**Alignment Application** 

SGALIGN

**Command Reference** 



# Contents

| 1. Command List                    | 4    |
|------------------------------------|------|
| 2. About Variables                 | . 14 |
| 3. About Commands                  | . 15 |
| 3-1. About Type                    | . 15 |
| 4. Commands in detail              | . 16 |
| 4-1. Algebraic operator (Numeric)  | . 16 |
| 4-2. String operation (TextSeq)    | . 22 |
| 4-3. Relational operator (Compare) | . 27 |
| 4-4. Control syntax (Control)      | . 32 |
| 4-5. Motor (Motor)                 | . 40 |
| 4-6. Multiple I/O (Multi I/O)      | . 48 |
| 4-7. Communication (Interface)     | . 49 |
| 4-8. Measurement (Measure)         | . 52 |
| 4-9. File (File)                   | . 54 |
| 4-10. Comment (Comment)            | . 60 |



# Introduction

This document lists and describes the commands for the automatically executable program (Sequence Program) that can be used in the SGALIGN series software for auto-alignment system control.

For details of the SGALIGN application software, please refer to the "Software User Manual".



# 1. Command List

<Algebraic operation>

| Туре               | Command      | Operation                     |
|--------------------|--------------|-------------------------------|
| Algebraic operator | =            | Assignment                    |
| Algebraic operator | +            | Addition                      |
| Algebraic operator | -            | Subtraction                   |
| Algebraic operator | *            | Multiplication                |
| Algebraic operator | 1            | Division                      |
| Algebraic operator | %            | Modulus                       |
| Algebraic operator | **           | Exponentiation                |
| Algebraic operator | 11           | Round down division           |
| Algebraic operator | abs          | Absolute                      |
| Algebraic operator | int          | Convert to integer type       |
| Algebraic operator | float        | Convert to floating type      |
| Algebraic operator | bin          | Convert to binary             |
| Algebraic operator | hex          | Convert to HEX                |
| Algebraic operator | max          | Maximum value                 |
| Algebraic operator | min          | Minimum value                 |
| Algebraic operator | perfcounter  | Time acquisition              |
| Algebraic operator | math.pi      | Circular constant             |
| Algebraic operator | math.e       | Natural logarithm             |
| Algebraic operator | math.sin     | Sine                          |
| Algebraic operator | math.cos     | Cosine                        |
| Algebraic operator | math.tan     | Tan                           |
| Algebraic operator | math.asin    | Inverse sine                  |
| Algebraic operator | math.acos    | Inverse cosine                |
| Algebraic operator | math.atan    | Inverse tangent               |
| Algebraic operator | math.degrees | Convert radian to degree      |
| Algebraic operator | math.radians | Convert degree to radian      |
| Algebraic operator | math.sqrt    | Square root                   |
| Algebraic operator | math.exp     | Exponentiation of e           |
| Algebraic operator | math.log     | Natural logarithm             |
| Algebraic operator | math.log10   | Common logarithm              |
| Algebraic operator | math.atan2   | Four-quadrant inverse tangent |
| Algebraic operator | math.pow     | Exponentiation                |



<String operation>

| Туре             | Command | Operation                        |
|------------------|---------|----------------------------------|
| String operation | =       | Assignment                       |
| String operation | +       | Concatenation                    |
| String operation | strip   | Remove two sides                 |
| String operation | Istrip  | Remove left side                 |
| String operation | rstrip  | Remove right side                |
| String operation | upper   | Convert to uppercase             |
| String operation | lower   | Convert to lowercase             |
| String operation | replace | Replacement                      |
| String operation | []      | Extraction                       |
| String operation | count   | Calculate the number of elements |
| String operation | find    | Find sub string from beginning   |
| String operation | rfind   | Find sub string from end         |
| String operation | len     | Length of a string               |
| String operation | str     | Convert to string                |
| String operation | join    | Concatenation                    |
| String operation | format  | Convert numeric value to string  |
| String operation | +       | Addition                         |
| String operation | -       | Subtraction                      |
| String operation | now     | Current date and time            |
| String operation | date    | Create DATE type                 |
| String operation | time    | Create TIME type                 |



<Relational operator>

| Туре                | Command | Operation  |
|---------------------|---------|--|
| Relational operator | =       | Assignment *Boolean type                               |
| Relational operator | = =     | Equality *Boolean type                                 |
| Relational operator | !=      | Inequality *Boolean type                               |
| Relational operator | or      | Logical OR *Integer, Floating, DATE, TIME type         |
| Relational operator | and     | Logical AND *Integer, Floating, DATE, TIME type        |
| Relational operator | = =     | Equality *Integer, Floating, DATE, TIME type           |
| Relational operator | !=      | Inequality *Integer, Floating, DATE, TIME type         |
| Relational operator | <       | Less than *Integer, Floating, DATE, TIME type          |
| Relational operator | >       | Greater than *Integer, Floating, DATE, TIME type       |
| Relational operator | <=      | Less than or equal *Integer, Floating, DATE, TIME type |
| Relational operator | >=      | Greater than or equal *Integer, Floating, DATE, TIME   |
|                     |         | type   |
| Relational operator | = =     | Equality *String type                                  |
| Relational operator | !=      | Inequality *String type                                |
| Relational operator | in      | In the set *String type                                |
| Relational operator | not in  | Not in the set *String type                            |



#### <Control syntax>

| Туре           | Command    | Operation                                      |
|----------------|------------|--|
| Control syntax | if         | Condition branch                               |
| Control syntax | elif       | Condition branch                               |
| Control syntax | else       | Condition branch                               |
| Control syntax | while      | Iterative operation 1                          |
| Control syntax | for        | Iterative operation 2                          |
| Control syntax | continue   | Return to the start of the iterative operation |
| Control syntax | def        | Define the function                            |
| Control syntax | return     | Return from the function                       |
| Control syntax | sleep      | Pause  |
| Control syntax | pause      | Standby  |
| Control syntax | input      | Screen input                                   |
| Control syntax | print      | Screen output                                  |
| Control syntax | messagebox | Message box                                    |



<Motor>

| Туре  | Command              | Operation  |
|-------|----------------------|--|
| Motor | axis[].name          | Acquisition of axis name                                   |
| Motor | axis[].getposition   | Acquisition of current position                            |
| Motor | axis[].getspeed      | Acquisition of setting speed                               |
| Motor | axis[].motionwait    | Wait for movement completion                               |
| Motor | axis[].speed         | Speed setting 1  |
| Motor | axis[].speed         | Speed setting 2  |
| Motor | axis[].acc           | Acceleration setting                                       |
| Motor | axis[].dec           | Deceleration setting                                       |
| Motor | axis[].limitplus     | Positive direction movement limit setting                  |
| Motor | axis[].limitminus    | Negative direction movement limit setting                  |
| Motor | axis[].incmove       | Relative movement  |
| Motor | axis[].absmove       | Absolute movement  |
| Motor | axis[].jogmove       | Jog movement   |
| Motor | multi.allstop        | Immediate stop for all axes                                |
| Motor | multi.stop           | Multi-axis deceleration stop                               |
| Motor | multi.setinc         | Relative Position Setting                                  |
| Motor | multi.incmove        | Relative movement for multiple axes                        |
| Motor | multi.increset       | Reset relative position for all axes                       |
| Motor | multi.setabs         | Absolute position setting                                  |
| Motor | multi.absmove        | Absolute movement for multiple axes                        |
| Motor | multi.absreset       | Reset of absolute movement for all axes                    |
| Motor | multi.allorigin      | Return to machine homing for all axes                      |
| Motor | multi.getallorigin   | Acquisition the status of machine homing for all axes      |
| Motor | multi.allmotionwait  | Wait until all axes have completed moving                  |
| Motor | multi.motionwait     | Wait until the specified axis have completed moving        |
| Motor | axis[].getemg        | Acquisition of emergency stop signal                       |
| Motor | axis[].getstop       | Acquisition of stop signal                                 |
| Motor | axis[].getstop0      | Acquisition of stop signal                                 |
| Motor | axis[].getstop1      | Acquisition of stop signal                                 |
| Motor | axis[].getstop2      | Acquisition of stop signal                                 |
| Motor | axis[].getdriveplus  | Acquisition of drive operation signal (positive direction) |
| Motor | axis[].getdriveminus | Acquisition of drive operation signal (negative direction) |
| Motor | axis[].getinposition | Acquisition of positioning completion                      |
| Motor | axis[ ].getalarm     | Acquisition of alarm status                                |
| Motor | jogshow              | Display the JOG screen                                     |



