

IFS関数 使い方

マスターガイド



※当ガイドに従うことで、法令違反がないことを保証する資料ではありません。
※あくまで参考としてご利用いただくことを想定している資料です。実際の制度内容は国の資料等をご確認ください。

IFS関数とIF関数の「使い分け」

IFS関数とIF関数の「使い分け」

IFS関数は、IF関数の弱点だった「3つ以上の条件分岐（ネスト）」を解消するために生まれました。まずは、どちらを使うべきか整理しましょう。

関数	特徴	主なシナリオ
IF関数	2択の分岐に強い	「合格 / 不合格」 「○ / ×」 「達成 / 未達」
IFS関数	3択以上の多分岐に強い	「A / B / C / D」 「優 / 良 / 可 / 不可」 「松 / 竹 / 梅」



IFS関数は「もしAなら1、もしBなら2、もしCなら3…」と、条件と結果のペアを並べていくだけで、複雑な分岐をシンプルに記述できます。

実践①「ネストIF」を「IFS」に書き換える

実践①「ネストIF」を「IFS」に書き換える

IFS関数の最大のメリットは、`=IF(..., IF(..., IF(..., )))`のような複雑なネスト（入れ子）を一掃できることです。

シナリオ： B列の点数に応じて、C列に「A, B, C, D」の評価を付けたい。

従来の書き方（ネストIF）

```
=IF(B2>=80, "A", IF(B2>=60, "B", IF(B2>=30, "C", "D")))
```

→カッコの対応関係が複雑で、条件の追加・修正も困難…

IFS関数を使った書き方

```
=IFS(B2>=80, "A", B2>=60, "B", B2>=30, "C", B2<30, "D")
```

→「条件, 結果」のセットを並べるだけなので、直感的で分かりやすい！

実践②「それ以外」を必ず設定するテクニック

実践②「それ以外」を必ず設定するテクニック

IFS関数は、どの条件にも当てはまらないと #N/A エラーになってしまいます。
これを防ぐため、必ず最後に「それ以外のすべての条件」を入れるのがプロのテクニックです。

ポイント：最後の論理式に TRUE を使います。

例：B2セルが80点以上なら「A」、60点以上なら「B」、それ以外は「C」

```
=IFS(B2>=80, "A", B2>=60, "B", TRUE, "C")
```

解説：

1. 80点以上か？ → YESなら「A」
2. (80点未満で) 60点以上か？ → YESなら「B」
3. TRUE (これまでのどの条件も満たさなかった場合) → YESなので「C」

このように TRUE を最後に置くことで、IF関数の「偽の場合」と同じ役割を果たし、代替処理を指定することでエラーを確実に防げます。

実践③ 「AND関数」との組み合わせ

実践③「AND関数」との組み合わせ

IFS関数は、AND（かつ）やOR（または）と組み合わせることで、さらに複雑な実務に対応できます。

シナリオ： B列（売上）とC列（契約件数）の2つの条件でボーナス判定を自動化したい。

数式：

```
=IFS( AND(B2>=100, C2>=10), "Sランク", AND(B2>=80, C2>=5), "Aランク", AND(B2>=50, C2>=3), "Bランク", TRUE, "Cランク")
```

解説：

AND関数で複数の条件をまとめたものを、IFS関数の「論理式」の部分に入れるだけです。
ネストIFで書くと非常に困難だった条件分岐も、IFSならスッキリ記述できます。

【免責】

※当資料に従うことで、法令違反がないことを保証する資料ではありません。

※あくまで参考としてご活用いただくことを想定している資料です。また当資料は、公開日時点の内容となっております。最新の情報、実際の制度内容は国の資料等をご確認ください。