

Appleworks GS

LES FONCTIONS ET FORMULES

une traduction B. CAPSLOCK

INTRODUCTION:

Dans Appleworks GS, la base de données et le tableur permettent l'emploi de fonctions et de formules dans les champs des fiches et les cellules des feuilles de calcul.

Un champ de calcul d'une base de données contient des formules calculant la valeur du champ en fonction de certaines autres valeurs. Par exemple, on peut avoir la formule suivante dans un champ calculé: "[champ1]+[champ2]". Dans ce cas, le champ calculé contiendra la somme des valeurs contenues dans les champs 1 et 2.

Une cellule du tableur peut contenir une formule de calcul qui déterminera le contenu en fonctions de valeurs contenues dans d'autres cellules. Par exemple une cellule calculée peut contenir la formule: "=A1+A2". Dans ce cas, la cellule calculée affichera la somme des valeurs contenues dans les cases A1 et A2.

Une formule de champ ou de cellule peut contenir un ou plusieurs éléments, y compris des constantes, des références aux autres cellules ou champs et des fonctions. Ces éléments peuvent être combinés à l'aide d'opérateurs donnant un résultat affiché dans la cellule ou le champ calculé.

Un des éléments de la formule peut être une fonction. Les fonctions de Appleworks GS sont des formules vides dans lesquelles on indique les valeurs (appelés arguments) permettant d'effectuer des calculs complexes.

LES FONCTIONS:

Appleworks GS dispose de 73 fonctions pour le tableur et de 69 fonctions pour la base de données. Ces fonctions sont listées dans le chapitre FONCTIONS PAR CATEGORIES et expliquées en détail au chapitre LISTE ALPHABETIQUE DES FONCTIONS.

Les constituants d'une fonction:

Chaque fonction est constituée d'un nom suivi d'un ou plusieurs arguments entre parenthèses. Un argument est une valeur utilisée par la fonction afin d'effectuer ses calculs. Par exemple, la fonction SUM a besoin d'une liste de valeurs comme arguments afin d'en calculer la somme:

=SUM (A1...C1,E1,100)

=SUM: nom de la fonction
A1,C1,E1 et 100 sont les arguments

Dans les fonctions, les arguments peuvent être des variables ou des constantes. Une variable est une valeur pouvant changer en fonction d'autres valeurs. Dans Appleworks GS les variables sont des références à des champs (base de données) ou à des cellules (tableur).

Appleworks GS

Une constante est une valeur fixe: valeur numérique ou chaîne de caractères (suite de caractères mis entre guillemets). Pour délimiter une chaîne de caractères on peut utiliser soit les " soit les '.

Arguments pour les fonctions:

Les arguments sont des valeurs que vous passez à la fonction pour qu'elle puisse effectuer ses calculs. Les différentes fonctions d'Appleworks GS ont besoin de plusieurs types d'arguments. Ces arguments sont regroupables en plusieurs catégories: expressions, nombres, liste numérique, zone (dans un tableur), chaîne de caractères, liste de chaîne, valeurs logiques, liste logique, date et valeurs financières.

Expression:

Une expression peut être une valeur - le contenu d'une cellule ou d'un champ, une valeur calculée, une valeur retournée par une autre fonction. Le mot EXPRESSION est utilisé dans cette documentation pour parler d'une valeur calculée, d'une valeur numérique ou d'une chaîne.

Nombre:

Quelques fonctions ont besoin d'une ou plusieurs valeurs numériques en tant qu'argument. Chaque valeur numérique peut être exprimée sous forme de valeur simple ou d'expression numérique (y compris une autre fonction ou un groupe de fonctions) donnant une valeur numérique en résultat. Les constantes numériques sont exprimées par des nombres; les variables numériques sont soit des cellules de tableur soit des champs de base de données formatés pour contenir des nombres. Par exemple, la fonction SIN a besoin d'une valeur numérique en argument. On peut donc écrire SIN(2), SIN(10/5) ou SIN(A1-2) - avec la cellule A1 contenant la valeur 4 - et le résultat sera toujours le même.

Liste numérique:

Certaines fonctions nécessitent une liste de valeurs numériques comme argument. Chaque valeur numérique peut être une valeur simple ou une expression définissant un nombre. Par exemple, la fonction SUM a besoin d'une liste de valeurs numériques afin de calculer la somme de ces valeurs.

Zone:

Certaines fonctions ont besoin d'une zone de valeurs dans un tableur. Ces valeurs sont spécifiées par la référence de la première cellule de la zone suivi de deux points (..) puis suivi de la référence de la dernière cellule de la zone. Par exemple la zone A1..A10 définit toutes les cellules comprises entre A1 et A10, bornes comprises.

Chaîne:

Certaines fonctions ont besoin d'un argument chaîne de caractères, c'est à dire un groupe de caractères alpha-numériques. Par exemple, la fonction LEN calcule la longueur (nombre de caractères) d'une chaîne. Si vous utilisez une constante chaîne comme argument, cette chaîne doit être entourée de guillemets (ces guillemets ne sont pas considérés comme faisant partie de la chaîne).

Appleworks GS

Liste de chaines:

Certaines fonctions ont besoin d'une liste de une ou plusieurs chaines. Par exemple la fonction CONCAT a besoin d'une ou plusieurs chaines en argument et retourne une seule chaine définie en 'collant' les unes à la suite des autres les chaines de l'argument.

Valeur logique:

Une valeur logique est une expression qui évalue par VRAI ou FAUX (Appleworks GS code la valeur VRAIE par la valeur numérique 1, et la valeur FAUX par la valeur numérique 0; toute valeur différente de 0 est considérée comme VRAIE). Certaines fonctions nécessitent une ou plusieurs valeurs logiques comme argument et retournent la valeur VRAIE ou FAUX. Par exemple, la fonction NOT a besoin d'une valeur logique en argument et retourne la valeur opposée comme résultat.

Liste logique:

Certaines fonctions nécessitent une liste d'une ou plusieurs valeurs logiques. Par exemple, la fonction OR a besoin d'une liste de valeurs logiques et retourne la valeur VRAIE si une des valeurs de la liste est VRAIE.

Date:

Certaines fonctions ont besoin d'une date en argument et retournent une valeur calculée à partir de la date. Par exemple, la fonction MONTH a besoin d'une date en argument et retourne une chaine qui représente la partie mensuelle de la date. Une date s'exprime comme une chaine; par exemple "7/21/88" et "July 22, 1988".

Arguments financiers:

Les fonctions financières d'Appleworks GS retournent des valeurs calculées à partir de valeurs financières actuelles ou projetées. Par exemple, la fonction PMT calcule le remboursement périodique nécessaire pour payer un crédit sur un nombre fixé de périodes. La fonction PMT a besoin de 3 arguments: le montant emprunté, le taux d'intérêt et le nombre de remboursements.

Les taux d'intérêt et les périodes de références doivent s'exprimer sur des périodes identiques (annuités, mensualités...); au besoin convertir le taux annuel en taux mensuel

Cellules et champs:

On peut définir les arguments des fonctions d'Appleworks GS par des valeurs constantes ou en se référant au contenu de cellules du tableur ou de champs de la base de données. Lorsqu'on se réfère à des cellules ou des champs, assurez-vous que ces cellules ou ces champs contiennent des données du type requis.

