

# マクロ割付関数仕様書

---

## 関数一覧

@MID <文字列範囲指定取得> .....	3
@LEFT / @RIGHT <文字列左右指定取得> .....	4
@LEN <文字数取得> .....	5
@FMT <フィルター変換> .....	6
@REP <置換> .....	7
@CNV <変換> .....	8
@INS <文字列挿入> .....	9
@REPT <文字列繰り返し> .....	10
@TRM <空白削除> .....	11
@NUM <数字のみに変換> .....	12
@ALR <右詰> .....	13
@SPL <文字列分割> .....	14
@CNT <文字列数取得> .....	15
@IDX <文字列検索> .....	16
@SEL <条件文字列選択> .....	17
@IF <条件文字列選択> .....	18
@ERR <エラー発生> .....	19
@NOW <現在日時取得> .....	20
@NOWADD <現在日時加算取得> .....	21
@WAR <割付名でデータ取得> .....	22
@FIL <フィールド名でデータ取得> .....	23
@LST <リストからデータ取得> .....	24
@TNG <単語からデータ取得> .....	25
@TAX <消費税計算> .....	26
@BCD <チェックデジット追加> .....	27
@GET <フィールドプロパティ取得> .....	28
@SET <フィールドプロパティ設定> .....	29

**■ 関数の書式**

関数は@で始まり、括弧 ( ) でくくる。

```
@MID(@WAR("品番"),1,2)+"-"+@MID(@WAR("品番"),3)
```

引数の渡し方は文字列を””で囲む必要あり

## @MID <文字列範囲指定取得>

### ■ 概要

指定された文字列を範囲指定し抜き出す

### ■ 関数名

@MID(arg1,arg2,[arg3])

arg1 対象となる文字列

arg2 先頭位置（1から）

arg3 桁数（省略すると最後まで）

### ■ 使用例

入力	結果
@MID("123456",1,6)	123456
@MID("123456",2,3)	234
@MID("123456",5)	56

## @LEFT / @RIGHT <文字列左右指定取得>

### ■ 概要

指定された文字列を左または右から指定文字数分抜き出す

### ■ 関数名

@LEFT(arg1,arg2) : 左から

@RIGHT(arg1,arg2) : 右から

※@LEFTB、@RIGHTB で文字数ではなくバイト数でカウントします。

arg1 対象となる文字列

arg2 文字数

### ■ 使用例

入力	結果
@LEFT("123456",3)	123
@RIGHT("123456",3)	456
@LEFTB("あいうえお",5)	あい

## @LEN <文字数取得>

### ■ 概要

指定された文字列の文字数を取得する。

### ■ 関数名

@LEN(arg1)

※@LENB でバイト数を取得します。

arg1 対象となる文字列

### ■ 使用例

入力	結果
@LEN("123")	3
@LEN("あいうえお")	5
@LENB("あいうえお")	10

## @FMT <フィルター変換>

### ■ 概要

指定された文字列をフィルターで変換する

フィルターの仕様は VB2005 の Format に準ずる

※旧@FLT

### ■ 関数名

@FMT(arg1,arg2)

arg1 対象となる文字列

arg2 フィルター文字列

### ■ 使用例

入力	結果
@FMT("123456","¥¥#,##0;;#")	¥123,456
@FMT("0","¥¥#,##0;;#")	

※フィルター式にカンマがあるので””で囲む

**@REP <置換>**

## ■ 概要

文字列の中の特定の文字列を置き換える

## ■ 関数名

@REP(arg1,arg2,arg3)

arg1 対象となる文字列

arg2 変換対象にしたい文字

arg3 変換後の文字

## ■ 使用例

入力	結果
@REP("123456","23","兄さん")	1 兄さん 456
@REP("123456","23","")	1456

**@CNV <変換>**

## ■ 概要

文字列の全角半角大文字小文字変換をする  
変換の仕様は VB2005 の StrConv に準ずる

## ■ 関数名

@CNV(arg1,arg2)

arg1 対象となる文字列

arg2 変換方法 Z=全角変換、H=半角変換、U=大文字変換、L=小文字変換

## ■ 使用例

入力	結果
@CNV("123456",Z)	1 2 3 4 5 6
@CNV("a b c",HU)	ABC
@CNV("あいう A B 漢字",HL)	アウ ab 漢字

