

TakiLibで、区切りの何番目・前後を取り出す

シャケモンとフナさんが、文字の取り方と「見つからないとき」を確かめる

Copyright © 2026 LWP 山中 一弘

本資料は、出典を明記いただければ、商用・非商用を問わず、ご自由に複製・改変・再配布していただけます。なお、著作権表示は改変せず、そのまま記載してご利用くださいますようお願いいたします。

ストーリー

シャケモンの会社・部署では、TakiLib(滝Lib)を共通の標準ライブラリーとして使っています。今回は、その文字列抽出の機能を調べる回です。コードの一部を取り出したいフナさんと、どの区切りを選ぶのか、結果が空欄だったらどう扱うのかを確かめます。

登場するtextBefore・textAfterはTakiLibの関数で、VBAの組込み関数ではありません。「社内標準」は、この会社で共通に採用しているという意味です。会話と場面は架空の説明例。対象はWindows版Excel VBAとTakiLib 3.5.4で、掲載条件の代表8例をExcelで確認しました。

1 今日は社内標準のTakiLibで

区切り文字を目印に、前か後ろの文字列を取り出します。まずは「どの区切りまで欲しいか」をそろえましょう。

1 今日は社内標準のTakiLibで 区切りの前後を取り出そう

 シャケモンの会社・部署では、TakiLib (滝Lib) を共通の標準ライブラリーとして使っています。

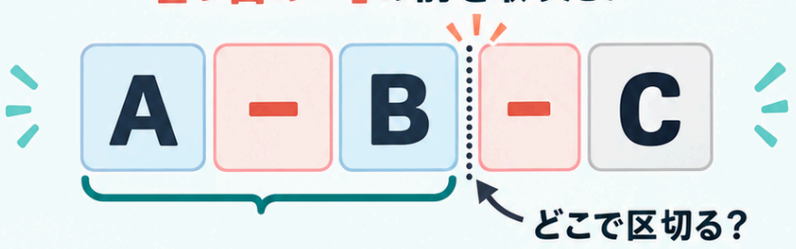
フナさん

コードの途中までを取り出したいんです。

シャケモン

今日はTakiLibの文字列抽出を調べましょう。

2つ目の「-」の前を取りたい



シャケモン

使うのは **textBefore** と **textAfter** です。

フナさん

うちで使っているTakiLibの関数ですね。

 **textBefore / textAfter** はTakiLibの関数です。VBAの組み込み関数ではありません。

図1 | 今回はTakiLibの文字列抽出を使う回。

2「2番目」は、2文字目ではない

出現番号は、区切り文字が何回現れたかを数えます。文字位置と区切りの番号を別に考えると、引数の意味がはっきりします。

2『2番目』は、2文字目ではない

? 2番目、って……文字を2つ数えるんですか?