Par exemple, si une fonction d'Appleworks GS a besoin d'une valeur numérique, assurez-vous que la cellule du tableur ou le champ de la base de données contient bien une valeur numérique. Si vous donnez à une fonction des arguments qui ne sont pas du type requis, un message d'erreur est affiché. Toutefois, dans la mesure du possible, Appleworks GS essaie d'évaluer l'expression même si des arguments ne sont pas du type requis. Par exemple, si on donne une chaine en argument à la place d'un nombre, la chaine est traitée comme étant la

Appleworks GS

valeur entière 0. Si on fournit un nombre à la place d'une chaîne, le nombre est traité comme étant la chaîne vide ("").

LISTE DES FONCTIONS PAR CATEGORIE:

Les fonctions arithmétiques:

Abs(nombre) : retourne la valeur absolue de 'nombre'
Exp(nombre) : retourne l'exponentielle de 'nombre'
Frac(nombre) : retourne la partie décimale de 'nombre'
Inf : retourne la valeur 'infini'
Int(nombre) : retourne la partie entière de 'nombre'
Ln(nombre) : retourne le logarithme naturel de 'nombre'
('nombre' doit être positif)
Log(nombre) : retourne le logarithme en base 10 de 'nombre'
Mod(nb1, nb2) : retourne nb1 modulo nb2 (le reste de la division de nb1 par nb2)
Product(liste num): retourne le produit de tous les nombres de la liste numérique
Round(nb1,nb2) : retourne le nombre 1 arrondi au nombre de décimales indiqué dans nombre 2
Sgn(nombre) : retourne le signe de 'nombre'. Si 'nombre' est positif, la valeur retournée est 1; s'il est négatif, c'est -1; et s'il est nul, c'est 0
Sqrt(nombre) : retourne la racine carrée de 'nombre'. Si 'nombre' est négatif, le résultat est ERROR

Les fonctions trigonométriques:

Acos(nombre) : retourne en radians l'arcosinus de 'nombre'
Asin(nombre) : retourne en radians l'arcsinus de 'nombre'
Atan(nombre) : retourne en radians l'arctangente de 'nombre'
Cos(nombre) : retourne le cosinus de 'nombre'. La fonction suppose que 'nombre' est en radians
Degrées(nombre) : retourne la valeur en degrés de 'nombre' exprimé en radians.
Pi : retourne une approximation de PI
Radians(nombre) : retourne la valeur en radians de 'nombre' exprimé en degrés
Sin(nombre) : retourne le sinus de 'nombre'. La fonction suppose que 'nombre' est en radians
Tan(nombre) : retourne la tangente de 'nombre'. La fonction suppose que 'nombre' est en radians

Les fonctions statistiques:

Avg(liste num) : retourne la moyenne de la liste numérique
Count(liste) : retourne le nombre d'items de la liste
Max(liste num) : retourne la valeur maxi contenue dans la liste numérique
Min(liste num) : retourne la valeur mini contenue dans la liste numérique
Random(nombre) : retourne un nombre aléatoire compris entre 0 et 'nombre'-1. Si l'argument est 0 ou 1, la valeur aléatoire générée est plus grande ou égale à 0 et plus petite que 1
Stdv(liste num) : retourne l'écart type des valeurs de la liste numérique
Sum(liste num) : retourne la somme des valeurs de la liste numérique
Var(liste num) : retourne la variance des nombres de la liste numérique

Appleworks GS

Les fonctions financières:

Les taux d'intérêt et les périodes de références doivent s'exprimer sur des périodes identiques (annuités, mensualités...); au besoin convertir le taux annuel en taux mensuel

FV(versement, intérêt, durée)	: calcule la valeur future d'un investissement
IRR(estimation, zone numérique)	: calcule le taux d'intérêt (rendement) produit par des entrées irrégulières d'argent (zone)
NPV(intérêt, zone numérique)	: calcule la valeur actuelle de futurs remboursements (zone)
Pmt(pv, intérêt, durée)	: calcule la valeur actuelle d'une série de remboursements identiques ou d'un investissement
PV(remboursement, taux, durée)	: Calcule la valeur actuelle d'une série de remboursements identiques à période fixe ou la rémunération d'un investissement.
Rate(fv, pv, durée)	: calcule le taux d'intérêt d'un investissement constant, à épargne régulière ou à versement unique
Term(remboursement, taux, fv)	: retourne le nombre de remboursements pour amortir un prêt à mensualités identiques (ou pour qu'un capital atteigne une valeur déterminée)

Les fonctions chaînes:

Concat(liste chaîne)	: retourne une chaîne unique formée de la succession des chaînes de la liste
Len(chaîne)	: retourne un nombre égal à la longueur de 'chaîne'
Lower(chaîne)	: transforme tous les caractères de 'chaîne' en minuscules
Null	: retourne une chaîne vide ("")
Position(ch1, ch2)	: retourne un nombre correspondant à la première position de 'chainel' dans 'chaîne2'
Proper(chaîne)	: retourne une chaîne avec le premier caractère en majuscules et tous les autres en minuscules
SubStr(chaîne, début, longueur)	: retourne une partie de 'chaîne' définie par les deux valeurs numériques
Upper(chaîne)	: transforme tous les caractères de 'chaîne' en majuscules

Les fonctions de conversion:

Asc(chaîne)	: retourne la valeur ASCII du premier caractère de 'chaîne'
Chr(nombre)	: retourne la chaîne correspondant à la valeur ASCII représentée par 'nombre'
String(nombre)	: retourne une chaîne constituée des chiffres de 'nombre'
Val(chaîne)	: retourne la valeur numérique de 'chaîne'

Les fonctions de date:

Date(nombre)	: retourne une chaîne-date équivalente à 'nombre'
Day(chaîne)	: retourne un nombre exprimant le jour de la date constituée par 'chaîne'

Appleworks GS

Month(chaine) : retourne une chaine exprimant le mois de la date constituée par 'chaine'
NumDate(chaine) : retourne une valeur entière représentant la date exprimée dans 'chaine'
Today : retourne une chaine représentant la date du jour
Weekday(chaine) : retourne une chaine représentant le nom du jour de la date exprimée dans 'chaine'
Year(chaine) : retourne un nombre représentant l'année de la date exprimée par 'chaine'

Les fonctions logiques:

And(liste logique): retourne la valeur numérique 1 (VRAI) si toutes les valeurs de la liste logique sont VRAIES (ou évaluées à la valeur numérique 1)
Begins(recherche,chaine) : retourne la valeur 1 (VRAI) si la 'chaine' commence par 'recherche'; ne tient pas compte du format Majuscules/minuscules
Contains(recherche,chaine) : retourne la valeur 1 (VRAI) si la 'chaine' contient 'recherche'; ne tient pas compte du format Majuscules/minuscules
Ends(recherche,chaine) : retourne la valeur 1 (VRAI) si la 'chaine' se termine par 'recherche'; ne tient pas compte du format Majuscules/minuscules
False : retourne la valeur numérique 0
If(expression logique,expr1,expr2): si l'expression logique est VRAIE (ou évaluée à la valeur numérique 1 ou toute autre valeur non nulle), cette fonction retourne 'expression1'; sinon c'est 'expression2' qui est retourné
IsBlank(expression): retourne la valeur numérique 1 (VRAI) si 'expression' est vide
IsEmpty(expression): retourne la valeur numérique 1 (VRAI) si 'expression' est vide
IsError(expression): si 'expression' est évaluée à ERROR, alors cette fonction retourne la valeur 1 (VRAI)
IsNA(expression) : retourne la valeur 1 (VRAI) si 'expression' est la valeur spéciale N/A
IsNumber(expression): retourne la valeur 1 (VRAI) si 'expression' est un nombre
IsString(expression): retourne la valeur 1 (VRAI) si 'expression' est une chaine
Not(expression) : retourne la valeur 1 (VRAI) si 'expression' est FAUX (ou évalué à la valeur 0)
Or(liste logique) : retourne la valeur numérique 1 (VRAI) si une des valeurs de la liste logique est VRAIE (ou évaluée à la valeur numérique 1)
True : retourne la valeur numérique 1 (VRAI)