## @INS <文字列挿入>

### ■ 概要

指定した位置に文字列を挿入する。

### ■ 関数名

@INS(arg1,arg2,arg3)

arg1 対象となる文字列

arg2 挿入する位置（1から）

arg3 挿入する文字列

### ■ 備考

args2 の値が args1 の桁数+1 を超えるとエラーになります。

### ■ 使用例

入力	結果
@INS("123456",4,"-")	123-456
@INS("123456",7,"-ABC")	123456-ABC

※わかりやすいように空白を△で表記しています

## @REPT <文字列繰り返し>

### ■ 概要

文字列を指定数分並べる。

### ■ 関数名

@REPT(arg1,arg2)

arg1 文字列

arg2 個数

### ■ 使用例

入力	結果
@REPT("1",5)	11111
@REPT("ab",3)	ababab

## @TRM <空白削除>

### ■ 概要

文字列の空白を削除する

### ■ 関数名

@TRM(arg1,[arg2])

arg1 対象となる文字列

arg2 削除方法（省略すると 0） 0=前後、1=先頭から、2=後ろから、3=文字列内すべて

### ■ 使用例

入力	結果
@TRM("△123△456△")	123△456
@TRM("△123△456△",1)	123△456△
@TRM("△123△456△",2)	△123△456
@TRM("△123△456△",3)	123456

※わかりやすいように空白を△で表記しています

## @NUM <数字のみに変換>

### ■ 概要

数字のみの文字列に変換する

### ■ 関数名

@NUM(arg1,[arg2])

arg1 対象となる文字列

arg2 対象外文字列 (省略可能)

### ■ 使用例

入力	結果
@NUM("123-456ABC")	123456
@NUM("123-456ABC","-")	123-456

**@ALR <右詰>**

## ■ 概要

文字列の右詰をする

## ■ 関数名

@ALR(arg1,arg2,[arg3])

arg1 対象となる文字列

arg2 桁数

arg3 詰める文字 (省略すると自動判定 対象文字列が半角なら半角空白)

## ■ 使用例

入力	結果
@ALR("123456",8)	△△123456
@ALR("123456",8,"0")	00123456

※わかりやすいように空白を△で表記しています

## @SPL <文字列分割>

### ■ 概要

文字列を指定した区切り文字で分割する

### ■ 関数名

@SPL(arg1,arg2,arg3)

arg1 対象となる文字列

arg2 区切り文字

arg3 何番目に区切られた文字列を取得するか 1～

### ■ 使用例

入力	結果
@SPL("ABC&XYZ", "&", 1)	ABC
@SPL("ABC△DEF△HIJ", "△", 3)	HIJ
@SPL("12345", "&", 1)	12345
@SPL("12345", "&", 2)	(文字なし)

※わかりやすいように空白を△で表記しています

## @CNT <文字列数取得>

### ■ 概要

文字列の数を取得します。

### ■ 関数名

@CNT(arg1,arg2)

arg1 対象となる文字列

arg2 検索する文字列 (/で囲むと正規表現)

return 文字数

### ■ 使用例

入力	説明	結果
@CNT("ABC123","BC")		1
@CNT("ABC123","Z")		0
@CNT("ABC123","/¥d/")	数字の数を取得	3

