeur de la variable nombre entier |
|--------------------------------|--|-------------------------------------|
| <code>MaVar = "A" + "E"</code> | "AE" | Erreur |
| <code>MaVar = 1 + 2</code> | "12" | 3 |

- ▶ Cela facilite le développement et l'usage de variable, le contenu de la variable est attendu.
- ▶ Chaque type de variable a une quantité de mémoire allouée, par conséquent, utiliser le bon type de variable permet d'économiser de la mémoire.

Voici les types de variable et le détail de chacune :

| Nom | Type | Détails |
|----------|-----------|---|
| Byte | Numérique | Nombre entier de 0 à 255 |
| Integer | Numérique | Nombre entier de -32'768 à 32'767 |
| Long | Numérique | Nombre entier de - 2'147'483'648 à 2'147'483'647 |
| Currency | Numérique | Nombre à décimale fixe de - 922'337'203'685'477.5808 à 922'337'203'685'477.5807 |
| Single | Numérique | Nombre à virgule flottante de -3.402823E38 à 3.402823E38 |
| Double | Numérique | Nombre à virgule flottante de -1.79769313486232D308 à 1.79769313486232D308 |
| String | Texte | Texte |
| Date | Date | Date et heure |
| Boolean | Boolean | True (vrai) ou False (faux) |
| Object | Objet | Objet Microsoft (exemple cellule, plage de cellule, feuille) |
| Variant | Tous | Valeur par défaut si non déclarée |

b. Concept de formulaire

Formulaire

Un **formulaire** (dit *form*) est une fenêtre d'interaction entre l'utilisateur et le système. Il s'agit d'une interface visuelle permettant de restituer et/ou collecter de l'information dans le but de faire des traitements.

L'objet formulaire est le contenant des autres objets visuels. Cela signifie qu'il comporte d'autres objets visuels, ceux-là même proposés dans la boîte à outils.

Une même application peut contenir plusieurs formulaires.

Les contrôles

Un formulaire est un contenant de contrôles. Ces **contrôles** sont des objets visuels qui permettent l'interaction avec l'utilisateur.