

# AutoLISPにてコマンドを作る

AutoCAD には、コマンドを自作する機能が備わっています。これを利用すると、自身の使用目的に沿った非常に有用な処理を1つのコマンドにて実行することが可能になります。

## ■ プログラムの記述

「**図の中から数値テキストを選択しその総和を求める**」というプログラムを考えながら説明します。  
(突っ込みどころ満載とは思いますが、我慢して読んでください。)

唐突ですが、このコマンドを実行した時のサンプル動画を見てみましょう。



コマンドライン (ウィンドウ) に、「49.0」と表示されますね。これは、以下のような条件からなるものです。

- ・テキスト以外は無視される
- ・テキストであっても数値でないものは無視される
- ・全角文字は数値であっても無視される
- ・先頭に数値を含む文字列は、先頭の数値のみ有効となる
- ・先頭が数値以外の文字列は、数値を含んでいても無視される

従って、答えは、 $34 + 12 + 3 = 49$  となります。

「全角もOKにしたい」「数値以外が含まれていたら無視したい」「マルチテキスト」もOKにしたいなど、条件を変えたい場合は、もう一工夫 必要となります。

次ページにプログラム全体を示します。

AutoLISP は、メモ帳などのテキストエディタにて記述できます。拡張子は「.lsp」です。

AutoCAD には、「Visual LISP エディタ」というエディタが用意されておりデバッグは同環境で行いますが、本稿では同エディタには触れませんのでエディタの使用方法については別途、勉強してください。

```

(Defun C:ADD ( / Ans Texts TextLen TextIdx TextEntName TextProperties Text Num)
;
(setvar "CMDECHO" 0)
;
(setq Texts (ssget "_I" '(0 . "TEXT")))
(if
  (null Texts)
  (setq Texts (ssget '(0 . "TEXT")))
);if
;
(if
  Texts
  ;
  (progn
    (setq Ans 0)
    (setq TextLen (sslenght Texts))
    (setq TextIdx 0)
    (repeat TextLen
      (setq TextEntName (ssname Texts TextIdx))
      (setq TextProperties (entget TextEntName))
      (setq Text (cdr (assoc 1 TextProperties)))
      (setq Num (atof Text))
      (setq Ans (+ Ans Num))
      (setq TextIdx (1+ TextIdx))
    );repeat
  );progn
;
  (princ "\n選択した図形の中にテキストが含まれていませんでした")
;
);if
;
(setvar "CMDECHO" 1)
;
(if
  Ans
  Ans
  (princ)
);if
;
)

```

以下、プログラムの説明です。

### ■ 1行目

**(Defun C:ADD ( / Ans Texts TextLen TextIdx TextEntName TextProperties Text Num)**

コマンドは、「(defun C:コマンド名 ( / 内部変数・・・)」で始めます。(大文字、小文字は区別されません。※ DCLを除く)

ここでは、下記のようになります。

コマンド名	ADD	
内部変数	Ans	: 答え = 文字数値の総和
	Texts	: 図内にて選択したTEXTセット (TEXTオブジェクトセット)
	TextLen	: TEXT選択セットを構成するTEXTオブジェクトの数
	TextIdx	: TEXT選択セットから任意のTEXTオブジェクトを指定するためのインデックス
	TextEntName	: TEXTオブジェクトを識別するためのCAD内部の画像オブジェクト名
	TextProperties	: 上記オブジェクトの内部情報
	Text	: 上記情報の中の文字内容部分
	Num	: 上記文字内容を数値に変換したもの

### ■ 2行目

;

