

## A. Objectifs du chapitre

Objet élémentaire d'Excel, les cellules seront votre première étape sur la programmation VBA des objets Excel.

Sans parcourir l'intégralité des actions possibles avec les cellules en VBA, ce chapitre vous permettra de connaître les principales propriétés et méthodes des objets élémentaires que sont les **cellules** et **plages de cellules** sous Excel.

Vous verrez ensuite des exemples de code les plus fréquents avant de terminer en validant vos nouveaux acquis au travers d'exercices.

### 1. Objet et variable Range

Lorsque vous avez créé votre première macro avec l'Enregistreur de macros dans le chapitre L'enregistreur de macros, vous avez déjà fait la connaissance de l'objet **Range**.

#### a. Objet Range

Le type de données Range est le premier que vous découvrez dans la trousse à objets de VBA Excel.

Cet objet représente aussi bien une cellule seule, une plage de cellules ou encore une série de cellules non contiguës. Lorsque vous utilisez cet objet, la syntaxe générale est la suivante :

```
Range ( Adresse )
```

*Exemple 1 : syntaxe générale de l'objet Range*

Selon vos besoins, l'adresse passée sous forme de chaîne de caractères peut représenter la ou les cellules que vous souhaitez manipuler :

```
Range ( "A1" )  
Range ( "C2:D5" )  
Range ( "A1 , B2 , G3" )  
Range ( "CelluleNommee" )
```

*Exemple 2 : possibilité de syntaxe pour l'objet Range*

## 2. Variable de type Range

Tout comme vous l'avez fait pour des chaînes de caractères, des valeurs numériques ou des booléens, il est possible de travailler avec l'objet **Range** au travers d'une variable. Ce paragraphe vous indique comment déclarer, affecter et manipuler ensuite cet objet.

### a. Déclaration

La déclaration d'une variable de type **Range** est similaire à celles vues jusqu'à présent.

```
Dim VotreCellule As Range
```

*Exemple 3 : déclaration d'une variable de type Range*

Il en est autrement pour l'affectation de valeurs.

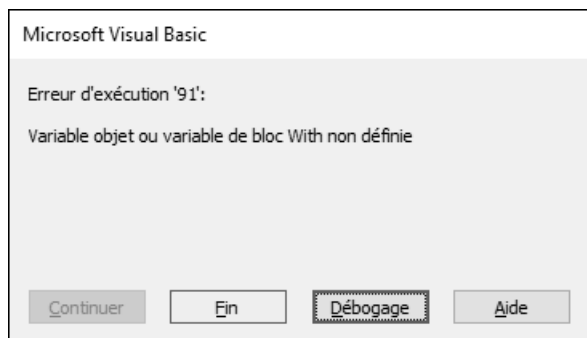
### b. Affectation

Contrairement aux types de données élémentaires comme les chaînes de caractères ou les valeurs numériques, l'affectation pour le type de donnée **Range** a pour syntaxe générale la suivante :

```
Dim rCellule As Range  
Set rCellule = Range("DateDuJour")
```

*Exemple 4 : affectation d'une variable de type Range*

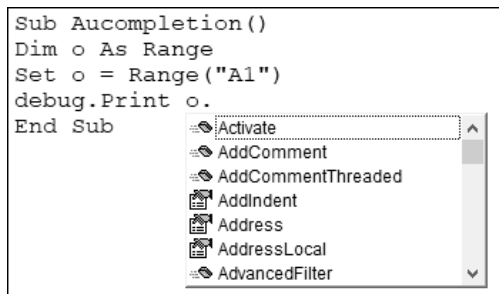
Vous pouvez constater la présence du mot-clé **Set** avant le nom de la variable **rCellule**. Si vous oubliez le mot-clé **Set**, une erreur 91 apparaîtra lors de l'exécution de la ligne de code.



L'usage du mot-clé **Set** devient nécessaire pour tous les types de données **Objet**.

### c. Utilisation

Une fois que la variable est instanciée, vous pourrez profiter de l'autocomplétion lorsque vous commencerez à vouloir accéder aux propriétés et méthodes de l'objet `Range`.