<Multiple I/O>

| Туре         | Command         | Operation                         |
|--------------|-----------------|-----------------------------------|
| Multiple I/O | dio.output      | General output signal setting     |
| Multiple I/O | analog.getvalue | Acquisition of input analog value |



| <communication></communication> |                    |                            |
|---------------------------------|--------------------|----------------------------|
| Туре                            | Command            | Operation                  |
| Communication                   | serial.connect     | Connect to the port        |
| Communication                   | serial.send        | Send text data             |
| Communication                   | serial.receive     | Receive text data          |
| Communication                   | serial.sendreceive | Send and receive text data |
| Communication                   | serial.disconnect  | Disconnect the port        |
| Communication                   | gpib.send          | Send text data             |
| Communication                   | gpib.receive       | Receive text data          |
| Communication                   | gpib.sendreceive   | Send and receive text data |
| Communication                   | gpib.open          | Connect to the board       |
| Communication                   | gpib.close         | Disconnect the board       |
| Communication                   | tcpip.connect      | Connect to the network     |
| Communication                   | tcpip.send         | Send text data             |
| Communication                   | tcpip.receive      | Receive text data          |
| Communication                   | tcpip.sendreceive  | Send and receive text data |
| Communication                   | tcpip.disconnect   | Disconnect the network     |



#### <Measurement>

| Туре        | Command              | Operation                    |
|-------------|----------------------|------------------------------|
| Measurement | alignment.raster     | Raster alignment             |
| Measurement | alignment.spiral     | Spiral alignment             |
| Measurement | alignment.line1      | 1-axis line alignment        |
| Measurement | alignment.line2      | 2-axis line alignment        |
| Measurement | alignment.line3      | 3-axis line alignment        |
| Measurement | alignment.neldermead | Neldermead alignment         |
| Measurement | alignment.min        | Acquisition of minimum value |
| Measurement | alignment.max        | Acquisition of maximum value |
| Measurement | alignment.moment     | Acquisition of barycenter    |



<File>

| <fiie></fiie> |                                  |                                 |
|---------------|----------------------------------|---------------------------------|
| Туре          | Command                          | Operation                       |
| File          | path.dirname                     | Obtain folder name              |
| File          | path.isdir                       | Confirm existence of folder     |
| File          | path.getcwd                      | Obtain working folder name      |
| File          | path.makedirs                    | Create folder                   |
| File          | path.rmtree                      | Delete folder                   |
| File          | path.basename                    | Obtain file name                |
| File          | path.isfile                      | Confirm existence of file       |
| File          | path.remove                      | Delete a file                   |
| File          | path.copy                        | Copy the file                   |
| File          | path.join                        | Join                            |
| File          | settingfile.readalignsettingfile | Load the alignment setting file |
| File          | csv.open                         | Open the file                   |
| File          | csv.setcell                      | Write to the specified cell     |
| File          | csv.setlinedata                  | Write the one-dimensional array |
| File          | csv.setarraydata                 | Write the two-dimensional array |
| File          | csv.getcell                      | Obtain the value in the cell    |
| File          | csv.close                        | Close the file                  |
| File          | text.open                        | Open the file                   |
| File          | text.write                       | Write                           |
| File          | text.writeline                   | Write with line feed            |
| File          | text.readline                    | Read                            |
| File          | text.close                       | Close the file                  |



| •Comment |         |           |
|----------|---------|-----------|
| Туре     | Command | Operation |
| Comment  | #       | Comment   |



# 2. About Variables

In the SGALIGN sequence program, use predefined variables to make calculation and character generation, etc. There are 6 types of variables, and the setting is performed from the variable list. Please refer software user manual for detail.

■Rules and precautions for setting variable names

Uppercase/lowercase sensitive

•Recommend to use uppercase for the first character. (A variable in lower-case may be misidentified as a command.)

- ·Double-byte characters can be used.
- •Using numerical value is not allowed.
- •Using ()[]., '" is not allowed.

• The first character cannot be + - \* / % : = ! < > ~ & | ^. There is no limitation for using at other places.

■About variable type

1)Boolean Type

A variable with Boolean data type that takes the value of either 0 (False) or 1 (True).

Up to 997 variables can be defined.

2)Integer Type A variable with integer data type. Up to 2999 variables can be defined.

3)Floating Type

A variable with floating-point data type. Up to 2999 variables can be defined.

#### 4)String Type

A variable with string data type. Control characters such as newline characters are not accepted. Up to 1999 variables can be defined.

#### 5)DATE Type

A variable with date data type. The format is "yyyy/MM/dd" in which "yyyy" is year, "MM" is month and "dd" is date. If no specification, the current date is used. Up to 499 variables can be defined.

#### 6)TIME Type

A variable with time date type. The format is "HH:mm:ss" in which "HH" is hour (24-hour time), the "mm" is minute and the "ss" is second. The input range is from 0:00:00 to 23:59:59. If no specification, the second is 0. Up to 499 variables can be defined.



# 3. About Commands

The commands consist of nine types and one comment command.

Most commands are performed with assignment operations, entering variables on the left side and expressions on the right side. The assignment (=) is always entered for assignment expressions.

The value can be directly entered one the right side.

When entering the value, using (") for string and (') for value.

### 3-1. About Type

| 1) Algebraic operation | Calculations and assignments of various values.                     |
|------------------------|---|
| 2) String operation    | Operation of string, date and time.                                 |
| 3) Relational operator | True/False judgments based on the results of variables comparison.  |
| 4) Control syntax      | Performing programmable control such as condition branch and loop.  |
| 5) Motor               | Control of the connected stages.                                    |
| 6) Multiple I/O        | Performing I/O control and analog data acquisition.                 |
| 7) Communication       | Performing Serial, Ethernet (TCP/IP) and GPIB communications.       |
| 8) Measurement         | Auto-alignment.   |
| 9) File                | Creating, deleting and existence confirming of the file and folder. |
| 10) Comment            | Comment function for memorizing the status and the process.         |



# 4. Commands in detail

4-1. Algebraic operator (Numeric)

#### Assignment (=)

Write right side value to Word1. Specify Integer type or Floating type for calculation. If Word1 is of type Integer, then Word3 can only specify of type Integer.

-Addition (+), Subtraction (-), Multiplication (\*), Division (/)

Specify Integer type or Floating type for calculation.

If Word1 is of type Integer, then Word3 and Word5 can only specify of type Integer.

Example: Numeric = 4 / 5 If Numeric is integer type, the result is 0. If Numeric is Floating type, 0.8 is assigned to.

Example: Numeric = 0.4 \* 5 If Numeric is a Floating type, the value of 2 is assigned. However, if it is an Integer type, an error occurs.

•Modulus (%)

Remainder after the division. If the value of Word3 is lower than Word5, it fails and Word1 is assigned the value of Word3. Specify Integer type or Floating type for calculation.

If Word1 is Integer type, the calculation side must be specified to Integer type.

Example: Numeric = 5 % 4 If Numeric is integer type, the result is 1.

Example: Numeric = 13 % 0.3 If Numeric is Floating type, the result is 0.1.