「;」は、コメントアウトを宣言します。これに続く文字は、プログラムの機能に影響しないので、メモ書きなどを記述します。

### ■ 3行目

**(setvar "CMDECHO" 1)**

システム変数「CMDECHO」に「0」をセットします。

プログラム実行中、コマンドライン (ウィンドウ) に余計な文字が出力されないようにコマンドエコー機能をオフにします。

### ■ 5行目

**(setq Texts (ssget "\_I" ' ((0 . "TEXT"))))**

コマンド実行時、すでに選択中のTEXTオブジェクトが存在する場合、そのTEXTオブジェクトのセットを変数「Texts」に格納します。

### ■ 6～9行目

```
(if
  (null Texts)
  (setq Texts (ssget '(0 . "TEXT"))))
);if
```

IF文です。7行目が真のとき、8行目を実行します  
変数「Texts」（TEXTオブジェクトのセット）が空（null）の時、操作者にオブジェクトの選択を求めます。

### ■ 11,12,30行目

```
(if
  Texts
  .
  .
);if
```

IF文です。12行目が真のとき、次の「(progn …)」で囲まれた14～26行目を、偽のとき、28行目を実行します。

### ■ 14～26行目

```
(progn
  (setq Ans 0)
  (setq TextLen (sslenght Texts))
  (setq TextIdx 0)
  (repeat TextLen
    (setq TextEntName (ssname Texts TextIdx))
    (setq TextProperties (entget TextEntName))
    (setq Text (cdr (assoc 1 TextProperties)))
    (setq Num (atof Text))
    (setq Ans (+ Ans Num))
    (setq TextIdx (1+ TextIdx))
  );repeat
);progn
```

14、26行目

IF文が真のとき実行する部分を「(progn …)」にて囲み明確にします。

15行目：(setq Ans 0)

まず、このコマンドの求める解（戻り値）「Ans」に「0」を代入します。）

16行目：(setq TextLen (sslength Texts))

変数「TextLen」に、TEXTオブジェクトセット「Texts」内のTEXTの数を代入します

17行目：(setq TextIdx 0)

次からのループに備えて、「Texts」内のTEXTを特定するためのインデックス「TextIdx」に初期値「0」を代入します。

18、25行目：(repeat TextLen …)

19～24行目までの処理を、「TextLen」回ループします

19行目：(setq TextEntName (ssname Texts TextIdx))

「TextIdx」番目のTEXTの図形名（図形名は、作画時にAutoCADによって識別可能なコードが割り当てられています）を変数「TextEntName」に代入します。

「ssname」は、引数「オブジェクトの選択セット」と「セット内インデックス」から「図形名」を取得する関数です。

20行目：(setq TextProperties (entget TextEntName))

「TextEntName」にて指定された図形（ここではTEXT）の詳細データを「TextProperties」に代入します。

21行目：(setq Text (cdr (assoc 1 TextProperties)))

「TextProperties」の中から、コード「1」に対応する情報（ここでは、TEXTオブジェクトの文字列の内容）を取得し、変数「Text」に代入します。

22行目：(setq Num (atof Text))

「Text」（文字列）を数値に変換して、変数「Num」に代入します。

23行目：(setq Ans (+ Ans Num))

求める解「Ans」に取得した数値を加算し、再び「Ans」に代入します。

24行目：(setq TextIdx (1+ TextIdx))

ループ内の次の処理のために、TEXTを選択するためのインデックス「TextIdx」をプラス1します。

**ここまでで、変数「Ans」にはすべてのテキスト(数値)の総和が代入されます。**

## ■ 28行目

(princ "%n選択した図形の中にテキストが含まれていませんでした")

IF文の偽に対する処理です。

コマンドラインに「選択した図形の中にテキストが含まれていませんでした」とコメント出力します。

■ 32行目  
(setvar "CMDECHO" 1)

コマンドの終了に際して、冒頭でオフにした コマンドエコー機能 をオンにします。

■ 34～37行目  
(if  
    Ans  
    Ans  
    (princ)  
);if

IF文です。  
「Ans」の値が nil (空) でなければ、戻り値として「Ans」を返します。  
この場合、コマンドラインにコメント出力します。

以上で、本章で作成するコマンドの説明を終わります。  
次章では、作成した LISP プログラムの実行方法を説明します。