Vous pourrez saisir les premières lettres de la propriété ou de la méthode et utiliser au choix la touche `→` ou la combinaison `Ctrl Espace` pour que la ligne se renseigne automatiquement.

## B. Objet Cells

Il existe un objet plus petit encore que l'objet `Range`, qui permet lui aussi de manipuler les cellules ; il s'agit de l'objet `Cells`. Il est plus petit car là où l'objet `Range` permet de gérer une plage de cellules, l'objet `Cells` se limite à une seule cellule. Il n'est par exemple ainsi pas possible de faire référence à la plage de cellules A1:D3 avec un objet `Cells`.

Pour utiliser l'objet **`Cells`**, il existe deux syntaxes possibles, chacune fournissant à l'objet `Cells` une ligne et une colonne, coordonnées de la cellule.

```
'Syntaxe générale
Cells(indiceLigne, indiceColonne)
'Syntaxe avec des valeurs numériques uniquement
Debug.Print Cells(3,4) 'affiche le contenu de la cellule D3
'Syntaxe avec une valeur numérique pour la ligne et une chaîne de
caractères pour la colonne
Debug.Print Cells(5, "F") 'affiche le contenu la cellule F5
```

*Exemple 5 : différentes syntaxes avec l'objet Cells*

L'objet **`Cells`** est utilisable avec une variable de type **`Range`**.

```
Dim o As Range
Set o = Cells(1,3)
```

*Exemple 6 : utilisation d'une variable de type Range pour l'objet Cells*

Les types de données `Range` et `Cells` partageront des propriétés et méthodes communes.

## C. Quelques cellules particulières : ActiveCell, Selection et Target

Lorsque vous utilisez l'Enregistreur de macros également, certains mots-clés spécifiques peuvent apparaître pour définir une cellule ou une plage de cellules. Cette courte section a pour objectif de vous en expliquer les grandes lignes.

### 1. Cellule active : ActiveCell

Lorsque vous cliquez sur une cellule, on dit que vous l'activez. Cette **cellule active** est représentée par un objet natif en VBA : **ActiveCell**. Cet objet, de type Range, représente la cellule active dans votre classeur.

Ci-dessous un exemple de cellule active :

	A	B	C
1	Date	Chiffre d'affaires	Pr
2	2021-01-01	123000	cl

Lorsque vous sélectionnez une plage de cellules, la cellule à partir de laquelle vous effectuez votre sélection est la cellule active.

	A	B	C	D
1	Date	Chiffre d'affaires		Produits
2	2021-01-01	123000		Clavier
3	2021-02-01	148500		Souris
4	2021-03-01	112800		Écran
5	2021-04-01	135300		Casque
6				

Enfin, si vous effectuez une série de sélections de cellules non adjacentes, la dernière cellule activée est la cellule active.

	A	B	C	D	E
1	Date	Chiffre d'affaires		Produits	Chiffre d'affaires
2	2021-01-01	123000		Clavier	129900
3	2021-02-01	148500		Souris	155880
4	2021-03-01	112800		Écran	88332
5	2021-04-01	135300		Casque	145488

Vous pouvez voir le mot-clé `ActiveCell` dans l'exemple suivant :

```
Sub RemplirCellule()  
    ActiveCell.FormulaR1C1 = "Bonjour"  
End Sub
```

*Exemple 7 : apparition du mot-clé `ActiveCell` depuis l'Enregistreur de macros.*

## 2. Sélection active : Selection

Quand vous travaillez dans Excel et que vous sélectionnez une ou plusieurs cellules, VBA utilise un objet spécifique : **Selection**. Cet objet, de type `Range` comme `ActiveCell`, couvre un spectre similaire de fonctionnalités à celui-ci. Ce mot-clé pourra apparaître lorsque vous utilisez l'Enregistreur de macros.

Lorsque vous travaillez avec une seule cellule, l'objet `Selection` est identique à l'objet `ActiveCell`. Cependant, dès l'instant que vous utilisez une plage de cellules ou des cellules éparpillées, l'objet `Selection` vient trouver toute sa place, pour représenter l'ensemble des cellules sélectionnées.