Exponentiation (\*\*)
 Specify Integer type or Floating type for calculation.
 If Word1 is Integer type, the calculation side must be specified to Integer type.

Example: Numeric = 2 \*\* 3 If Numeric is integer type, the result is 8.

Example: Numeric = 0.2 \*\* 3 If Numeric is Floating type, the result is 0.008.



Round down division (//) Specify Integer type or Floating type for calculation.
If Word1 is of type Integer, then Word3 and Word5 can only specify of type Integer.
Example: Numeric = 2 // 3
If Numeric is integer type, the result is 0.
Example: Numeric = 0.4 // 0.3
If Numeric is Floating type, the result is 1.

#### Absolute (abs)

Specify Integer type or Floating type for calculation. If Word1 is of type Integer, then Word4 can only specify of type Integer.

```
Example: Numeric = abs(-5)
Assign "5" to Numeric.
```

Convert to integer type (int)

Convert String type value to Integer type. Word1 is Integer type. Discard all digits to the right of the decimal point in String type value. Return "0" if the value cannot be converted.

```
Example: Integer01 = int( Data01 ) *Data01 = "214.4msec"
Assign "0" to Integer01.
```

Example: Integer01 = int( Data01 ) \*Data01 = "214.56" Assign "214" to Integer01.

Convert to floating type (float)

Convert String type value to Floating type. Word1 is Floating type. Return "0" if the value cannot be converted.

```
Example: Floating01 = float( Data01 ) *Data01 = "214.4msec"
```

Assign "0" to Floating01.

•Tips: To convert the String "214.4msec" to Floating type?

```
Data01 = Data01.replace( 'msec' '')
```

Floating01 = float( Data01 )

For example, use the "replace" command to delete "msec" and replace it with "214.4". Then, execute the "float" command.

```
Example: Floating01 = float( Data01 ) *Data01 = "214.56"
Assign "214.56" to Floating01.
```



Convert to binary (bin)
Convert Integer type value to binary digit string. Word1 is String type.
Example: Data01 = bin(240)
"0b11110000" is assigned to Data01 starting with "0b".

Convert to HEX (hex)
 Convert Integer type value to hexadecimal digit string. Word1 is Floating type.

Example: Data01 = hex(240) "0xf0" is assigned to Data01 starting with "0x".

Maximum value (max)

Return the maximum value in the two or three values.

Specify Integer type or Floating type for calculation.

If Word1 is Integer type, the calculation side must be specified to Integer type.

Example: Numeric = max(14.0 51) "51" is assigned to Numeric.

Example: Numeric = max(14.0 51 -6.45 "51" is assigned to Numeric.

•Minimum value (min)

Return the minimum value in the two or three values.

Specify Integer type or Floating type for calculation.

If Word1 is Integer type, the calculation side must be specified to Integer type.

Example: Numeric = max(14.0 51) "14.0" is assigned to Numeric.

Example: Numeric = min(14.0 51 -6.45 "-6.45" is assigned to Numeric.

Time acquisition (perfcounter)

Obtains the elapsed time since the sequence execution started. The unit is msec. Word1 is Integer type.

Example: Integer01 = perfcounter()



Circular constant (math.pi)
 Fixed value of Floating type. Return the circular constant (3.14...).
 Word1 is Floating type.

Example: Floating01 = math .pi

Natural logarithm (math.e)
Fixed value of Floating type. Return the natural logarithm base (2.71...).
Word1 is Floating type.

Example: Floating01 = math .e

Sine (math.sin)
 Return the sine of the input value whose unit is radian.
 Word1 is Floating type. Word5 is Integer type or Floating type.

Example: Floating01 = math .sin( 0.7853 ) "0.7070..." is assigned to Floating01

Cosine (math.cos)

Return the cosine of the input value whose unit is radian. Word1 is Floating type. Word5 is Integer type or Floating type.

Example:Floating01 = math .cos( 0.7853 ) "0.7071..." is assigned to Floating01.

Tangent (math.tan)

Return the tangent of the input value whose unit is radian. Word1 is Floating type. Word5 is Integer type or Floating type.

Example : Floating01 = math .tan( 0.7853 ) "0.9998..." is assigned to Floating01.

Inverse sine (math.asin)

Return the inverse sine of the input value whose unit is radian. Word1 is Floating type. Word5 is Integer type or Floating type.

Example: Floating01 = math .asin(1) "1.5707..." is assigned to Floating01.



Inverse cosine (math.acos)
 Return the inverse cosine of the input value whose unit is radian.
 Word1 is Floating type. Word5 is Integer type or Floating type.

Example:Floating01 = math .acos(1) "0" is assigned to Floating01.

Inverse tangent (math.atan)

Return the inverse tangent of the input value whose unit is radian. Word1 is Floating type. Word5 is Integer type or Floating type.

Example: Floating01 = math .atan(1) "0.7853..." is assigned to Floating01.

Convert radian to degree (math.degrees)
Covert radian to degree.
Word1 is Floating type. Word5 is Integer type or Floating type.

Example:Floating01 = math .degrees( 0.7853 ) "44.9943..." is assigned to Floating01.

Convert degree to radian (math.radians)
 Convert degree to radian.
 Word1 is Floating type. Word5 is Integer type or Floating type.

Example: Floating01 = math .radians(45) "0.7853..." is assigned to Floating01.

•Square root (math.sqrt)

Return square root of the input value.

Word1 is Floating type. Word5 is Integer type or Floating type.

Example:Floating01 = math .sqrt( 256 ) "16" is assigned to Floating01.

Exponentiation of e (math.exp)

Return the result of the e (the base of the natural logarithm) raised to the power of the input value. Word1 is Floating type. Word5 is Integer type or Floating type.

Example: Floating01 = math .exp(2) "7.3890..." is assigned to Floating01.



Natural logarithm (math.log)
Return the natural logarithm (base e) of the input value.
Word1 is Floating type. Word5 is Integer type or Floating type.

Example: Floating01 = math .log( 7.4) "2.0014..." is assigned to Floating01.

Common logarithm (math.log10)
Return the common logarithm of the input value.
Word1 is Floating type. Word5 is Integer type or Floating type.

Example:Floating01 = math .log10( 100 ) "2" is assigned to Floating01.

Four-quadrant inverse tangent (math.atan2)
 Return the four-quadrant inverse tangent of 2 arguments. Equal to atan(NumericA / NumericB).
 Word1 is Floating type. Word5 and Word6 is Integer type or Floating type.

Example: Floating01 = math .atan2( NumericA NumericB \*NumericA = 1.732 , NumericB = 1 "0.5236..." is assigned to Floating01.

Exponentiation (math.pow)

Determine the exponentiation. Equal to NumericA \*\* NumericB.

Word1 is Floating type. Word5 and Word6 is Integer type or Floating type.

Example: Floating01 = math .pow( NumericA NumericB ) \*NumericA = 10 , NumericB = 2 "100" is assigned to Floating01.



## 4-2. String operation (TextSeq)

Assignment (=)

Write right side value to Word1. Specify String type, DATE type or TIME type for calculation. When entering text directly into Word3, enclose it in "" (Pairs of double quotation marks) or " (Pairs of single quotation marks). However, it is not required for DATE and TIME types.

| Example: Assign "equal" t | o the String type variable in String01.                             |
|---------------------------|---|
| String01 = 'equal'        | "equal" is assigned to String01.                                    |
| String01 = "equal"        | "equal" is assigned to String01.                                    |
| String01 = equal          | error occurs. (Not enclosed in the quotation marks.)                |
| String01 = 24             | error occurs. (Numbers cannot be specified directly.)               |
|                           |   |
| Example: Assign March 1   | 1, 2010 to the DATE type variable in Date01.                        |
| Date01 = 3/11/10          | "3/11/10" is assigned to Date01.                                    |
| Date01 = "3/11/10"        | error occurs. (It does not need to be enclosed in quotation marks.) |
| Date01 = 3/10             | The current year and March 11 are assigned to Date01.               |
|                           |   |
| Example: Assign 15:11:10  | ) to the TIME type variable in Time01.                              |
| Time01 = 15:11:10         | "15:11:10" is assigned to Time01.                                   |
| Time01 = "15:11:10"       | error occurs. (It does not need to be enclosed in quotation marks.) |
| Time01 = 15:11            | "15:11:00" is assigned to Time01.                                   |

Concatenation (+)

Concatenate Word3 and Word5. Specify String type for calculation.