## @IDX <文字列検索>

### ■ 概要

文字列の位置を取得します。

### ■ 関数名

@IDX(arg1,arg2)

arg1 対象となる文字列

arg2 検索する文字列 (/で囲むと正規表現)

return 1以上=文字列の開始位置

0=文字列なし

### ■ 使用例

入力	説明	結果
@IDX("ABC123","BC")		2
@IDX("ABC123","Z")		0
@IDX("ABC123","/¥d/")	数字を検索	4

**@SEL <条件文字列選択>**

## ■ 概要

条件により文字列を選択する

## ■ 関数名

@SEL(arg1,arg2,arg3)

arg1 対象となる文字列

arg2 条件 (|で区切る)

arg3 返す文字列 (|で区切る)

## ■ 使用可能な比較演算子

= (等しい), >(より大きい), <(より小さい), >=(以上), <=(以下), <>(等しくない)

## ■ 使用例

入力	結果
@SEL("1","0"  "1"*,"なし"  "あり"  "その他")	あり
@SEL("5","0"  "1"*,"なし"  "あり"  "その他")	その他
@SEL(10,0 <0 >0,"ゼロ"  "負の数"  "正の数")	正の数

## ■ 変更履歴

2014/05/12 比較演算子に対応。

**@IF <条件文字列選択>**

## ■ 概要

条件により文字列を選択する

## ■ 関数名

@IF(arg1,arg2,arg3)

arg1 条件式

arg2 条件が真のときに返す文字列

arg3 条件が偽のときに返す文字列

## ■ 使用可能な比較演算子

= (等しい), >(より大きい), <(より小さい), >=(以上), <=(以下), <>(等しくない)

== (文字列が等しい)

## ■ 使用例

入力	結果
@IF(3>=5,"正しい","誤り")	誤り
@IF("ABC"="ABC","正しい","誤り")	正しい
@IF("00"=="0","正しい","誤り")	誤り
@IF("00"="0","正しい","誤り")	正しい

## ■ 備考

= は可能な場合、数値に変換して比較をします。== は文字列として比較をします。

## ■ 変更履歴

2019/10/03 ==演算子を追加。

## @ERR <エラー発生>

### ■ 概要

エラーを発生させます。

エラーが発生すると印字が終了します。

### ■ 関数名

@ERR(arg1,arg2)

arg1 0 : エラーにしない。

1 : エラーにする。

arg2 エラーメッセージ。

return エラーが発生したかどうか。

0 : なし。

1 : エラー発生。

### ■ 使用例

入力	説明	結果
@ERR(1,"エラー発生")		エラー発生
@ERR(0," エラー発生")		
@ERR(@IF(3>=5,0,1),"エラー")		エラー

**@NOW <現在日時取得>**

## ■ 概要

現在日時を取得する。

## ■ 関数名

@NOW()

## ■ 使用例

入力	結果
@NOW()	2019-05-30 15:59:02

## ■ 備考

※日時の書式を変更する場合は、フィールドのフィルタを使用してください。

## @NOWADD <現在日時加算取得>

### ■ 概要

現在日時に指定した年月日を加算して取得する。

### ■ 関数名

@NOWADD(arg1,arg2,arg3)

arg1 加算する年数

arg2 加算する月数

arg3 加算する日数

### ■ 使用例

入力	結果
@NOWADD(0, 0, 0)	2019-05-30 15:59:02
@NOWADD(2, 0, 0)	2021-05-30 15:59:02
@NOWADD(0, 0, 60)	2019-07-29 15:59:02

### ■ 備考

※日時の書式を変更する場合は、フィールドのフィルタを使用してください。

## @WAR <割付名でデータ取得>

### ■ 概要

割付名からデータを取得する

### ■ 関数名

@WAR(arg1)

arg1 割付名

### ■ 使用例

入力	結果
@WAR(“品番”)	123456

### ■ 備考

フィールドマクロでは使用できません。

## @FIL <フィールド名でデータ取得>

### ■概要

フィールド名でデータを取得する

### ■関数名

@FIL(arg1)