ここでは『区切りが何回出たか』です。

文字位置	1	2	3	4	5
文字	A	-	B	-	C

1つ目 2つ目

なるほど。2つ目の区切りは、4文字目ですね。

その位置を見つけてから切り出します。

 **文字の位置 $\neq$ 区切りの出現番号** 

図2 | 2つ目の区切りは、文字位置4にある。

3 textBeforeで、前まで取る

TakiLibのtextBeforeに出現番号2を渡すと、この例ではA-Bを返します。選んだ区切り自体を含めず、その直前までを取り出します。

3 textBeforeで、前まで取る

左から2つ目の『-』を探します。

区切りの『-』そのものは、入らないんですね。

TakiLibの呼出し例

```
textBefore("A-B-C", "-", 2)
```

2つ目の『-』

A - B - C

1 2 3 4 5

取り出す範囲

戻り値

A-B

2は区切りの出現番号。文字数ではありません。

はい。指定した区切りの直前までです。

💡 textBeforeは、指定した区切りの前までの文字列を返します。

図3 | 出現番号2を、取得する文字数と取り違えない。

4 右から探して、後ろを取る

TakiLibのtextAfterで出現番号を-1にすると、右から最初の区切りを選びます。探索方向を変えても、返す文字列を逆順にするわけではありません。

4

右から探して、後ろを取る

使いこなそう
 便利な関数で
 業務をもっとラクに

探す向きが逆でも、文字は逆さまにならないんですね。



TakiLibの呼出し例

```
textAfter("A-B-C", "-", -1)
```

右から探す



右から1つ目

戻り値

C

-1なら、右から最初の区切りです。





取り出す文字の並びは、そのままです。



💡 探す向きを指定できます

正の数：左から

→

先頭から順に区切りを探します。

負の数：右から

←

末尾から逆に区切りを探します。

業務に合わせて、必要な部分だけを取り出し、集計や照合に活用できます。

図4 | 正の数は左から、負の数は右から数える。

5 空文字と、先頭区切りの注意点

末尾に区切りがあるABC-の後ろには、文字がありません。この場合のtextAfterの戻り値は空文字です。一方、TakiLib 3.5.4のtextBeforeでは、先頭が区切りの-ABCを渡すと、空文字にならず-ABC全体が返ることを実測しました。前後を対称と決めつけず、先頭・末尾も確認して使います。

5 空文字と、先頭区切りの注意点



結果が空欄なら、見つからなかったんですか？



見つかっていても、空文字は返ります。

✓

末尾の区切りの後

`textAfter("ABC-", "-")`
 → ""

区切りはある。後ろに文字がない。

⚠

先頭の区切りには注意

`textBefore("-ABC", "-")`
 → "-ABC"

TakiLib 3.5.4の実測：
空文字にならない

連続した区切りも、1つずつ数える

A
↑
1

--
↑
2

B



前と後で、同じように動くと思わない方がいいですね。



先頭・末尾の例も確かめて使いましょう。

図5 | 空文字と未検出を分け、現行版の境界動作も確かめる。

6 見つからないときは、別の結果

区切りがない場合だけでなく、指定した回数まで区切りが現れない場合も未検出です。代替値を省略すると、TakiLibはCVerErr(xlErrNA)によるError値を返します。CVerErrは標準VBAの関数です。戻ってきたError値は、実行時エラーで処理が停止することとは区別します。