Example: String01 = '24 ' + 'Hours' \*Error happens when using numerical value of 24. "24 Hours" is assigned to String01.

•Remove two sides (strip) Remove spaces at the beginning and at the end of the string. Specify String type for calculation.

Example: String01 = ' 24 Hour '.strip() "24 Hours" is assigned to String01.

```
•Remove left side (Istrip)
```

Remove spaces at the beginning of the string. Specify String type for calculation.

Example: String01 = ' 24 Hour '.lstrip() "24 Hours " is assigned to String01.



```
•Remove right side (rstrip)
Remove spaces at the end of the string. Specify String type for calculation.
```

```
Example: String01 = ' 24 Hour '.rstrip()
```

" 24 Hours" is assigned to String01.

Convert to uppercase (upper)
 Convert a string to uppercase letter. Specify String type for calculation.

Example: String01 = '24 Hours' .upper() "24 HOURS" is assigned to String01

•Convert to lowercase (lower) Convert a string to lowercase letter. Specify String type for calculation.

Example: String01 = '24 Hours' .lower() "24 hours" is assigned to String01

Replacement (replace)

Replace all of the string Word5 in the string Word3 to the string Word6. Specify String type for calculation.

Example: String01 = '24 Hours' .replace( 'ours' 'ur' ) "24 hour" is assigned to String01.

#### Extraction ([])

The Word4 is the start position in Word3 and the Word6 is the end position in Word3.

Extract the string from the start position to one before the end position.

The first character starts from 0 in original string.

Specify String type for Word3 and Integer type for Word4 & Word5 for calculation.

Example: String01 = 'Let's try!!' 6 :: 9 "try" is assigned to String01.

Example: It is also possible to refer to a variable of type String.

Data01 = 'Let's try!!' String01 = Data01 6 :: 9

"try" is assigned to String01.



•Number of elements (count)

Count the number of the string Word5 in the string Word3. Return 0 if not found Specify String type for calculation. The calculation is uppercase, lowercase, single-byte character and double-byte character sensitive. Word1 is Integer type.

Example: Integer01 = 'Let's try!!' .count( 't' ) "2" is assigned to String01.

Example: Integer01 = 'Let's Try!!' .count( 't' ) "1" is assigned to String01.

Find sub string from beginning (find)
 Return the position where the string Word5 is first found from the beginning of the string Word3.
 The first character starts from 0 in the string Word3. Word1 is Integer type.
 Return -1 if not found.

Example: Integer01 = 'Let's try!!' .find( '!' ) "9" is assigned to String01.

Find sub string from end (rfind)
Return the position where the string Word5 is first found from the end of the string Word3.
The first character starts from 0 in the string Word3. Word1 is Integer type.
Return -1 if not found.

Example 1: Integer01 = 'Let's try!!' .rfind( '!' ) "10" is assigned to String01.

Example 2: Integer01 = 'Let's try!!' .rfind( '!!' ) "9" is assigned to String01.

·Length of a string (len)

Return the length of the string Word4. A double-width character is treated as a single character. Specify String type, DATE type or TIME type for calculation. Word1 is Integer type.

```
Example: Integer01 = len( 'Let's try!!' )
"11" is assigned to String01.
```



Convert to string (str)
Convert the values in Word4 to string.
DATE type is converted to Windows "Long date" format.
TIME type is converted to Windows "Short time" format.

Example: If Word4 is numeric (Integer or Floating type) String01 = str( 3.14159 ) "3.14159" is assigned to String01.

Example: If Word4 is of type DATE Date01 = 3/10/2011 String01 = str( Date01 ) Assign "Thursday,3/10,2011" to String01.

Example: If Word4 is of type TIME Time01 = 15:10:30 String01 = str( Time01 ) Assign "03:10 PM" to String01.

| Regio   | on               |                               | X |
|---------|------------------|-------------------------------|---|
| Formats | Administrative   |                               |   |
| Forma   | t: English (Unit | ed States)                    |   |
| Match   | h Windows disj   | olay language (recommended) V |   |
| Langu   | age preference   | 5                             |   |
| Date    | and time form    | ats                           |   |
| Shor    | t date:          | M/d/yyyy $\checkmark$         |   |
| Long    | date:            | dddd, MMMM d, yyyy            |   |
| Shor    | t time:          | h:mm tt 🗸 🗸                   |   |
| Long    | j time:          | h:mm:ss tt 🗸 🗸                |   |
| First   | day of week:     | Sunday 🗸                      |   |
| France  |                  |                               |   |
| Shor    | t date:          | 8/22/2024                     |   |
| Long    | date:            | Thursday, August 22, 2024     |   |
| Shor    | t time:          | 3:22 PM                       |   |
| Long    | ) time:          | 3:22:29 PM                    |   |
|         |                  | Additional settings           |   |
|         |                  |                               |   |

Concatenation (join)

Join Word4, Word5 and Word6. All variable type can be specified for calculation. If Word6 is not used, input ")".

Example: String01 = join( "Time: " 124.6 "msec" " Time: 124.6msec" is assigned to String01.

Example: String01 = join( 124.6 "msec" ) "124.6msec" is assigned to String01.

·Convert numeric value to string (format)

Convert the value of Word4 to string in the format specified in the Word5 and return. Specify Integer type or Floating type for Word4, and String type for Word5 for calculation.

The value is rounded off to the specified place.

Example: String01 = format( -1234567.895678 "0,0.000" ) "-1,234,567.896" is assigned to String01.

Example: String01 = format( 4567.8 "000,000.000" ) "004,567.800" is assigned to String01.



```
 Addition (+)

DATE type is calculated in days and TIME type is calculated in minutes.
Example: Date01 = 4/8 + 9
"2022/04/17" is assigned to Date01. *In the case that the year is 2022
Example: Time01 = 17:23 + 95
"18:58:00" is assigned to Time01.

    Subtraction (-)

DATE type is calculated in days and TIME type is calculated in minutes.
Example: Date01 = 4/8 - 9
"2022/03/30" is assigned to Date01. *In the case that the year is 2022
Example: Time01 = 17:23 - 95
"15:48:00" is assigned to Time01.
•Current date and time (now)
Return the current date and time. Specify DATE type or TIME type for calculation.
Example: When the present date is March 30, 2022.
        Date01 = now()
                                  * Date01 is a variable of type DATE.
Date01 is assigned 03/30/2022.
Example: When the present time is 15:48:05
                                  * Time01 is a variable of type TIME.
        Time01 = now()
Date01 is assigned 15:48:05.

    Create DATE type (date)

Create DATE type by entering the year in Word4, the month in Word5, and the day in Word6.
Example: Date01 = date( 2022 3 30 )
"2022/03/30" is assigned to Date01.
•Create TIME type (time)
Create TIME type by entering the hour in Word4, the minute in Word5, and the second in Word6.
Example: Time01 = time(15 \ 48 \ 5)
"15:48:05" is assigned to Time01.
```



## 4-3. Relational operator (Compare)

#### Assignment (=)

Write right side value to Word1. Word1 and Word3 is Boolean type.

Example: Boolean01 = False Assign False to Boolean01.

#### •Equality (==)

Return True to the Word1 if the value of Word3 and the value of Word5 are same.