*Bien que la sélection puisse représenter plusieurs cellules, vous n'aurez toujours qu'une seule cellule active.*

```
Sub UsageDeSelection()  
    Range("D3").Select  
    With Selection.Interior  
        .Pattern = xlSolid  
        .PatternColorIndex = xlAutomatic  
        .Color = 255  
        .TintAndShade = 0  
        .PatternTintAndShade = 0  
    End With  
End Sub
```

*Exemple 8 : cas d'utilisation du mot-clé `Selection` par l'Enregistreur de macros*

## 3. Cellule(s) impliquée(s) dans les événements Excel : Target

Comme vous le verrez dans le chapitre suivant Manipuler les feuilles Excel, le code VBA peut se déclencher automatiquement à partir d'événements qui ont lieu dans votre feuille ou votre classeur. Que cela soit un changement dans une feuille ou un double clic sur une cellule, ces événements utilisent le mot-clé **Target** pour représenter la référence de l'objet qui est ciblé dans la procédure événementielle. Dans ces événements qui ont trait à des cellules, le type de l'objet **Target** sera **Range**.

```
Private Sub Worksheet_BeforeDoubleClick(ByVal Target As Range, Cancel As Boolean)
    If Target.Address = "$C$3" Then
        MsgBox "Interdiction de modifier cette cellule", vbCritical + vbOKOnly
    End If
End Sub
```

*Exemple 9 : cas d'utilisation du mot clé Target dans un événement de feuille Excel*

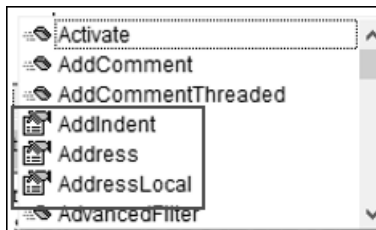
## D. Les propriétés des cellules

Dans un premier temps, nous verrons ce qu'on appelle une **propriété** puis nous verrons les principales propriétés qui peuvent servir avec les cellules.

### 1. Définition d'une propriété

En informatique, les propriétés d'un objet sont ses caractéristiques, ce qui le définit. Prenons exemple d'une voiture, vous aurez parmi ses propriétés sa marque, son modèle, sa couleur, son année de sortie. Les propriétés peuvent être de tout type, (numérique, date, chaîne de caractères). Certaines propriétés seront modifiables, d'autres seulement lisibles.

En cours de programmation, vous pouvez reconnaître une propriété à l'icône de doigt pointé :



### 2. Le contenu d'une cellule : Value, Value2

Lorsque vous commencez à programmer avec Excel, la première chose qui vous intéresse avec une cellule sera la valeur qu'elle contient. Dans ce paragraphe, vous retrouverez une partie des lignes de code que vous avez découvertes avec l'Enregistreur de macros.

La valeur contenue dans une cellule est obtenue grâce à la propriété **Value**. VBA considère d'ailleurs cette propriété comme celle par défaut.

La syntaxe générale est la suivante :

```
Debug.Print Range("A1").Value 'Affiche la valeur contenue dans la cellule A1.
```

*Exemple 10 : syntaxe générale de la propriété Value d'un objet Range*

Cette propriété peut aussi bien être lue que modifiée par votre programme.

```
Range("A1").Value = 3 'Affecte la valeur 3 dans la cellule A1
MsgBox Range("A1").Value 'Affiche la valeur de la cellule A1
```

*Exemple 11 : usage de la propriété Value*

Selon le type de valeur contenue dans la cellule, le type de donnée retourné par la propriété **Value** sera automatiquement adapté par VBA. Il conviendra donc d'utiliser les bons types de données pour vos variables, au risque d'avoir une mauvaise interprétation des valeurs par votre programme.