Les fonctions spéciales:

Alert(chaine) : retourne une fenêtre d'alerte avec 'chaine' affiché dedans
Choose(nombre,liste): retourne la valeur numérique de la liste se trouvant à la position 'nombre'
Error : retourne la valeur ERROR
HLookup(valeur,zone,distance): cherche 'valeur' dans une table contenue dans la 'zone' du tableur. Elle trouve la valeur immédiatement plus petite ou égale à 'valeur' dans la première ligne de 'zone' et retourne alors la valeur trouvée 'distance' lignes

Appleworks GS

plus bas (on suppose que la première ligne de 'zone' est triée en ordre croissant)

NA : retourne la valeur N/A
VLookup(valeur,zone,distance): cherche 'valeur' dans une table contenue dans la 'zone' du tableur. Elle trouve la valeur immédiatement plus petite ou égale à 'valeur' dans la première colonne de 'zone' et retourne alors la valeur trouvée 'distance' colonnes plus à droite (on suppose que la première colonne de 'zone' est triée en ordre croissant)

LISTE DES FONCTIONS PAR ORDRE ALPHABETIQUE:

Abs(nombre)

retourne la valeur absolue de 'nombre'

exemples:

Abs(-1) retourne 1

Abs(1) retourne 1

Abs(0) retourne 0

Abs(A1) retourne la valeur absolue du nombre dans la cellule A1

Abs([champ]) retourne la valeur absolue du nombre dans le [champ] spécifié

Acos(nombre)

retourne en radians l'arcosinus de 'nombre'

exemples:

Acos(-0.5) retourne 2.094 (2 pi/3 radians)

Acos(A1) retourne l'arcosinus de la valeur dans la cellule A1

Acos([champ]) retourne l'arcosinus de la valeur dans le [champ]

Alert(chaine)

retourne une fenêtre d'alerte avec 'chaine' affiché dedans

exemples:

la formule If(IsError(A1),Alert("ERREUR CELLULE A1"),NULL)

retourne une fenêtre d'alerte avec le message ERREUR CELLULE A1 si une erreur est détectée dans la cellule A1

And(liste logique)

retourne la valeur numérique 1 (VRAI) si toutes les valeurs de la liste logique sont VRAIES (ou évaluées à la valeur numérique 1)

exemples:

And(1+1=2,2+2=4) retourne VRAI

And(A1=B1,C1=D1) retourne VRAI si la valeur dans la cellule A1 est égale à la valeur dans la cellule B1; ET que la valeur dans la cellule C1 est égale à la valeur dans la cellule D1

Appleworks GS

Asc(chaine)

retourne la valeur ASCII du premier caractère de 'chaine'

exemples:

Asc("A") retourne 65 (ASCII de "A")

Asc("alphabet" retourne 97 (ASCII de "a")

Asc(A1) retourne le code ASCII du premier caractère situé dans A1

Asc([champ]) retourne le code ASCII du premier caractère situé dans le [champ]

Asin(nombre)

retourne en radians l'arcsinus de 'nombre'

exemples:

Asin(-0.5) retourne -0.524 (-pi/6 radians)

Asin(A1) retourne l'arcsinus de la valeur dans la cellule A1

Asin([champ]) retourne l'arcsinus de la valeur dans le [champ]

Atan(nombre)

retourne en radians l'arctangente de 'nombre'

exemples:

Atan(1) retourne 0.785 (pi/4 radians)

Atan(A1) retourne l'arctangente de la valeur dans la cellule A1

Atan([champ]) retourne l'arctangente de la valeur dans le [champ]

Avg(liste num)

retourne la moyenne de la liste numérique

exemples:

Avg(1,3,5) retourne 3

Avg(A1..A10) retourne la moyenne des valeurs contenues des cellules A1 à A10 (ignore les cellules vides)

Avg([champ1],[champ2],...) retourne la moyenne des valeurs contenues dans les champs spécifiés

Begins(recherche,chaine)

retourne la valeur 1 (vrai) si la chaîne 'recherche' est en tête de la 'chaine'. Ne tiens pas compte du format majuscules ou minuscules des caractères

exemples:

Begins(chaine,[champ]) retourne la valeur 1 si 'chaine' commence le contenu de [champ]; 0 dans le cas contraire

Begins(chaine,A1) retourne la valeur 1 si 'chaine' commence le contenu de la cellule A1; 0 dans le cas contraire

Choose(nombre,liste)

retourne la valeur numérique de la liste numérique/chaine se trouvant à la position 'nombre'

Appleworks GS

exemples:

Choose(2,5,6,8) retourne la valeur 6 (le 2ème nombre de la liste)
Choose(3,A1..A10) retourne la 3ème valeur dans la liste des cellules (ignorant les cellules vides)
Choose(3,[champ1],[champ2],..) retourne la 3ème valeur de la liste des valeurs se trouvant dans les champs

Chr(nombre)

retourne la chaîne correspondant à la valeur ASCII spécifiée par 'nombre'

exemples:

Chr(65) retourne "A"
Chr(97) retourne "a"
Chr(A1) retourne le caractère ayant pour valeur ASCII le nombre se trouvant dans la cellule A1
Chr([champ]) retourne le caractère ayant pour valeur ASCII le nombre se trouvant dans le [champ]

Concat(liste chaîne)

retourne une chaîne unique formée de la succession des chaînes de la liste

exemples:

Concat("voir"," le"," chat") retourne la chaîne "voir le chat" (à noter que Concat("voir","le","chat") donnerait "voirlechat")
Concat(A1,A2,A3) retourne une chaîne unique formée des juxtapositions des valeurs contenues dans les cellules A1,A2 et A3
Concat([champ1],[champ2],[champ3]) retourne une chaîne unique formée des juxtapositions des valeurs contenues dans les champs 1,2 et 3

Contains(recherche,chaîne)

retourne la valeur 1 (vrai) si la chaîne 'recherche' est contenue dans la 'chaîne'. Ne tiens pas compte du format majuscules ou minuscules des caractères

exemples:

Contains(chaîne,[champ]) retourne la valeur 1 si 'chaîne' est contenu dans [champ]; 0 dans le cas contraire
Contains(chaîne,A1) retourne la valeur 1 si 'chaîne' est contenu dans la cellule A1; 0 dans le cas contraire

Cos(nombre)

retourne le cosinus de 'nombre'. La fonction suppose que 'nombre' est exprimé en radians

exemples:

Cos(1.047) retourne 0.5
Cos(A1) retourne le cosinus de la valeur contenue dans A1

Appleworks GS

`Cos([champ])` retourne le cosinus de la valeur contenue dans le [champ]

`Count(liste)`

retourne le nombre d'items de la liste contenant des valeurs

exemples:

`Count(A1..A10)` retourne le nombre de cellules entre A1 et A10 qui contiennent des valeurs

`Count([champ1],[champ2],...)` retourne le nombre de champs de la liste contenant des valeurs

`Date(nombre)`

retourne une chaîne-date équivalente à 'nombre'. La valeur 1 retourne la date "Jan. 1, 1900", la valeur 2 retourne "Jan. 2, 1900" et ainsi de suite

exemples:

`Date(10894)` retourne la date "Oct. 29, 1929"

`Date(32873)` retourne la date "Jan. 1, 1990"

`Date(A1)` retourne la date correspondant à la valeur numérique contenue dans la cellule A1