arg1 フィールド名

### ■使用例

入力	結果
@FIL("フィールド名")	123456

### ■備考

2019/08/02 非推奨のコマンドです。

**@LST <リストからデータ取得>****■概要**

リストからデータを取得する

**■関数名**

@LST(arg1,arg2,arg3[,arg4] [,arg5])

arg1 検索条件となる文字列

arg2 リストのグループ名

arg3 取得したいデータ 1=キーから内容を取得、2=内容からキーを取得、3=他のグループで同じキーを持つ内容を取得

arg4 上記が3の場合のグループ名

arg5 リストに登録がない場合に返す文字列

**■使用例**

入力	結果
@LST("01","日本語原産国",1)	日本製
@LST("日本製","日本語原産国",2)	01
@LST("日本製","日本語原産国",3,"英語原産国")	Made In Japan
@LST("台湾製","日本語原産国",1,,"外国製")	外国製

※上記使用例での原産国内容

グループ	キー	内容
日本語原産国	01	日本製
日本語原産国	02	中国製
日本語原産国	03	韓国製
英語原産国	01	Made In Japan
英語原産国	02	Made In China
英語原産国	03	Made In Korea

**@TNG <単語からデータ取得>**

## ■ 概要

単語からデータを取得する

## ■ 関数名

@TNG(arg1,arg2)

arg1 検索条件となる文字列

arg2 取得したいデータ 1=キーから内容を取得、2=内容からキーを取得

## ■ 使用例

入力	結果
@TNG("01", 1)	日本製
@TNG("日本製", 2)	01

※上記使用例での原産国内容

キー	内容
01	日本製
02	中国製
03	韓国製

**@TAX <消費税計算>**

## ■ 概要

消費税を計算する

※消費税率と切り上げ方法はデータから取得する

## ■ 関数名

@TAX(arg1,arg2,arg3)

arg1 金額

arg2 税率 (%)

arg3 端数処理方法 1=切り上げ、2=切り捨て、3=四捨五入

## ■ 使用例

入力	結果
@TAX(10000,5%,1)	10500

## ■ 概要

環境変数を取得する

## ■ 関数名

@ENV(arg1)

arg1 変数名

使用例	結果
@ENV(消費税)	5
@ENV(端数処理)	1 *切り上げ

## @BCD <チェックデジット追加>

## ■ 概要

チェックデジットを計算し追加する

## ■ 関数名

@BCD(arg1,[arg2])

arg1 対象となる文字列

arg2 チェックデジット名 (省略すると JAN)

JAN

JAN13、JAN8 用

## ■ 使用例

入力	結果
@BCD(490123456789)	4901234567894

## @GET <フィールドプロパティ取得>

※フィールドマクロのみ対応

### ■概要

フィールドプロパティの値を取得します。

### ■関数名

@GET(arg1)

arg1 対象となるプロパティ

引数 : 取得内容

DATA : 内容

FNTW : フォント幅

FNTH : フォント高さ

LEN : 内容の文字数

LENB : 内容のバイト数 (Shift-JIS)

NOPRN : 印字しない (0=する, 1=しない)

### ■使用例

内容=12345, フォント幅 3mm

入力	結果
@GET(DATA)	12345
@GET(LEN)	5

### ■備考

マクロ割付では使用できません。

## @SET <フィールドプロパティ設定>

※フィールドマクロのみ対応

### ■概要

フィールドプロパティに値を設定します。

### ■関数名

@SET(arg1,arg2)

arg1 対象となるプロパティ

引数 : 取得内容

DATA : 内容

FNTW : フォント幅

FNTH : フォント高さ

NOPRN : 印字しない (0=する, 1=しない)

Arg2 設定する値

### ■使用例

内容=12345, フォント幅 3mm

入力	結果
@SET(DATA,"ABC")	内容=ABC
@SET(FNTW,5)	フォント幅=5mm

### ■備考

マクロ割付では使用できません。