Return False if they are different.

#### Boolean

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| False | ==    | False | True   |
| False | ==    | True  | False  |
| True  | ==    | False | False  |
| True  | ==    | True  | True   |

#### Integer and Floating

TIN 40

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| 5     | ==    | 5.0   | True   |
| 5     | ==    | -5    | False  |
| 5     | ==    | 5.8   | False  |

#### DATE

| Word3   | Word4 | Word5   | Result |
|---------|-------|---------|--------|
| 4/17    | ==    | 04/17   | True   |
| 4/17    | ==    | 3/30    | False  |
| 22/4/17 | ==    | 4/04/17 | False  |

| Word3   | Word4 | Word5   | Result |
|---------|-------|---------|--------|
| 9:20:44 | ==    | 9:20:44 | True   |
| 9:20:44 | ==    | 9:20    | False  |
| 9:20    | ==    | 21:20   | False  |

#### Example: Boolean01 = False == False

"True" is assigned to Boolean01.

#### Inequality (!=)

Return True to the Word1 if the value of Word3 and the value of Word5 are different.

Return False if they are same.

#### Boolean

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| False | !=    | False | False  |
| False | !=    | True  | True   |
| True  | !=    | False | True   |
| True  | !=    | True  | False  |

#### DATE

| Word3   | Word4 | Word5   | Result |
|---------|-------|---------|--------|
| 4/17    | !=    | 04/17   | False  |
| 4/17    | !=    | 3/30    | True   |
| 22/4/17 | !=    | 4/04/17 | True   |

Example: Boolean01 = False != False "False" is assigned to Boolean01.

#### Integer and Floating

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| 5     | !=    | 5.0   | False  |
| 5     | !=    | -5    | True   |
| 5     | !=    | 5.8   | True   |

#### TIME

| Word3   | Word4 | Word5   | Result |
|---------|-------|---------|--------|
| 9:20:44 | !=    | 9:20:44 | False  |
| 9:20:44 | !=    | 9:20    | True   |
| 9:20    | !=    | 21:20   | True   |



#### ■Boolean type

#### Logical OR (or)

Return True if either the value of Word3 or the value of Word5 is True. If they are both False, return False.

| Example: Boolean01 = False or False |
|-------------------------------------|
| "False" is assigned to Boolean01.   |
|                                     |

Example: Boolean01 = False or True "True" is assigned to Boolean01.

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| False | or    | False | False  |
| False | or    | True  | True   |
| True  | or    | False | True   |
| True  | or    | True  | True   |

#### •Logical AND (and)

Return True if the value of Word3 and the value of Word5 are both True. Otherwise, returns False.

Example: Boolean01 = False and False "False" is assigned to Boolean01.

Example: Boolean01 = False and True "False" is assigned to Boolean01.

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| False | and   | False | False  |
| False | and   | True  | False  |
| True  | and   | False | False  |
| True  | and   | True  | True   |



#### ■Integer type, Floating type, DATE type, TIME type

#### Less than (<)</li>

Return True to Word1 if the value of Word5 is greater than the value of Word3.

Return False if equal or less.

#### Integer and Floating

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| 5     | <     | 5.0   | False  |
| 5     | <     | -5    | False  |
| 5     | <     | 5.8   | True   |

#### DATE

| Word3   | Word4 | Word5    | Result |
|---------|-------|----------|--------|
| 4/17    | <     | 04/17    | False  |
| 4/17    | <     | 3/30     | False  |
| 22/4/17 | <     | 24/04/17 | True   |

| TIME    |       |         |        |  |
|---------|-------|---------|--------|--|
| Word3   | Word4 | Word5   | Result |  |
| 9:20:44 | <     | 9:20:44 | False  |  |
| 9:20:44 | <     | 9:20    | False  |  |
| 9:20    | <     | 21:20   | True   |  |

Example: Boolean01 = 5 < 5.8

"True" is assigned to Boolean01.

#### •Greater than (>)

Return True to Word1 if the value of Word5 is less than the value of Word3.

Return False if equal or greater.

#### Integer and Floating

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| 5     | >     | 5.0   | False  |
| 5     | >     | -5    | True   |
| 5     | >     | 5.8   | False  |

#### DATE

| Word3   | Word4 | Word5    | Result |
|---------|-------|----------|--------|
| 4/17    | >     | 04/17    | False  |
| 4/17    | >     | 3/30     | True   |
| 22/4/17 | >     | 24/04/17 | False  |

#### TIME

| Word3   | Word4 | Word5   | Result |
|---------|-------|---------|--------|
| 9:20:44 | >     | 9:20:44 | False  |
| 9:20:44 | >     | 9:20    | True   |
| 9:20    | >     | 21:20   | False  |

# Example: Boolean01 = 5 > 5.8

"False" is assigned to Boolean01.



#### Less than or equal (<=)</li>

Return True to Word1 if the value of Word5 is equal to or greater than the value of Word3.

#### Return False if less.

#### Integer and Floating

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| 5     | <=    | 5.0   | True   |
| 5     | <=    | -5    | False  |
| 5     | <=    | 5.8   | True   |

#### DATE

| Word3   | Word4 | Word5    | Result |
|---------|-------|----------|--------|
| 4/17    | <=    | 04/17    | True   |
| 4/17    | <=    | 3/30     | False  |
| 22/4/17 | <=    | 24/04/17 | True   |

| TIME    |       |         |        |  |
|---------|-------|---------|--------|--|
| Word3   | Word4 | Word5   | Result |  |
| 9:20:44 | <=    | 9:20:44 | True   |  |
| 9:20:44 | <=    | 9:20    | False  |  |
| 9:20    | <=    | 21:20   | True   |  |

#### Example: Boolean01 = 5 <= 5.8

"True" is assigned to Boolean01.

•Greater than or equal (>=)

Return True to Word1 if the value of Word5 is equal to or less than the value of Word3.

Return False if greater.

#### Integer and Floating

| Word3 | Word4 | Word5 | Result |
|-------|-------|-------|--------|
| 5     | >=    | 5.0   | True   |
| 5     | >=    | -5    | True   |
| 5     | >=    | 5.8   | False  |

#### DATE

| Word3   | Word4 | Word5    | Result |
|---------|-------|----------|--------|
| 4/17    | >=    | 04/17    | True   |
| 4/17    | >=    | 3/30     | True   |
| 22/4/17 | >=    | 24/04/17 | False  |

# Word3 Word4 Word5 Result 9:20:44 >= 9:20:44 True 9:20:44 >= 9:20 True 9:20 >= 21:20 False

Example: Boolean01 = 5 >= 5.8

"False" is assigned to Boolean01.



#### ■String type

```
•Equality (==)
```

Return True to Word1 if the value of Word3 and the value of Word5 are same. Returns False if they are different.

Example: Boolean01 = "Hello" == "Hello" "True" is assigned to Boolean01.

Example: Boolean01 = "Hello" == "hello" "False" is assigned to Boolean01.

Inequality (!=)

Return True to Word1 if the value of Word3 and the value of Word5 are different. Returns False if they are same.

Example: Boolean01 = "Hello" != "hello" "True" is assigned to Boolean01.

```
Example: Boolean01 = "Hello" != "Hello"
"False" is assigned to Boolean01.
```

In the set (in)

Return True to Word1 if the value (element) of Word3 is included in the value (set) of Word5. Return False if no.

Example: Boolean01 = "el" in "Hello" "True" is assigned to Boolean01.

Example: Boolean01 = " ei" in "Hello" "False" is assigned to Boolean01.

Not in the set (not in)

Return True to Word1 if the value (element) of Word3 is not included in the value (set) of Word5. Return False if includes.

Example: Boolean01 = "he" not in "Hello" "True" is assigned to Boolean01.

Example: Boolean01 = "He" not in "Hello" "False" is assigned to Boolean01.