`Date([champ])` retourne la date correspondant à la valeur numérique contenue dans le [champ]

`Day(chaine)`

retourne un nombre exprimant le jour de la date constituée par 'chaine'

exemples:

`Day("Jan. 24,1988")` retourne 24

`Day("Jan. 1, 1989")` retourne 1

`Day(A1)` retourne le quantième mensuel du jour de la date située dans la cellule A1

`Day([champ])` retourne le quantième mensuel du jour de la date située dans le [champ]

`Degrees(nombre)`

retourne la valeur en degrés de 'nombre' exprimé en radians

exemples:

`Degrees(Pi)` retourne la valeur 180

`Degrees(3.14)` retourne la valeur 179.9087476

`Ends(recherche,chaine)`

retourne la valeur 1 (vrai) si la chaîne 'recherche' est en fin de la 'chaine'. Ne tiens pas compte du format majuscules ou minuscules des caractères

exemples:

`Ends(chaine,[champ])` retourne la valeur 1 si 'chaine' termine le

Appleworks GS

contenu de [champ]; 0 dans le cas contraire
Ends(chaine,A1) retourne la valeur 1 si 'chaine' termine le
contenu de la cellule A1; 0 dans le cas contraire

Error

retourne la valeur ERROR

exemples:

Error retourne la valeur ERROR

Exp(nombre)

retourne la valeur e élevée à la puissance 'nombre'. Le symbole e
représente la constante mathématique 2.7182818... , base des
logarithmes naturels. La fonction EXP est l'inverse de la
fonction LN.

exemples:

Exp(-1) retourne .368

Exp(0) retourne 1

Exp(1) retourne 2.718

Exp(A1) retourne la valeur de e à la puissance 'nombre contenu
dans la cellule A1'

Exp([champ]) retourne la valeur de e à la puissance 'nombre
contenu dans le [champ]'

False

retourne la valeur numérique 0

exemples:

False retourne la valeur FAUX (0)

Frac(nombre)

retourne la partie décimale de 'nombre'

exemples:

Frac(3.5) retourne 0.5

Frac(2.75) retourne 0.75

Frac(A1) retourne la partie décimale de la valeur contenue dans
la cellule A1

Frac([champ]) retourne la partie décimale de la valeur contenue
dans le [champ]

FV(Versement,intérêt,durée)

calcule la valeur future d'un investissement.

'Versement' est le montant du capital épargné; ce versement est
une valeur positive et le signe \$ en en-tête est optionnel.

'Intérêt' est le taux; cette valeur est exprimée en % (ex: 10%)
ou sous forme décimale (ex: 0.10)

'Durée' est le nombre de périodes durant lesquelles le capital
est placé

exemples:

Appleworks GS

FV(5000,8%,20) retourne le montant future d'un investissement de \$5000 au taux annuel d'intérêt de 8% pendant 20 ans

HLookup(valeur,zone,distance)

cherche 'valeur' dans une table contenue dans la 'zone' du tableur. Elle trouve la valeur immédiatement plus petite ou égale à 'valeur' dans la première ligne de 'zone' et retourne alors la valeur trouvée 'distance' lignes plus bas (on suppose que la première ligne de 'zone' est triée en ordre croissant)

exemples:

Hlookup(2,A1..D4,3) retourne la valeur trouvée selon la méthode suivante: trouve la valeur V la plus proche de 2 (inférieure ou égale) dans les lignes A1 à D1 puis retourne la valeur située dans la cellules 3 lignes en dessous de la cellule contenant V

If(expression logique,expr1,expr2)

si l'expression logique est VRAIE (ou évaluée à la valeur numérique 1 ou toute autre valeur non nulle), cette fonction retourne 'expression1'; sinon c'est 'expression2' qui est retourné

exemples:

If(A1<90,"B","A") retourne la valeur "B" si la valeur contenue dans la cellule A1 est inférieure à 90; et elle retourne "A" dans le cas contraire

If(A1=B1,10.5,12.75) retourne la valeur 10.5 si la valeur contenue dans la cellule A1 est égale à la valeur contenue dans la cellule B1; sinon elle retourne 12.75

Inf

retourne la valeur 'infini'

Int(nombre)

retourne la partie entière de 'nombre'

exemples:

Int(3.5) retourne 3

Int(2.75) retourne 2

Int(A1) retourne la partie entière de la valeur contenue dans la cellule A1

Int([champ]) retourne la partie entière de la valeur contenue dans le [champ]

IRR(estimation,zone numérique)

calcule le taux d'intérêt (rendement) produit par des entrées irrégulières d'argent (zone).

Le taux d'intérêt est le taux produisant la valeur nette actuelle des entrées d'argent d'un investissement pour égaler le cout de cet investissement.

Appleworks GS

'Estimation' est votre approximation de ce taux d'intérêt; la fonction IRR utilise cette valeur afin de commencer ses calculs. Cette valeur sera indiquée en % ou sous forme décimale (ex: 10% ou 0.10).

Cette fonction sert à évaluer le rendement d'un investissement.

exemples:

IRR(0.10,A1..A10) retourne le taux d'intérêt produit par un investissement remboursé par les versements périodiques stockés dans les cellules A1 à A10. Le taux estimé étant ici de 10%.

IsBlank(expression)

retourne la valeur numérique 1 (VRAI) si 'expression' est une chaîne vide

exemples:

IsBlank(A1) retourne 1 si la cellule A1 ne contient aucune valeur

IsEmpty(expression)

retourne la valeur numérique 1 (VRAI) si 'expression' est vide

exemples:

IsEmpty(A1) retourne 1 si la cellule A1 ne contient aucune valeur
IsEmpty([champ]) retourne 1 si le [champ] ne contient aucune valeur

IsError(expression)

si 'expression' est évaluée à ERROR, alors cette fonction retourne la valeur 1 (VRAI)

exemples:

IsError(SQRT(A1)) retourne 1 (VRAI) si le contenu de la cellule A1 est négatif (impossible de calculer la racine carrée d'un nombre négatif)

IsNA(expression)

retourne la valeur 1 (VRAI) si 'expression' est la valeur spéciale N/A

exemples:

IsNA(A3) retourne 1 si la cellule A3 contient la valeur spéciale N/A

IsNumber(expression)

retourne la valeur 1 (VRAI) si 'expression' est un nombre

exemples:

IsNumber(A1) retourne VRAI si le contenu de A1 est un nombre
IsNumber([champ]) retourne VRAI si le contenu du [champ] est un nombre

IsString(expression)

retourne la valeur 1 (VRAI) si 'expression' est une chaîne

exemples:

IsString(A1) retourne VRAI si le contenu de A1 est une chaîne

IsString([champ]) retourne VRAI si le contenu du [champ] est une chaîne

Len(chaine)

retourne un nombre égal à la longueur de 'chaîne'

exemples:

Len("Chat") retourne 4

Len("Salut") retourne 5

Len(A1) retourne la longueur de la chaîne stockée en A1

Len([champ]) retourne la longueur de la chaîne stockée dans le [champ]

Ln(nombre)

retourne le logarithme naturel de 'nombre' ('nombre' doit être positif). La fonction Ln est l'inverse de la fonction Exp

exemples:

Ln(1) retourne la valeur numérique 0

Ln(100) retourne la valeur numérique 4.605

Ln(0) retourne INF

Ln(A1) retourne le logarithme naturel de la valeur contenue dans la cellule A1

Ln([champ]) retourne le logarithme naturel de la valeur contenue dans le [champ]

Log(nombre)

retourne le logarithme en base 10 de 'nombre'

exemples:

Log(1) retourne la valeur numérique 0

Log(100) retourne la valeur numérique 2

Log(0) retourne INF

Log(A1) retourne le logarithme décimal de la valeur contenue dans la cellule A1

Log([champ]) retourne le logarithme décimal de la valeur contenue dans le [champ]

Lower(chaine)

transforme tous les caractères de 'chaîne' en minuscules

exemples:

Lower("CHAT") retourne "chat"

Lower(A1) retourne la chaîne contenue dans la cellule A1, avec tous ses caractères convertis en minuscules

Appleworks GS

Lower([champ]) retourne la chaîne contenue dans le [champ] avec tous ses caractères convertis en minuscules

Max(liste num)

retourne la valeur maxi contenue dans la liste numérique. Les éléments de la liste peuvent contenir des zones de cellules.

exemples:

Max(2,10,4,6) retourne 10

Max(A1..A10) retourne la valeur maxi contenue dans une cellule de la zone A1..A10 en ignorant les cellules vides ou contenant des chaînes

Max(2,3,A1..A5,89) retourne la valeur maxi trouvée dans la liste

Max([champ1],[champ2],...) retourne la valeur maxi trouvée dans la liste des champs

Min(liste num)

retourne la valeur mini contenue dans la liste numérique. Les éléments de la liste peuvent contenir des zones de cellules

exemples:

Min(2,10,4,6) retourne 2

Min(A1..A10) retourne la valeur mini contenue dans une cellule de la zone A1..A10 en ignorant les cellules vides ou contenant des chaînes

Min(2,3,A1..A5,89) retourne la valeur mini trouvée dans la liste

Min([champ1],[champ2],...) retourne la valeur mini trouvée dans la liste des champs

Mod(nb1,nb2)

retourne nb1 modulo nb2 (le reste de la division de 'nombre1' par 'nombre2')

exemples:

Mod(10,3) retourne la valeur numérique 1

Mod(3,0) retourne la valeur ERROR

Mod(A1,A2) retourne le reste de la division du contenu numérique de la cellule A1 par le contenu numérique de la cellule A2

Mod([champ1],[champ2]) retourne le reste de la division du contenu numérique du [champ1] par le contenu numérique du [champ2]

Month(chaine)

retourne une chaîne exprimant le mois de la date constituée par 'chaîne'. La chaîne peut être exprimée sous la forme: mm/jj/aa ou mm/jj/aaaa ou Mois. jj/aaaa ou Mois jj/aaaa

exemples:

Month("Jan. 24,1988") retourne "Jan."

Month("2/1/89") retourne "Feb."

Month(A1) retourne le mois de la date exprimée dans la cellule A1

Month([champ]) retourne le mois de la date exprimée dans le [champ]

NA

retourne la valeur N/A

Not(expression)

retourne la valeur 1 (VRAI) si 'expression' est FAUX (si elle retourne la valeur 0)

exemples:

Not(0) retourne la valeur 1

Not(1) retourne la valeur 0

Not(10) retourne la valeur 0

Not(A1) retourne 1 si la valeur dans la cellule A1 est 0;
autrement retourne 0

Not([champ]) retourne 1 si la valeur dans le [champ] est 0;
autrement retourne 0

NPV(intérêt,zone numérique)

calcule la valeur actuelle de futurs remboursements (zone).

'Intérêt' représente un taux d'escompte exprimé en % ou en valeur décimale (ex: 10% ou 0.10)

'Zone numérique' représente le montant de l'investissement est la liste des futurs crédits et débits. Les crédits sont des valeurs positives, les débits étant négatifs. Les débits futurs sont supposés se produire à intervalles réguliers et le premier crédit arrivant à la fin du premier intervalle.

exemples:

NPV(12%,A1..A10) définit la valeur nette actuelle de trois futurs débits de la valeur contenue dans la zone A1 à A10 à un taux constant de 12%

Null

retourne une chaîne vide ("")

exemples:

Null retourne la chaîne vide ""

NumDate(chaine)

retourne une valeur entière représentant la date exprimée dans 'chaîne'. Jan. 1, 1900 correspond à la valeur 1; Jan. 2, 1900 à la valeur 2 et ainsi de suite. La chaîne peut être exprimée de différentes manières: mm/jj/aa ou mm/jj/aaaa ou Mois. jj/aaaa ou Mois jj/aaaa

exemples:

NumDate("Jan. 1, 1900") retourne la valeur 1

NumDate("Jan. 1, 1989") retourne la valeur 32508

NumDate(A1) retourne la valeur équivalente à la date exprimée dans la chaîne contenue dans la cellule A1

NumDate([champ]) retourne la valeur équivalente à la date exprimée dans la chaîne contenue dans le [champ]

Or(liste logique)

retourne la valeur numérique 1 (VRAI) si une des valeurs de la liste est VRAIE

exemples:

Or(1+1=2,2+254) retourne VRAI puisque le premier argument est vrai

Or(A1=B1, C1=D1) retourne VRAI si la valeur contenue dans la cellule A1 est égale à la valeur contenue dans la cellule B1 ou si la valeur contenue dans la cellule C1 est égale à la valeur contenue dans la cellule D1

Pi

retourne une approximation de la constante mathématique PI. Elle retourne 3.14159... avec le nombre de décimales spécifié dans l'application.

exemples:

Pi retourne 3.1459... avec le nombre de décimales spécifié dans le format de la cellule ou du [champ]

Pmt(pv,intérêt,durée)

calcule la valeur actuelle d'une série de remboursements identiques pour rembourser un investissement sur un nombre déterminé de mensualités.

'Pv' est la valeur actuelle du capital exprimé par une valeur positive avec optionnellement \$ en en-tête

'Intérêt' est le taux exprimé en % ou valeur décimale (ex: 10% ou 0.10)

'Durée' est le nombre de remboursements

exemples:

Pmt(10000,10.5%/12,240) retourne les mensualités pour rembourser 10000\$ empruntés sur 240 mois à un taux annuel de 10.5% (la durée étant exprimée en mois, l'intérêt doit l'être également, d'où la division par 12)

Position(ch1,ch2)

retourne un nombre correspondant à la première position de 'chaine1' dans 'chaine2'

exemples:

Position("d","abcde") retourne la valeur 4 car "d" est en 4ème position dans la chaîne "abcde"

Position("amitiés","Mes amitiés") retourne 5

Position("pa","papa est parti") retourne 1

Position(A1,A2) retourne la position de la chaîne contenue dans la cellule A1 par rapport à la chaîne contenue dans la cellule A2

Position([champ1],[champ2]) retourne la position de la chaîne contenue dans le [champ1] par rapport à la chaîne contenue dans le [champ2]

Appleworks GS

Product(liste num)

retourne le produit de tous les nombres de la liste

exemples:

Product(2,3,4) retourne 24

Product(A1..A10) retourne le produit des valeurs contenues dans les cellules A1 à A10 (en ignorant les cellules vides ou contenant une chaîne)

Product([champ1],[champ2],...) retourne le produit des valeurs contenues dans les champs 1,2... (en ignorant les cellules vides ou contenant une chaîne)

Proper(chaine)

retourne une chaîne avec le premier caractère converti en majuscules et les autres en minuscules

exemples:

Proper("capslock") retourne "Capslock"

Proper(A1) retourne la chaîne trouvée dans la cellule A1 avec le premier caractère converti en majuscule et le reste en minuscules

Proper([champ]) retourne la chaîne trouvée dans le [champ] avec le premier caractère converti en majuscule et le reste en minuscules