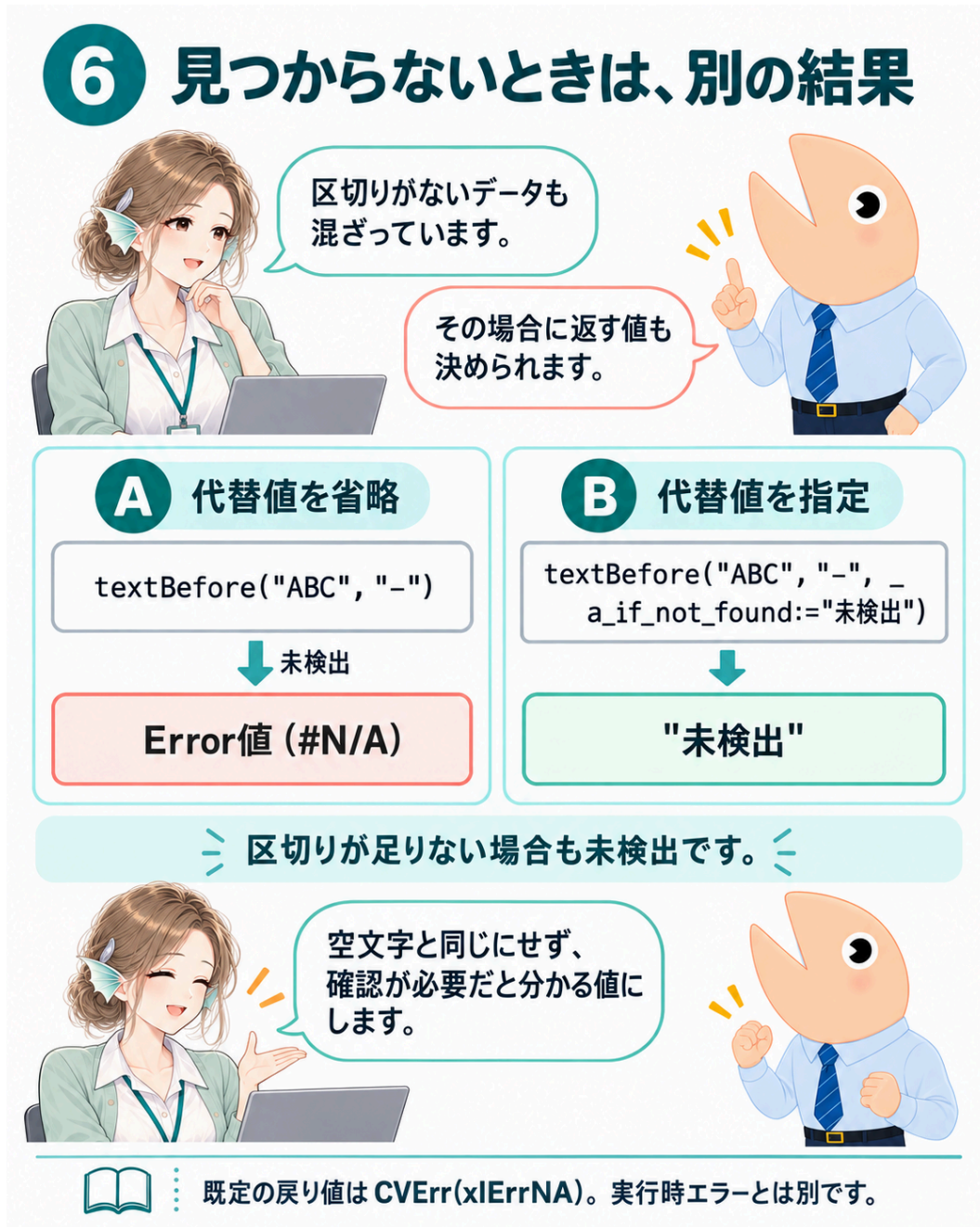


図6 | 確認が必要なデータを、正常な空文字へ紛れ込ませない。

7 取り方と、見つからない時をセットで

一文字の区切り、出現番号と方向、未検出の扱いを決めてから使います。空欄をどう解釈するかまで共有しておく、と、取り出した結果を次の照合へ渡しやすくなります。

7 取り方と、見つからない時をセットで



抽出の方法だけで、
終わりではないですね。



1

区切りは1文字

例：『-』

AA-001-BB

区切りに使うのは
1文字です。

2

何番目を、どちらから？

正の数は左から、負の数は右から

AA-001-BB

→ 正の数
(左から)

← 負の数
(右から)

3

未検出なら何を返す？

正常な空文字と区別する

例：「未検出」を返す

未検出の時に返す値を、
正常な空文字と
区別できるように
しましょう。

TakiLibの機能 + 業務での扱いを確認



手順の方は、
空欄にしませんから。



そこまでそうと、
引き継ぎも安心です。

図7 | 関数の使い方と、業務での結果の扱いをそろえる。

まとめ

TakiLibのtextBefore・textAfterは、一文字の区切りを何番目に見つけたかに応じて、その前後を取り出します。空文字、未検出、現行版の境界動作を分けて確かめることが、結果を業務で使うための要点です。

今回の仕掛け

両関数は、まずtextDelimiterIndexで区切りの文字位置を求め、次にtextMidで切り出します。出現番号は省略時1で、正なら左から、負なら右から探します。0は指定せず、区切りは一文字にします。Excelのワークシート関数TEXTBEFORE・TEXTAFTERの全仕様を再現するものではありません。

先頭区切りの注意点は、textBeforeがtextMidへ終了位置0を渡す経路にあります。現行実装のtextLength・mathCanonicalIndexでは、その0を文字列長へ正規化するため、-ABC全体を取り出します。図5はこの版での観察結果を示したもので、期待する仕様や改善済みの動作としては扱っていません。ライブラリー本体の変更は行っていません。

今回のソースコードのサンプル

サンプルコード TakiLibを直接呼び、取り方と境界を確かめる

```
01 Public Sub SampleTakiLibTextParts()
02     Debug.Print textBefore("A-B-C", "-", 2)
03     Debug.Print textAfter("A-B-C", "-", -1)
04     Debug.Print textBefore("A--B", "-", 2)
05     Debug.Print "[" & textAfter("ABC-", "-") & "]"
06     Debug.Print textBefore("-ABC", "-")
07
08     Dim result As Variant
09     result = textBefore("ABC", "-")
10     Debug.Print IsError(result)
11
12     result = textBefore("ABC", "-", _
13         a_if_not_found:="未検出")
14     Debug.Print result
15 End Sub
```