Dans le cas de figure où vous avez les données suivantes dans les cellules :

	A	B	C
1	123	exemple de texte	2021-08-17

Le code suivant permettra de récupérer chacune des valeurs avec le bon type de donnée :

```
Dim iNumerique As Integer, sTexte As String, dtDate As Date
iNumerique = Range("A1").Value '123
sTexte = Range("B1").Value '"exemple de texte"
dtDate = Range("C1").Value '2021-08-17
```

*Exemple 12 : récupération des valeurs au travers de la propriété Value*

De la même façon, il est possible d'écrire des valeurs dans les cellules à partir des variables :

```
Dim strTexte As String
strTexte = "Bonjour"
Range("A1").Value = strTexte
```

*Exemple 13 : affectation de valeur à une cellule par la propriété Value*

Il existe une autre propriété, **Value2**, plus rarement utilisée mais que vous pourrez rencontrer dans les macros d'autres développeurs ; elle est similaire à **Value**, à ceci près qu'elle ne retourne pas de valeur de type **Currency** ou **Date** (ce type étant remplacé par le numéro de série, par exemple 2021-08-17 qui retournera 44425).

En reprenant le cas de l'illustration précédente, la valeur récupérée sera la suivante :

```
?Range("C1").Value2
44425
```

*Exemple 14 : usage de la propriété Value2*

## 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)**

Renvoie l'année, un nombre entier entre 1900 et 9999.

*Cette fonction permet d'isoler l'année d'une date ; exemple : la cellule A1 contient la valeur 12/12/2021, la fonction =ANNEE(A1) renvoie 2021.*

#### **AUJOURDHUI()**

Renvoie la date du jour au format de date.

*Cette fonction a la particularité de ne pas posséder d'arguments, il n'y a rien à écrire entre parenthèses, mais attention à ne pas oublier ces deux parenthèses.*

#### **DATE(année;mois;jour)**

Renvoie un numéro de série.

*La fonction **DATE** permet la reconstitution d'une date à partir de trois valeurs : supposons qu'en A1, B1 et C1 nous ayons respectivement l'année 2021, le mois 12, le jour 20 ; la formule =DATE(A1;B1;C1) renverra 20/12/2021.*

#### **DATEVAL(date\_texte)**

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

*Sous forme de texte signifie que la date saisie 20/12/2021 n'est pas reconnue comme date par Excel mais comme donnée de type texte. Ceci arrive généralement lorsque des données ont été importées.*



## Les calculs

### **FIN.MOIS(date\_départ;mois)**

Renvoie le numéro de série du dernier jour du mois situé dans un intervalle exprimé en nombre de mois dans le futur ou le passé.

*Cette fonction est très utile pour calculer les dates d'échéances.*

*Exemple : pour déterminer la date d'échéance d'une facture payable à 30 jours fin de mois en supposant que la date de facturation saisie en B2 est le 15/01/2022, la fonction s'écrit =FIN.MOIS(B2;1) et renvoie 28/02/2022.*

### **FRACTION.ANNEE(date\_début;date\_fin;[base])**

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

*Cette fonction est pratique pour calculer le nombre d'années d'ancienneté, l'âge et, plus généralement, le nombre d'années écoulées entre deux dates. La valeur obtenue est souvent une valeur décimale du type 12,4568256 (12 années et plus...).*

### **HEURE(numéro\_de\_série)**

Renvoie le nombre d'heures : un nombre entier entre 0 et 23.

*Numéro\_série correspond à un temps écrit de la manière suivante hh:mm:ss (heures:minutes:secondes) exemple : 12:25:30 renverra 12.*

### **JOUR(numéro\_de\_série)**

Donne le jour du mois (un nombre entier entre 1 et 31).

*Même principe que la fonction ANNEE, elle isole le jour d'une date quelconque.*

### **JOURS(date\_fin;date\_début)**

Calcule le nombre de jours entre les deux dates.

*À noter que vous obtenez le même résultat par simple soustraction. Par exemple : A1 contient 01/01/2022 et A2 contient 31/01/2022 en saisissant =A2-A1 vous obtenez 30.*

### **JOURS360(date\_début;date\_fin;[méthode])**

Calcule le nombre de jours séparant deux dates sur la base d'une année de 360 jours (12 mois de 30 jours).

### **JOURSEM(numéro\_de\_série;[type\_retour])**

Renvoie un chiffre entre 1 et 7 désignant le jour de la semaine d'une date.

### **MAINTENANT()**

Renvoie la date du jour et l'heure de l'instant présent sous la forme d'une date et d'un temps comme par exemple 09/02/2022 11:14.