PV(remboursement,taux,durée)

calcule la valeur actuelle d'une série de remboursements identiques à période fixe ou la rémunération d'un investissement. 'Remboursement' est le montant de l'épargne ou des traites périodiques exprimé en valeur positive précédé optionnellement d'un \$

'Taux' est le montant de l'intérêt exprimé en % ou par une valeur décimale (ex: 10% ou 0.10)

'Durée' est le nombre de remboursements

exemples:

PV(10000,8%,10) retourne la valeur actuelle d'un capital de 10000\$ investi sur 10 ans à un taux annuel de rémunération de 8%

Radians(nombre)

retourne la valeur en radians de 'nombre' exprimé en degrés.

exemples:

Radians(180) retourne la valeur 3.14 (approximation de pi)

Random(nombre)

retourne un nombre aléatoire compris entre 0 et 'nombre'. Si l'argument est 0 ou 1, la valeur aléatoire générée est plus grande ou égale à 0 et plus petite que 1

exemples:

Appleworks GS

exemples:

Random(50) retourne une valeur aléatoire comprise entre 0 et 49, bornes comprises
Random(0) ou Random(1) retournent une valeur plus grande ou égale à 0 et inférieure à 1
Random(A1) retourne une valeur comprise entre 0 et la valeur de la cellule A1 diminuée de 1
Random([champ]) retourne une valeur comprise entre 0 et la valeur du [champ], diminuée de 1

Rate(fv,pv,durée)

Calcule le taux d'intérêt d'un investissement constant, à épargne régulière ou à versement unique.

'Fv' est la future valeur de l'emprunt, exprimé en valeur positive avec un \$ optionnel en en-tête

'Pv' est la valeur présente du capital, exprimé en valeur positive avec un \$ optionnel en en-tête

'Durée' est le nombre de périodes du remboursement ou du placement

exemples:

Rate(10000,2000,5) retourne le taux d'intérêt d'une somme placée sur 5 ans et qui donnera 10000\$ pour un capital initial de 2000\$

Round(nb1,nb2)

retourne le nombre1 arrondi au nombre de décimales spécifiées par le nombre2

exemples:

Round(3.14,1) retourne 3.1

Round(2.789,2) retourne 2.79

Round(1.5,0) retourne 2

Round(A1,1) retourne la valeur de la cellule A1 arrondie avec 1 décimale

Round([champ],1) retourne la valeur du [champ] arrondie avec 1 décimale

Sgn(nombre)

retourne le signe de 'nombre'. Si 'nombre' est positif, la valeur retournée est 1; s'il est négatif, c'est -1; et s'il est nul, c'est 0

exemples:

Sgn(2) retourne 1

Sgn(-2) retourne -1

Sgn(0) retourne 0

Sgn(A1) retourne le signe de la valeur placée dans la cellule A1

Sgn([champ]) retourne le signe de la valeur placée dans le [champ]

Sin(nombre)

retourne le sinus de 'nombre'. La fonction suppose que 'nombre'

Appleworks GS

est exprimé en radians

exemples:

Sin(0) retourne 0

Sin(Pi) retourne 1

Sin(A1) retourne le sinus de la valeur contenue dans A1

Sin([champ]) retourne le sinus de la valeur contenue dans le [champ]

Sqrt(nombre)

retourne la racine carée de 'nombre'. Si 'nombre' est négatif, la fonction retourne ERROR

exemples:

Sqrt(16) retourne 4

Sqrt(0) retourne 0

Sqrt(-1) retourne ERROR

Sqrt(A1) retourne la racine carrée de la valeur de la cellule A1

Sqrt([champ]) retourne la racine carrée de la valeur du [champ]

Stdv(liste num)

retourne l'écart type des valeurs de la liste numérique

exemples:

Stdv(2,7,9,10,27) retourne 9.46

Stdv(A1..A10) retourne l'écart type des valeurs des cellules de la zone A1..A10 en ignorant les cellules vides ou contenant des chaînes

Stdv([champ1],[champ2],...) retourne l'écart type des valeurs des champs indiqués en ignorant les champs vides ou contenant des chaînes

String(nombre)

retourne une chaîne constituée des chiffres de 'nombre'. Cette fonction sert à convertir un nombre en chaîne afin de le manipuler avec d'autres fonctions chaînes.

exemples:

String(43) retourne la chaîne "43"

String(999) retourne la chaîne "999"

String(A1) retourne une chaîne constituée de la valeur numérique de la cellule A1

String([champ]) retourne une chaîne constituée de la valeur numérique du [champ]

SubStr(chaine,début,longueur)

retourne une partie de 'chaîne' définie par les deux valeurs numériques.

Le premier argument est une chaîne d'où on va extraire la sous-chaîne.

'Début' est la position dans la chaîne d'où on va extraire la sous-chaîne

'Longueur' est la longueur de la sous-chaîne

Appleworks GS

exemples:

SubsStr("Mes amitiés",5,7) retourne "amitiés"
SubsStr("abcdefg",3,2) retourne "cd"
SubsStr(A1,3,5) retourne la sous-chaine définie à partir du 3ème caractère de la chaine contenue dans la cellule A1 et ayant 5 caractères de long
SubsStr([champ],3,5) retourne la sous-chaine définie à partir du 3ème caractère de la chaine contenue dans le [champ] et ayant 5 caractères de long

Sum(liste num)

retourne la somme des valeurs de la liste numérique

exemples:

Sum(1,3,5) retourne 9
Sum(A1..A10) retourne la somme des valeurs contenues dans les cellules A1 à A10 (en ignorant les cellules vides)
Sum([champ1],[champ2],...) retourne la somme des valeurs contenues dans les champs spécifiés

Tan(nombre)

retourne la tangente de 'nombre'. La fonction suppose que 'nombre' est exprimé en radians.

exemples:

Tan(Pi/4) retourne 1
Tan(A1) retourne la tangente de la valeur contenue dans A1
Tan([champ]) retourne la tangente de la valeur contenue dans le [champ]

Term(remboursement, taux, fv)

retourne le nombre de remboursements pour amortir un prêt à mensualités identiques.
'Remboursement' est le montant des mensualités exprimé par une valeur positive précédée optionnellement de \$
'Taux' est le montant des intérêts que l'on exprime en % ou par valeur décimale (ex: 10% ou 0.10)
'Fv' est la future valeur du capital exprimé par une valeur positive précédée optionnellement de \$

exemples:

Term(1000,8%,50000) retourne le nombre d'annuités pour rembourser un emprunt de 50000\$ avec 8% d'intérêt annuel et des annuités de 1000\$

Today

retourne une chaine représentant la date du jour (cette date est fournie par l'horloge interne du GS)

exemples:

Today retourne "Jan. 23, 1990" par exemple

True

retourne la valeur numérique 1 (VRAI)

exemples:

True retourne la valeur 1

Upper(chaine)

transforme tous les caractères de 'chaine' en majuscules

exemples:

Upper("capslock") retourne "CAPSLOCK"

Upper(A1) retourne la chaine trouvée en A1 et dont tous les caractères sont transformés en majuscules

Upper([champ]) retourne la chaine trouvée dans le [champ] et dont tous les caractères sont transformés en majuscules

Val(chaine)

retourne la valeur numérique de 'chaine'. Cette fonction sert à convertir un nombre, exprimé sous forme de chaine, en valeur numérique.

exemples:

Val("43") retourne la valeur numérique 43

Val("999") retourne la valeur numérique 999

Val(A1) retourne la valeur numérique de la chaine se trouvant dans la cellule A1

Val([champ]) retourne la valeur numérique de la chaine se trouvant dans le [champ]