## 4-4. Control syntax (Control)

```
    Condition branch (if, elif, else)

If the conditional expression inside the parentheses is True, the sequence command in the next line is executed.
If False, it moves to the next conditional branch of "elif" or "else". There is no conditional expression for else.
"elif" can be used more than once. "else" should be used in the last conditional branch.
"end" is required at the end of the conditional branch.
* Condition branch "if" statement cannot be inserted into the "if-elif-else-end" statement.
* Iterations such as while and for can be inserted into the condition branch.
Example: "if" only
if (Floating01 == 1.23)
print( "Execute this line when Floating01 is 1.23.")
end
Example: "if" and "else"
if (Boolean01)
print( "Execute this line when Boolean 01 is True.")
else
print( "Execute this line when Boolean 01 is False.")
end
Example: "if" and "elif" and "else"
                                             * The "else" may be omitted.
if (Boolean01)
print( "Execute this line when Boolean 01 is True.")
elif (Boolean02)
print( "If Boolean01 is False and Boolean02 is True, execute this line." )
elif (Boolean03)
print( "If Boolean01 and Boolean02 are False and Boolean03 is True, execute this line." )
else
print( "If Boolean01, Boolean02 and Boolean03 are false, execute here." )
end
```



• Iterative operation 1 (while)

If the conditional expression inside the parentheses is True, the sequence command lines to the "end" are repeatedly executed.

\* Iterative operation such as "while" and "for" cannot be inserted into the Iterative operation.

\* Condition branch "if" can be inserted into the Iterative operation.

```
Example:
Integer01 = 0
while (Integer01 < 5)
Integer01 = Integer01 + 1
end
```

Assign values to Integer01 in the order of 1, 2, 3, 4, 5 and exit.

```
·Iterative operation 2 (for)
```

Perform iterative operation with a specified loop count. Specify Integer type for Word2.

The value in Word2 is incremented by 1 for each loop, and when the value reaches the value set in Word5 or Word6, the iteration ends.

\* Iterative operation such as "while" and "for" cannot be inserted into the Iterative operation.

\* Condition branch "if" can be inserted into the Iterative operation.

There are two patterns of syntax.

Pattern 1:

The value in Word5 is assigned to Word2 as the initial value.

The line in "for ~ end" is repeated from the value in Word2 to before the value in Word6.

Example: for Integer01 in range (2 7) Integer02 = Integer01 - 1 end

Assign values to Integer02 in the order of 1, 2, 3, 4, 5 and exit when Integer01 is 6.

Pattern 2: The line in for ~ end is repeated from the value in Word2 to before the value in Word5.

```
Example:
for Integer01 in range ( 5 )
Integer02 = Integer01 + 1
end
```

Assign values to Integer02 in the order of 1, 2, 3, 4, 5 and exit when Integer01 is 4.



•Exit from Iterative operation (break)

Exit from Iterative operation "while" and "for".

Used to exit from a loop or to interrupt processing in the middle of a loop.

```
Example: When using the "while" command
Integer01 = 0
while (True)
if (Integer01 == 5)
break
end
Integer01 = Integer01 + 1
print(Integer01)
end
Example: When using the "for" command
Integer01 = 0
for Integer01 in range(8)
if (Integer01 == 5)
break
end
Integer02 = Integer01 + 1
print(Integer02)
end
```

After the values 1, 2, 3, 4, and 5 are assigned and displayed in sequence, the loop is exited.



•Return to the start of the iterative operation (continue)

Return to the start of the iterative operation "while" and "for".

Used when a process in a loop does not want to proceed under certain conditions.

```
Example:
Integer01 = 0
for Integer01 in range(7)
if (Integer01 == 5)
continue
end
Integer02 = Integer01 + 1
print(Integer02)
End
```

Assign values 1, 2, 3, 4, 5, and 7 to Integer02 in that order and display. 6 is not assigned to Integer02 and is not displayed.



•Define the function (def)

•Return from the function (return)

Call the function whose name is defined in Word2 from the main sequence and execute the lines following the "def" to "return". Return to the call position once execution is completed.

\*The number of function definitions that can be registered is 1000.

\*The function definition must be defined below the end line of the main sequence.

\*Variable cannot be assigned to Word2

\*Arguments cannot be contained within the parentheses of a function definition.

\*If a comment is given in the line before the function definition, it is displayed as a tooltip when the function definition is registered.

\* The functions you define can be called any number of times at any location.

Example: Create a function name "Test".

```
# Check for positive integers.
                                            *Add Tips to the target function with a comment line.
def Test ()
String01 = input( "Enter a positive integer.")
Integer01 = int(String01)
Floating01 = float(String01)
if (Floating01 < 0)
Boolean01 = False
print( "A negative number was entered." )
elif (Floating01 == 0)
Boolean01 = False
print( " Characters have been entered." )
elif (Floating01 == Integer01)
Boolean01 = True
print( " Input value = " Integer01 )
end
return
```

In "Test function", "Boolean01" is set to "True" when a positive integer is entered. Otherwise, "False" is set. If you want to execute a function you have created, the format is as follows.

Test()

When the mouse cursor is hovered over this Test(), "Check for positive integers." is displayed as a tooltip.



Pause (sleep)

Sequence execution is stopped for a time (unit: msec) set by the value of Word2. Specify Word2 as Integer type for calculation.

Example: print( 'Hello' ) sleep( 300 ) The text "Hello" is displayed for 300 msec.

Standby (pause)

The program is paused and waits until the resume button is pressed.

Example: print( 'Hello' ) pause()

The text "Hello" appears and stops. Pressing the Resume button resumes program execution.

Screen input (input)

An input screen window is displayed with the value of Word4 as the message. The input value is assigned to Word1. Word1 has specified variable type for calculation. If the input value cannot be converted to the specified variable type, an error occurs and the sequence execution is terminated. It is recommended to calculate by specifying Word1 as String type.

Example: Integer01 = input( 'Please enter a number' ) If enter "20", "20" is assigned to Integer01. If enter "E20", an error occurs.

Example: String01 = input( 'Please enter a number' ) Integer01 = int( String01 )

If enter "20", "20" is assigned to String01 and "20" is assigned to Integer01. If enter "E20", "E20" is assigned to String01 and "0" is assigned to Integer01.

Screen output (print)

Displays values from Word2 to Word6 as text. All variable types can be specified.

(When in edit mode, the output screen window is displayed.)

Using parentheses ")" to disable the display up to Word6. When outputting, a space is inserted between the words.

Example: String01 = "The value is" Integer01 = 20 print( String01 Integer01 '.' )

Output "The value is 20.



Message box (messagebox)

The message can be defined by a combination of 6 types of buttons and icons.

Once a button is clicked, corresponding data is returned to Word1.

Word5 is message and Word6 is title.

The icon is displayed according to the specified mode.

Word1 is specified as Boolean type, Integer type, Floating type, and String type for calculation

\* Return data of the messagebox

| Button | String | Value | Boolean |
|--------|--------|-------|---------|
| OK     | OK     | 1     | True    |
| Cancel | Cancel | 2     | False   |
| Yes    | Yes    | 6     | True    |
| No     | No     | 7     | False   |

1) No icon (messagebox.show)

The only button is OK. There is no icon for edit mode.

Example: Boolean01 = messagebox .show( "Message" "No Icon" ) Clicking the OK button assigns "True" to Boolean01.

2) Information icon (messagebox.showinfo)

The only button is OK. The edit mode icon is letter "i" with blue background. Example: String01 = messagebox .showinfo( "Message" "Information Icon" ) Clicking the OK button assigns "OK" to String01.

#### 3) Warning icon (messagebox.showwarning)

The only button is OK. The edit mode icon is letter "!" with yellow background. Example: Integer01 = messagebox .showwarning( "Message" "Warning Icon" ) Clicking the OK button assigns "1" to Integer01.

