

IF 関数

IF 関数は、特定の条件によって返す値を変えたい(つまり条件に合うかどうかの結果で次の処理方法をコントロールしたい)時に使う関数です。

様式: =IF(条件式,A,B)

『もし、この条件式(論理式と呼ばれます)が成立する(True)なら A、
成立しない時(False)は B にする』

ここで、A や B は処理の仕方を示しているもので、具体的には数値や文字列であったり、値などを保持しているセルを参照するものであったり、数式や条件式であったりと、いろいろです。数式や条件式にはもちろん関数が含まれることがあります。

IF 関数は便利な関数で、他の関数と組み合わせたり、IF 関数の中に IF 関数を入れたりして、複雑な数式の組み立てができます。

Excel が好きになるかどうかの分かれ道が IF 関数かもしれません。頑張って理解し、使いこなしてください。

条件式に使う記号(比較演算子)

条件式(論理式)で使う記号(比較演算子といいます。)は、以下の 6 通りです。
等しくない(Not Equal)には、<> を用いることが意外に知られていません。
(Excel では不等号(≠)は使えません。)

比較の対象には文字列も可能です。比較する文字列はダブルクォーテーション(")で囲みます。

文字列の比較は、=(等しい)か、<>(等しくない)だけが使われます。

A=B	A と B が等しい
A>B	A が B よりも大きい
A<B	A が B よりも小さい
A>=B	A が B 以上(大きいか等しい)
A<=B	A が B 以下(小さいか等しい)
A<>B	A と B が等しくない

IF 関数の使用例

IF 関数の具体的な使用例を見ていきましょう。

例 1: `=IF(B11="", "", B11/F1)`

セル B11 が空白であればこのセルを空白とし、セル B11 が空白でなければセル B11 の値をセル \$F\$1 の値で割った数値を求める。

単に「`=B11/F1`」とすると、B11 のセルが空白の時は「0」という数字が表示されてしまいます。0 のデータばかり羅列されているような表は見づらいですね。例 1 のような形で「こちらのセルが空白の場合はこのセルも空白にする」という IF 関数を使用すれば見栄えがずっと良くなります。

例 2: `=IF(A1="", "", B1/A1)`

セル A1 が空白であればこのセルを空白とし、セル A1 が空白でなければセル B1 の値をセル A1 の値で割った数値を求める。

単に「`=B1/A1`」とすると、A1 のセルが空白の時は「`#DIV/0!`」(0 で割る)というエラーで表示されてしまいます。例 2 のように、あるセルが空白の場合はスキップ(空白で埋める)し、空白でない場合だけ処理をさせる IF 関数は、「`#DIV/0!`」とか「`#N/A!`」のようなエラー表示を避けるテクニックとして頻繁に使われます。

例 3: `=IF(OR(A1="", A1=0), "", B1/A1)`

セル A1 が空白または数値 0 であればこのセルを空白とし、セル A1 が空白でなければセル B1 の値をセル A1 の値で割った数値を求める。

例 2 では、セル A1 が数値 0 であるとやはり「`#DIV/0!`」になってしまいますので、改良してみました。

お分かりのように、IF 関数の中に、OR 関数(指定したいいずれかの条件が真の場合に真を実行する関数)が入れ子になっています。このように関数の引数として関数を使うことをネストと言います。

例 4: `=IF(AND(B2>=F2, C2<=G2), "目標達成", "未達成")`

セル B2 の値がセル F2 の値より上回るか等しく、かつ、セル C2 の値がセル G2 より下回るか等しい場合に「目標達成」、さもなければ「未達成」の文字をこのセルに埋める。

AND 関数と組み合わせた例です。

B 列: 売上実績、C 列: 費用実績、F 列: 売上目標、G 列: 費用予算のような表をイメージして、この式を解釈してみてください(^^)

【便利知識】

AND 関数の代わりに、論理積(論理演算子 *)を使うこともできます。

=IF((B2>=F2)*(C2<=G2),"目標達成","未達成")

同様に、OR 関数の代わりに、論理和(論理演算子 +)を使うこともできます。

=IF((A1="")+ (A1=0),"",B1/A1)

例 5: =IF(B2="", "", IF(B2>=60, "合格", "不合格")

セル B2 が空白ならば空白にして、そうでなければ、60 点以上で合格、60 点未満で不合格にする。

IF 関数の中に IF 関数をネストした例です。

OR 関数と AND 関数

OR 関数は、いずれかの論理式が真ならば"TRUE"を、すべての論理式が F 偽ならば"FALSE"を返します。

AND 関数は、すべての論理式が真ならば"TRUE"を、いずれかの論理式が偽ならば"FALSE"を返します。

OR 関数、AND 関数は単独で使用した場合は、その結果が真の時"TRUE"、偽の時"FALSE"という文字が返されるのですが、通常は、上述の例 3、例 4 のように IF 関数と組み合わせてネストの形で使用されます。

【便利知識】

上述の 2 例では、IF 関数の引数として OR 関数、AND 関数を使っていますが、逆に OR 関数、AND 関数の引数として IF 関数をネストすることも少なくありません。

=OR(論理式 1,[論理式 2]··)

=AND(論理式 1,[論理式 2]··)

関数のネスト(入れ子)

例 6: =IF(B11="", "", AVERAGE(B4:B10))

セル B11 が空白であればこのセルも空白とし、空白でなければ B4 から B10 までのセルの値の平均値を求める。

このように AND や OR 関数に限らず、いろいろな関数をネストすることができます。

例 7: =IF(B11<40,"不可",IF(B11<60,"可",IF(B11<80,"良","優")))

IF 関数はそのままでは、条件に合致した場合(真の場合)と合致しない場合(偽の場合)の二通りしか設定できません。3 つ以上の条件設定を行う場合には IF 関数の中に IF 関数をネストすれば良いのです。

例 8: =ROUNDUP(AVERAGE(A1:A5),1)

A1 から A5 までのセルの平均値を切り上げて小数点以下 1 桁を表示する。

関数のネストは IF 関数に限りません。このように ROUNDUP(あるいは ROUNDDOWN)と AVERAGE を組み合わせるようなケースも多くあります。

ネストの階層

ネストされている関数がさらに別の関数をネストすることもできます。

Excel ではひとつの関数の中で最大 7 階層まで関数をネストさせることができますが、余りに階層の深い関数は本人ですらどのような処理をしているのかが解りづらくなり、後で修正する必要が生じた場合など苦勞することになります。**3~4 階層位までに留めた方が良いでしょう。**

【便利知識】

言葉で説明するのはなかなか大変なのですが、ネストが深くなることを避ける方法は比較的簡単です。いったんある段階までの処理を別のセルで行わせ、その処理結果を本来のセルで引数として用いれば良いのです。

例 9: =IF(AND(B2>=F2,C2<=G2),"目標達成","未達成")

セル K2 に「=AND(B2>=F2,C2<=G2)」と記述しておけば、
=IF(K2="TRUE","目標達成","未達成")と記述することと同じになります。