Var(liste num)

retourne la variance des nombres de la liste numérique. La fonction suppose que les valeurs de la liste sont des échantillons et non la population totale. La fonction calcule la variance de la population. Si la liste représente la population totale plutôt qu'un échantillon, on peut calculer la variance de la population globale en mettant la moyenne de l'exemple dans la liste.

exemples:

Var(A1..A20) calcule la variance des valeurs situées dans les cellules A1 à A20

VLookup(valeur,zone,distance)

cherche 'valeur' dans une table contenue dans la 'zone' du tableur. Elle trouve la valeur immédiatement plus petite ou égale à 'valeur' dans la première colonne de 'zone' et retourne alors la valeur trouvée 'distance' colonnes plus à droite (on suppose que la première colonne de 'zone' est triée en ordre croissant)

exemples:

Appleworks GS

Vlookup(2,A1..A4,3) retourne la valeur trouvée selon la méthode suivante: trouve la valeur V la plus proche de 2 (inférieure ou égale) dans les colonnes A1 à A4 puis retourne la valeur située dans la cellules 3 colonnes plus à droite de la cellule contenant V

Weekday(chaine)

retourne une chaine représentant le nom du jour de la date exprimée dans 'chaine'

exemples:

Weekday("Jan. 24, 1988") retourne "Monday"

Weekday("June 21, 1988") retourne "Tuesday"

Weekday(A1) retourne le nom du jour correspondant à la date exprimée dans la cellule A1

Weekday([champ]) retourne le nom du jour correspondant à la date exprimée dans le [champ]

Year(chaine)

retourne un nombre représentant l'année de la date exprimée par 'chaine'

exemples:

Year("Jan. 24, 1988") retourne 1988

Year("Jan. 1, 1989") retourne 1989

Year(A1) retourne l'année correspondant à la date exprimée dans la cellule A1

Year([champ]) retourne l'année correspondant à la date exprimée dans le [champ]

LES FORMULES:

Introduction:

On peut placer une formule dans toute cellule du tableur ou dans tout champ de la base de données. Une formule permet de calculer une nouvelle valeur à partir d'autres valeurs.

Une formule dans un tableur peut calculer une valeur à partir de constantes, de références à d'autres cellules ou de valeurs obtenues par des expressions calculées. On peut aussi utiliser toute fonction d'Appleworks GS dans une formule.

La base de données permet de placer une formule dans un champ calculé. Une formule permet de calculer une valeur à partir de références à d'autres champs, de constantes ou d'expressions calculées. On peut aussi utiliser toute fonction d'Appleworks GS dans une formule (excepté les fonctions faisant appel à des zones du tableur ainsi que NPV, IRR, HLookup et VLookup).

Comme un champ calculé est déterminé par des valeurs propres à chaque fiche, le contenu du champ sera particulier à chaque fiche.

Eléments de la formule:

Les formules dans le tableur et la base de données peuvent

Appleworks GS

contenir différents éléments. Ces éléments sont des constantes, des opérateurs, des références à des cellules ou à des champs, des fonctions Appleworks GS, des expressions formées par des combinaisons de ces éléments...

Chaque formule dans le tableur doit commencer par le signe égal (=). Lorsqu'on crée une formule dans le tableur, taper le signe égal comme premier éléments de la formule.

Plusieurs éléments vont suivre, tels que constantes, expressions et fonctions. Dans le tableur, une formule peut aussi contenir des références à d'autres cellules. Dans la base de données, la formule peut contenir des références à d'autres champs.

Les constantes:

Une constante est une valeur qui ne change jamais. Ce peut être un nombre ou une chaîne. Certaines chaînes, comme des adresses, peuvent commencer par un nombre; si on inclut une constante chaîne dans une formule, la chaîne doit commencer et se terminer par des guillemets.

Exemples:

constantes numériques: 25, 100, 98.6, 1.024e6, \$100, 10%
constantes chaînes : "Robert", "Susan", "5250 Gulfon"

Références à d'autres cellules:

Une formule du tableur peut comprendre des références à des valeurs se trouvant dans d'autres cellules. Ces références sont soit des références absolues, soit des références relatives. Lorsqu'on crée une formule dans une cellule, les références absolues ou relatives à d'autres cellules produisent le même résultat. Cependant, si vous copiez la formule dans une autre cellule, la formule est interprétée différemment selon qu'il s'agit de références absolues ou relatives.

Références absolues:

Une référence absolue se réfère toujours à la même adresse, même si la formule est copiée à un autre endroit. Par exemple, si une formule dans la cellule B1 se réfère à la cellule A1 de manière absolue, la formule se réfèrera toujours à A1 même si la formule est copiée plus tard dans d'autres cellules.

Références relatives:

Une formule peut contenir des références relatives à d'autres cellules. Une formule avec des références relatives se réfèrera à différentes adresses de cellules selon la localisation de la formule.

Dans sa cellule d'origine, une formule avec une référence relative se réfère à une cellule bien précise (par exemple elle peut se référer à la cellule située sur la même ligne, mais deux colonnes plus à gauche).

Lorsqu'on copie cette formule dans une autre cellule, la formule se réfère alors à une nouvelle cellule (dans notre exemple, elle se réfèrera à la cellule située sur la même ligne et deux colonnes à gauche de la cellule où la formule vient d'être copiée).

Appleworks GS

Par exemple, si vous tapez la formule "=A1" dans la cellule B1, la formule place le contenu de A1 dans B1; et la formule se réfère relativement à la cellule A1 (référence relative sur la même ligne, une colonne à gauche).

Si on copie la formule contenant la référence relative dans une autre cellule que B1, cette formule ne se réfèrera plus à A1. En fait, elle va alors se référer à la cellule située sur la même ligne, une colonne à gauche.

Créer des références relatives/absolues:

La majorité des références que vous utiliserez dans le tableur seront des références relatives. Pour se référer relativement à une cellule, il suffit de taper les coordonnées de la cellule.

Par exemple, si on tape la formule "=A1" dans la cellule B1, la référence à A1 est relative: référence à la cellule située sur la même ligne, une colonne à gauche.

Si on copie la formule contenant la référence relative dans une autre cellule que B1, cette formule ne se réfèrera plus à A1. En fait, elle va alors se référer à la cellule située sur la même ligne, une colonne à gauche.

Lorsqu'on veut se référer à des cellules de manière absolue, la référence ne change pas si on copie la formule dans une autre cellule. Pour créer une référence absolue, mettez le signe "\$" devant l'élément absolu de la coordonnée à laquelle on se réfère. Par exemple, si on tape la formule "=\$A\$1" dans la cellule B1, la formule se réfèrera toujours à la cellule A1 quelquesoit l'emplacement de la cellule contenant la formule.

Si on met le signe "\$" devant les deux coordonnées (ligne,colonne), la référence se fait toujours sur la même cellule même si on copie la formule dans une autre cellule.

Mélanger des références absolues et relatives:

Une référence peut contenir un mélange des deux types de références. On peut en effet ne mettre le signe \$ que devant un des éléments de la référence: par exemple, on peut écrire \$A1.