4) Error icon (messagebox.showerror)

The only button is OK. The edit mode icon is letter "X" with red background. Example: Boolean01 = messagebox .showerror( "Message" "Error Icon" ) Clicking the OK button assigns "True" to Boolean01.



No Icon

X



5) Question icon 1 (messagebox.askyesnocancel)

"OK" and "Cancel" buttons are displayed. The edit mode icon is letter "?" with blue background.

Example: String01 = messagebox .askokcancel( "Message" "Question Icon 1" ) Clicking the OK button assigns "OK" to String01. Clicking the Cancel button assigns "Cancel" to String01.

6) Question icon 2 (messagebox.askyesno)

"Yes" and "No" buttons are displayed. The edit mode icon is letter "?" with blue background.

Example: Integer01 = messagebox .askyesno( "Message" "Question Icon 2" )

Clicking the Yes button assigns 6 to Integer01.

Clicking the No button assigns 7 to Integer01.



OK

X

Cancel



#### 4-5. Motor (Motor)

There are two control methods: simultaneous drive and independent drive.

For simultaneous drive, enter "multi" in Word 1, and for independent drive, enter "axis[]" in Word 1.

Acquisition of axis name (axis[].name)

The axis name specified in Word1 is assigned to Word3. Word3 is specified to String type for calculation.

Example: axis[X] .name( String01 ) "X" is assigned to String01.

·Speed setting 1 (axis[].speed)

Set the driving speed for the axis specified in Word1

Set the starting speed (initial speed) [mm/sec] in Word3 and the maximum speed [mm/sec] in Word4. Specify the Floating type for calculation.

Example: axis[X] .speed(0.5 5.0)

The X axis speed is set with a starting speed of 0.5 mm/sec and a maximum speed of 5.0 mm/sec.

Speed setting 2 (axis[].speed)

Set the driving speed for the axis specified in Word1

Set the starting speed (initial speed) [mm/sec] in Word3, the maximum speed [mm/sec] in Word4, and the acceleration [mm/sec] in Word5. Specify the Floating type for calculation.

Example: axis[X] .speed(0.5 5.0 20)

The X axis speed is set with a starting speed of 0.5 mm/sec, a maximum speed of 5.0 mm/sec, and a acceleration of 20 mm/sec.

Acceleration setting (axis[].acc)

Set the acceleration for the axis specified in Word1. Set the acceleration [mm/sec] in Word3. Specify the Floating type for calculation.

Example: axis[X] .acc( 20 ) The X axis acceleration is set to 20 mm/sec.

Deceleration setting (axis[].dec)
 Set the deceleration for the axis specified in Word1. Set the deceleration [mm/sec] in Word3.

Example: axis[X].dec( 20 ) The X axis deceleration is set to 20 mm/sec.



•Relative movement (axis[].incmove)

Perform relative movement of the axis specified in Word1. Set the moving distance [mm] in Word3 and "MotionWait" in Word4. If "MotionWait" is not set, Word4 can be closed with parentheses ")". Specify the Floating type for Word3 and the Boolean type for Word4 for calculation.

Example: axis[X] .incmove( 5.46 True )

Move X axis +5.46 mm from the current position. The next line cannot be executed until the movement is completed.

```
Example: axis[X] .incmove( 5.46 False )
sleep( 30 )
axis[Y] .incmove( 3.6 )
multi .motionwait( axis[X] axis[Y] )
```

Starts moving to a position +5.46mm from the current position on the X axis, 30msec later starts moving to a position +3.6mm from the current position on the Y axis, and waits until each move is completed.

Absolute movement (axis[].absmove)

Perform absolute movement of the axis specified in Word1. Set the moving distance [mm] in Word3 and "MotionWait" in Word4. If "MotionWait" is not set, Word4 can be closed with parentheses ")". Specify the Floating type for Word3 and the Boolean type for Word4 for calculation.

Example: axis[X] .absmove( 5.46 True )

Move X axis +5.46 mm from the origin position. The next line cannot be executed until the movement is completed.

Example: axis[X] .absmove( 5.46 False ) axis[Y] .incmove( 3.6 True ) axis[X] .motionwait ()

Starts moving to a position +5.46 mm from the X-axis home position (coordinate 0), and immediately starts moving to a position +3.6 mm from the current position on the Y-axis. It waits until the Y-axis move is completed, and then waits until the X-axis move is completed in the next line.

Jog movement (axis[].jogmove)

Perform JOG movement of the axis specified in Word1. Set the moving direction in Word3.

"True" for moving in positive direction and "False" for negative direction.

\*Movement continues until a deceleration stop command is issued. However, it stops when an emergency stop or limit signal is detected.

```
Example: axis[X] .jogmove( False )
sleep( 150 )
axis[X] .stop()
Move in the negative direction for 150ms.
```



•Deceleration stop (axis[].stop) Perform deceleration stop for the axis specified in Word1.

Example:axis[X].stop() Decelerates the X axis to a stop.

•Wait for movement completion (axis[].motionwait) Execute the next line after the movement completion of the axis specified in Word1.

Example:axis[X] .motionwait() Wait until the completion of the X axis movement.

Acquisition of setting speed (axis[].getspeed()

Return the setting sped of the axis specified in Word1.

Assign the starting speed (initial speed) [mm/sec] to Word3, the maximum speed [mm/sec] to Word4, and the acceleration [mm/sec] to Word5.

Specify the Floating type for calculation.

Example:

axis[X] .speed( 0.5 5.0 20 ) axis[X] .getspeed( LowSpeed HighSpeed Rate ) Obtain the speed setting for the X axis. Assign 0.5 to LowSpeed, 5 to HighSpeed, and 20 to Rate.

•Acquisition of current position (axis[].getposition)

The current position (in mm) of the axis specified in Word1 is assigned to Word3. Word3 is specified to Floating type for calculation.

Example:axis[X] .getposition( Floating01 ) The current position of "X" axis is assigned to Floating01.

•Positive direction movement limit setting (axis[].limitplus) Set the soft limit position [mm] in positive direction for the axis specified in Word1.

Example: axis[X] .limitplus( 115.4 )

Set the position 115.4 mm from the X-axis origin position as a soft limit in the positive direction.

Negative direction movement limit setting (axis[].limitminus)

Set the soft limit position [mm] in negative direction for the axis specified in Word1.

Example: axis[X] .limitminus( -1.58 )

Set the position -1.58 mm from the X-axis origin position as a soft limit in the negative direction.



•Relative Position Setting (multi.setinc)

Set the relative position [mm] of the axis specified in Word3 with the value of Word4.

Relative movement for multiple axes (multi.incmove)
 Move all axes, whose relative positions are set, to the relative position. Set MotionWait to Word3.
 Specify Boolean type for Word3 for calculation.

•Reset relative position for all axes (multi.increset) Reset the relative position of all axes, whose relative positions are set, to zero.

Example: multi .setinc( axis[X] 5.46 ) multi .setinc( axis[Y] 3 ) multi .setinc( axis[Z] -25.4 ) multi .setinc( axis[CZ] 35.4) multi .incmove( True ) multi .incmove( True ) ←Move again after completion of movement multi .increset()

Relative movements of 5.46 mm for the X axis, 3 mm for the Y axis, -25.4 mm for the Z axis, and 35.4 mm for the CZ are performed two times.

Reset the position setting of four axes after completion of all axes movement.



Absolute position setting (multi.setabs)

Set the absolute position [mm] of the axis specified in Word3 with the value of Word4.

Absolute movement for multiple axes (multi.absmove)
 Move all axes, whose absolute positions are set, to the position from the origin position.
 Set MotionWait to Word3.

Specify Boolean type for Word3 for calculation.