入力・準備

TakiLib 3.5.4の通常版taki3lib.basを導入したマクロ有効ブックを使います。上のSubは別の標準モジュールへ置き、VBEのイミディエイトウィンドウで出力を確認します。社内で導入済みでも、読者自身のブックに自動で備わる関数ではありません。

コードの説明

最初の3つのDebug.Printは、左から2番目・右から1番目・連続区切りの抽出です。角括弧は空文字を見分けるためにサンプル側で付けています。resultは文字列とError値の両方を受け取れるVariantとし、IsErrorで未検出のError値を確認します。a_if_not_foundは未検出時の代替値です。

結果

上からA-B、C、A-、[]、-ABC、True、未検出となります。[]は空文字を角括弧で囲んだ表示です。関数呼出しの代表条件は隔離したExcelで確認しました。代替値の限定検証ではNOT_FOUNDを指定し、このサンプルでは読者向けに未検出へ変更しています。

読むポイント

先頭区切りの-ABCは現行版の注意例です。未検出を空文字に置き換えると、本当に取り出す文字がなかった場合と区別しづらくなります。業務で区別が必要なら、衝突しない代替値かError値を使います。

サンプルコード textBeforeの原典抜粋(読解用)

```

01 Select Case xindex
02     Case Is = -1
03         If valueIsMissing(a_if_not_found) Then
04             xresult = CVErr(xlErrNA)
05         Else
06             valueSet(xresult) = a_if_not_found
07         End If
08     Case Is = 0
09         xresult = ""
10     Case Is = Len(a_text) + 1
11         xresult = textMid(a_text, 1, xindex - a_offset)
12     Case Else
13         xresult = textMid(a_text, 1, xindex - 1 - a_offset)
14 End Select

```

コードの説明

xindex=-1は未検出です。通常の区切りではCase Elseへ進み、区切り直前の位置まで切り出します。先頭の区切りはxindex=1なので、この分岐で終了位置0が生じます。xindex=0の分岐は、通常の「先頭に実在する区切り」とは別です。

サンプルコード textAfterの原典抜粋(読解用)

```

01 Select Case xindex
02     Case Is = -1
03         If valueIsMissing(a_if_not_found) Then
04             xresult = CVErr(xlErrNA)
05         Else
06             valueSet(xresult) = a_if_not_found
07         End If
08     Case Is = 0
09         xresult = textMid(a_text, 1 + a_offset)
10     Case Is = Len(a_text) + 1
11         xresult = ""
12     Case Else
13         xresult = textMid(a_text, xindex + 1 + a_offset)
14 End Select

```

コードの説明

未検出の扱いはtextBeforeと共通です。通常の区切りでは、その一文字後から取り出します。末尾の区切りなら開始位置は文字列長の一つ後となり、今回の例では空文字を返します。

****抜粋の範囲:**** 2つの抜粋はTakiLib 3.5.4の対象関数からコメントを除き、字下げを掲載用に整えた読解用の一部分です。xindexを求める前処理、既定値設定、戻り値の代入、依存関数は省略しています。単独で実行したり、導入済みライブラリーへ重複追加したりするコードではありません。仮想区切り・位置調整の詳細は今回の解説範囲外です。

出典メモ

- 元ネタ: TK-051「区切りの何番目・前後を取り出す」。元ネタは保全し、先頭区切りについては現行ソースと今回の実測に基づいて説明を補正しました。
- 実装: 公開版TakiLib 3.5.4、textBefore・textAfter・textDelimiterIndex・textMid・textLength・mathCanonicalIndex。通常CP932版taki3lib.basを隔離したExcelへ読み込み、8ケースを検証しました。UTF-8版は読解・照合専用です。
- [Microsoft Learn: CVErr関数](#) — ErrorサブタイプのVariantを返す標準VBA関数の説明。
- 図版: LWPの現行キャラクター共通資産と設定V009を参照し、この記事用に生成・確認。図中の文字列は仕組みを示す模式図です。TakiLibの全入力・全オプションを検証したものではありません。