**MINUTE(numéro\_de\_série)**

Renvoie les minutes d'un temps.

*Comme la fonction HEURE, cette fonction isole les minutes indiquées dans un temps.  
Exemple : 12:25:40 renvoie la valeur 25.*

**MOIS(numéro\_de\_série)**

Renvoie le mois d'une date.

**MOIS.DECALER(date\_départ;mois)**

Renvoie une 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])**

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

**NB.JOURS.OUVRES.INTL(date\_début;date\_fin;[weekend];[jours\_fériés])**

Renvoie le nombre de jours ouvrés entiers compris entre deux dates à l'aide de paramètres identifiant les jours du week-end et leur nombre (cf. section Calculer le nombre de jours ouvrés ou non entre deux dates).

**NO.SEMAIN(numéro\_de\_série;[type\_retour])**

Renvoie le numéro de série en numéro de semaine dans l'année correspondant à la date indiquée.

**NO.SEMAIN.ISO(date)**

Renvoie le numéro ISO de la semaine de l'année correspondant à une date donnée. ISO correspond à la norme européenne.

**SECONDE(numéro\_de\_série)**

Renvoie les secondes d'un temps.

*C'est le même principe que pour les fonctions HEURE et MINUTE.*

**SERIE.JOUR.OUVRE(date\_départ;nb\_jours;[jours\_fériés])**

Renvoie le numéro de série de la date avant ou après le nombre de jours ouvrés spécifiés (cf. section Calculer la date située après un nombre de jours ouvrés donné).

**SERIE.JOUR.OUVRE.INTL(date\_départ;nb\_jours;  
[nb\_jours\_week-end];[jours\_fériés])**

Renvoie 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.

## Les calculs

### TEMPS(heure;minute;seconde)

Recompose un temps à partir de trois valeurs numériques représentant les heures, les minutes et les secondes.

*Exemple : A1 contient l'heure 14 ; A2 contient les minutes 19 ; A3 contient les secondes 40 ; la formule =TEMPS(A1;A2;A3) renverra 14:19:40.*

### TEMPSVAL(heure\_texte)

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

*Même logique que la fonction DATEVAL pour les dates ; cette fonction permet la correction de données importées dans un mauvais format.*

### 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		<b>Noms</b>	<b>Date de naissance</b>	<b>Age</b>	<b>Jour et mois de naissance</b>
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

*La fonction TEXTE a permis, à partir de la date de naissance, la transcription du jour en nombre et du mois en lettre ce qui, à partir du 17/05/1968, donne 17 mai.*

## 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 est :

**DATEDIF** est **DATEDIF**(Date\_début;Date\_fin;Type).

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

"y" pour calculer la différence absolue en années (nombre entier).

"m" pour calculer la différence absolue en mois.

"d" pour calculer la différence absolue en jour.

"ym" pour calculer le reliquat de mois une fois déduites toutes les années entières.

"yd" pour calculer le reliquat de jours une fois déduites toutes les années entières.

"md" pour calculer le reliquat de jours une fois déduits tous les mois entiers.

*Voici un exemple d'utilisation :*

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

## Les calculs

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		<b>Date de naissance</b>	<b>25/02/1964</b>	<b>FORMULES</b>	
3		<b>Age en années</b>	<b>51 ans</b>	<b>=DATEDIF(C1;AUJOURDHUI();"y")&amp;" ans"</b>	
4		<b>Nombre de mois</b>	<b>51 ans et 4 mois</b>	<b>=DATEDIF(C1;AUJOURDHUI();"y")&amp;" ans et "&amp;DATEDIF(C1;AUJOURDHUI();"ym")&amp;" mois"</b>	
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		<b>Date de début</b>	<b>30/11/2013</b>
2		<b>Date de fin</b>	<b>01/12/2015</b>
4		<b>Nombre de jours ouvrés</b>	<b>522</b>
5		<b>Formule de calcul</b>	<b>=NB.JOURS.OUVRES(C1;C2)</b>
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 B4 à B15.

Année		2015		Nombre de jours chômés <i>(hors week-end)</i>		9			
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)**.

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 exclusion du calendrier des jours de travail (jours fériés, congés, absence...). Cet argument est facultatif.