Si cette référence mixte est copiée, la nouvelle référence se fera comme suit:

- dans la même colonne que la référence originale
- dans la ligne située à la même distance relative de la nouvelle cellule que l'était la cellule de référence par rapport à la cellule d'origine

Une référence peut aussi contenir le signe \$ uniquement devant la ligne de coordonnées: par exemple A\$1. Dans ce cas si cette référence mixte est copiée, la nouvelle référence se fera comme suit:

- dans la même ligne que la référence originale
- dans la colonne située à la même distance relative de la nouvelle cellule que l'était la cellule de référence par rapport à la cellule d'origine

En résumé: la référence à une cellule peut être relative, absolue

Appleworks GS

ou mixte. Pour illustrer chaque type de référence, voici un tableau des différentes possibilités avec les résultats produits sur la référence lors de la copie de la formule depuis B1 vers une autre cellule:

formule	résultat lors de la copie vers une autre cellule
=A1	référence à la cellule dans la même ligne, une colonne à gauche de la nouvelle cellule
=\$A\$1	référence à la cellule A1 quelquesoit la localisation de la formule
=\$A1	référence à la cellule sur la même ligne que la nouvelle cellule, et dans la colonne A
=A\$1	référence à la cellule sur la ligne 1 et une colonne à gauche de la nouvelle cellule

Références à d'autres champs:

Lorsqu'on crée une formule dans un champ calculé de la base de données, on procède de la même manière que pour les formules du tableur. La formule doit commencer par le signe = et contient des constantes, expressions et fonctions.

De plus, la formule peut contenir des références à d'autres valeurs situées dans d'autres champs. Pour inclure la valeur dans un autre champ, vous pouvez cliquer sur le nom du champ concerné dans le dialogue Field Formula.

Lorsque vous cliquez sur le nom d'un champ, ce champ est automatiquement inséré dans la formule à l'endroit du curseur d'insertion.

On peut aussi taper directement le nom du champ dans la formule. lorsqu'on tape le nom d'un champ dans une formule, celui-ci doit être compris entre crochets *]. Par exemple: =[champ1] + [champ2]

Expressions:

Une expression est un élément permettant d'évaluer une valeur. Une expression peut être un nombre, ou une formule arithmétique complexe. Par exemple:

```
43
43+24
(43+24)/(879.43*34.9)
```

On peut combiner des parties de l'expression avec des opérateurs arithmétiques, des opérateurs de comparaison ou des opérateurs logiques.

Opérateurs arithmétiques:

Les éléments numériques d'une expression peuvent être combinés avec des opérateurs arithmétiques. Un opérateur effectue une opération sur une ou plusieurs valeurs appelées opérands. Par exemple, l'opérateur d'addition (+) ajoute deux opérands entre eux

Appleworks GS

Voici la liste des opérateurs disponibles dans Appleworks GS:

symbole	nom	exemple	opération
-	moins unaire	- opérande	inverse le signe
+	plus unaire	+ opérande	conserve le signe
acc.circ.	exponentiation	op1 acc.circ op2	élève op1 à puissance op2
*	multiplication	op1*op2	multiplie op1 par op2
/	division	op1/op2	divise op1 par op2
+	addition	op1+op2	ajoute op1 et op2
-	soustraction	op1-op2	soustrait op2 de op1

Les opérateurs sont listés par ordre décroissant (de haut en bas) de priorité. On peut utiliser des parenthèses pour regrouper des éléments ou une expression afin de modifier l'ordre des priorités.

Opérateurs de comparaison:

Les expressions logiques peuvent aussi contenir des opérateurs de comparaison et des opérateurs logiques. Par exemple, la fonction IF utilise une expression logique pour décider quelle action accomplir, et l'expression logique contient souvent un opérateur de comparaison.

Voici deux exemples d'expressions logiques contenant des opérateurs de comparaison:

exemple	résultat
If(A1=A2,A3,A4)	teste si la valeur de la cellule A1 est égale à la valeur de la cellule A2. Si les valeurs sont égales, la fonction retourne la valeur contenue dans A3; si les valeurs sont différentes, la fonction retourne la valeur contenue dans A4.
If([champ1]=[champ2],1,0)	teste si la valeur dans le champ1 est égale à la valeur située dans le champ2. Si les valeurs sont égales, la fonction retourne 1; si les valeurs sont différentes, la fonction retourne 0

Voici la liste des opérateurs de comparaison:

symbole	nom	exemple	opération
=	égale	If(op1=op2...	teste si la valeur de op1 est égale à la valeur de op2
<>	différent	If(op1<>op2...	teste si la valeur de op1 est différente de la valeur de op2

Appleworks GS

>	plus grand	If(op1>op2...	teste si la valeur de op1 est plus grande que la valeur de op2
>=	plus grand ou égal	If(op1>=op2...	teste si la valeur de op1 est plus grande ou égale à la valeur de op2
<	plus petit	If(op1<op2...	teste si la valeur de op1 est plus petite que la valeur de op2
<=	plus petit ou égal	If(op1<=op2...	teste si la valeur de op1 est plus petite ou égale à la valeur de op2

On peut remplacer le symbole => au symbole >; et =< au symbole <; on peut aussi remplacer <> par le symbole x.

Les opérateurs de comparaison de cette table sont moins prioritaires que les opérateurs numériques.

Opérateurs logiques:

Les expressions logiques peuvent également contenir des opérateurs logiques. Un opérateur logique effectue une opération sur des opérandes ayant leurs valeurs VRAIES ou FAUSSES.

Voici deux exemples d'expressions logiques contenant des opérateurs logiques:

exemple	résultat
If(A1&A2,A3,A4)	teste si les valeurs logiques dans les cellules A1 et A2 sont toutes deux vraies (non nulles). Si elles sont toutes deux vraies, la fonction retourne la valeur de la cellule A3; si ces valeurs sont différentes, elle retourne la valeur contenue dans A4.
If([champ1]&[champ2],1,0)	teste si les valeurs dans les champ1 et 2 sont toutes deux vraies. La fonction retourne VRAI si elles sont toutes deux vraies; la fonction retourne FAUX si les valeurs sont différentes.

Voici la liste des opérateurs logiques:

symbole	nom	exemple	opération
&	ET logique	If(op1&op2...	teste si les deux opérandes op1 et op2 sont vrais. La fonction retourne VRAI si les deux sont vraies; autrement elle retourne FAUX

Appleworks GS

ù	OU logique	If(op1ùop2...	teste si une valeur des deux opérandes est vraie. La fonction retourne VRAI si une des deux est vraie; autrement elle retourne FAUX
trema	Non	trema opérande	inverse la valeur logique de l'opérande

Les ordres de priorité:

Les ordres de priorité des opérateurs définit l'ordre dans lequel les opérations sont effectuées lors de l'évaluation de l'expression. Par exemple, dans Appleworks GS, l'opérateur multiplicateur (*) est prioritaire sur l'opérateur d'addition binaire (+). Ceci signifie que les multiplications sont effectuées AVANT les additions.

Par exemple avec $4+5*3$: il y a deux valeurs possibles pour cette expression. Si l'opérateur d'addition est prioritaire sur l'opérateur de multiplication, on doit effectuer d'abord l'addition puis la multiplication et on a donc $9*3$ soit 27. Mais comme dans Appleworks GS c'est l'opérateur de multiplication qui est prioritaire sur l'opérateur d'addition, on va donc effectuer la multiplication avant l'addition et on obtient $4+15$ soit 19.

Voici la table des ordres de priorité (ordre décroissant):

trema - +	opérateurs unaires, utilisés avant un opérande
accent circonflexe	
* /	
+ -	opérateurs binaires, utilisés entre deux opérandes
< > <= >=	
= <>	
&	
ù	

Les fonctions dans les expressions:

En plus des constantes, des valeurs des cellules ou des champs, une expression peut contenir des fonctions Appleworks GS. Voici des exemples d'expressions contenant des fonctions:

```

=43 / Log(4.3)
=43 * Sin(6.7)
=A1 * Cos(A2)
=[champ1] + Sqrt([champ2])
    
```