·Reset of absolute movement for all axes (multi.absreset)

Reset the absolute position of all axes, whose absolute positions are set, to zero.

Example:

- multi .setabs( axis[X] 2.46 )
- multi .setabs( axis[Y] 0.5 )
- multi .setabs( axis[Z] 10.6 )
- multi .setabs( axis[CZ] 35.4)
- multi .absmove( True )
- multi .absreset()

Move to absolute position of 2.46 mm for the X axis, 0.5 mm for the Y axis, 10.6 mm for the Z axis, and 35.4 mm for the CZ.

Reset the position setting of four axes after completion of all axes movement.



Immediate stop for all axes (multi.allstop)
 Perform an immediate stop for all axes being driven.

Example: multi.allstop()

•Deceleration stop for multiple axes (multi.stop)

Perform deceleration stop for axes specified in Word3 to Word6. Can be close with parentheses ")" if the quantity of the axis is less than 4.

Example: multi .stop( axis[X] axis[Y] ) Perform deceleration stop for X axis and Y axis.

Example: multi .stop( axis[X] axis[Y] axis[Z] axis[CZ] Perform deceleration stop for X axis, Y axis. Z axis and CZ axis.

•Return to machine homing for all axes (multi.allorigin)

Performs a machine homing move for all axes. The contents of the operation are the same as those of the "machine origin" button on the main screen.

Example: multi .allorigin()

·Acquisition the status of machine homing for all axes (multi.getallorigin)

Obtains the status of the machine homing operation for all axes since the application was launched. The acquired content is assigned to Word3. If the acquired value is True, the operation is completed. If the machine homing operation has not been performed or if the machine is stopped in the middle of the machine homing operation, the value is False.

Example : multi .getallorigin( StatusOrg ) if ( StatusOrg != True ) multi .allorigin() endif Machine homing is executed when machine homing is not performed for all axes.

•Wait until all axes have completed moving (multi.allmotionwait) Wait until all axes have been driven.

Example : multi .allorigin() multi .allmotionwait() Performs machine homing for all axes and waits until the operation is complete.



•Wait until the specified axis have completed moving (multi.motionwait)

Wait until the movement of the axis specified in Word3 to Word6 is completed. Upon completion, the next line is executed. If there is no axis to specify, close with parentheses ")". Example: multi.motionwait (axis[X] axis[Y])

Wait until X-axis and Y-axis move is completed.

Example: multi .motionwait ( axis[X] axis[Y] axis[Z] axis[CZ] Wait until X, Y, Z, and CZ axes have completed their travel.

Acquisition of emergency stop signal (axis[].getemg)

Obtain the states of the emergency stop input signal (EMG) from the interface A connector of the driver box SDB-08. The acquired state is assigned to Word1.

\*The states are same for all connected axes.

Example: Boolean01 = axis[X] .getemg() The state of the X axis emergency stop signal is assigned to Boolean01.

•Acquisition of stop signal (axis[].getstop)

Obtain the logical OR of the states of three axis stop signal (STOP0, STOP1, and STOP2) from the interface A connector of the driver box SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] getstop() The state of the X axis stop signal is assigned to Boolean01.

Acquisition of stop signal (axis[].getstop0)

Obtain the state of the axis stop signal (STOP0) from the interface A connector of the driver box SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] .getstop0() The state of the X axis stop signal is assigned to Boolean01.

Acquisition of stop signal (axis[].getstop1)

Obtain the state of the axis stop signal (STOP1) from the interface A connector of the driver box SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] .getstop1() The state of the X axis stop signal is assigned to Boolean01.



•Acquisition of stop signal (axis[].getstop2)

Obtain the state of the axis stop signal (STOP2) from the interface A connector of the driver box SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] .getstop2() The state of the X axis stop signal is assigned to Boolean01.

Acquisition of drive operation signal (positive direction) (axis[].getdriveplus)
 Obtain the state of the axis driving operation signal (P+P) from the interface A connector of the driver box
 SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] .getdriveplus() The state of the X axis driving operation signal is assigned to Boolean01.

Acquisition of drive operation signal (negative direction) (axis[].getdriveminus)
 Obtain the state of the axis driving operation signal (P-N) from the interface A connector of the driver box
 SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] .getdriveminus() The state of the X axis driving operation signal is assigned to Boolean01.

Acquisition of positioning completion (axis[].getinposition)

Obtain the state of the axis positioning completion signal (INPOS) from the interface A connector of the driver box SDB-08. The axis is specified in Word3. The result is assigned to Word1.

\*The INPOS signal is for servo motor. It is not used for stepping motors.

Example: Boolean01 = axis[X] .getinposition() The state of the X axis positioning completion signal is assigned to Boolean01.

•Acquisition of alarm status (axis[].getalarm)

Obtain the state of the axis alarm signal (ALARM) from the interface A connector of the driver box SDB-08. The axis is specified in Word3. The result is assigned to Word1.

Example: Boolean01 = axis[X] .getalarm() The state of the X axis alarm signal is assigned to Boolean01.

Display the JOG screen (jogshow)

The jog operation screen appears and the execution sequence is paused. Pressing the resume button closes the jog screen and restarts the execution sequence.



## 4-6. Multiple I/O (Multi I/O)

·General output signal setting (dio.output)

Output the signal with the signal name specified in Word3 from the general-purpose I/O connector of the driver box SDB-08.

If the value of Word5 is True, the output is On. If the value of Word5 is False, the output is Off.

The value of Word4no is 0 for the first SDB-08 unit (Axis0 to Axis7) and 1 for the second unit (Axis8 to Axis15).

Example: dio .output( BYOUT2 0 True ) Set BYOUT2 signal output of the first SDB-08 to be On.

Acquisition of input analog value (analog.getvalue)

Outputs the input value of the A/D analog signal.

Set the value of "SET No." on A/D setting screen to Word3 and the value of "setting" to Word4.

The calculation result on A/D setting screen is assigned to Word5. The voltage is not assigned.

Specify Floating type for Word5 for calculation.

Example: analog .getvalue( 1 2 Floating01 )

The calculation result of the input analog value set in "SET No.1" and "Setting 2" is assigned to Floating01.



#### 4-7. Communication (Interface)

Serial Communication

·Connect to the port (serial.connect)

Connect to the serial port specified in Word3. Without the connection, data communication error occurs and the sequence execution is terminated. <u>The connected serial port must be disconnected by use of "serial.disconnect"</u> <u>command until the sequence execution is finished.</u>

Example: serial .connect( com[SPS-SLSI] ) Connect to the serial port named "SPS-SLSI" set in serial setting screen.

•Send text data (serial.send)

Send the value in Word4 to the serial port specified in Word3. The delimiter is assigned during the sending. Specify String type for Word4 for calculation.

Example: serial .send( com[SPS-SLSI] 'EMISS:1' )

Send "EMISS:1" to the serial port named "SPS-SLSI" set in serial setting screen.

·Receive text data (serial.receive)

Receive the data from the serial port specified in Word3 and assign to Word4 with the delimiter removed. Specify String type for Word4 for calculation.

Example:

serial .send( com[SPS-SLSI] 'EMISS:1' )

serial .receive( com[SPS-SLSI] String01 )

Send "EMISS:1" to the serial port named "SPS-SLSI" set in serial setting screen. And receive the data in String01.

Send and receive text data (serial.sendreceive)

Send the value in Word4 to the serial port specified in Word3. The delimiter is assigned during the sending. Assign received data to Word5 with the delimiter removed. Specify String type for Word4 and Word 5 for calculation.

Example: serial .send.receive( com[SPS-SLSI] 'EMISS:1' String01 )

Send "EMISS:1" to the serial port named "SPS-SLSI" set in serial setting screen. And receive the data in String01.

Disconnect the port (serial.disconnect)
 Disconnect the serial port specified in Word3.

Example: serial .disconnect( com[SPS-SLSI] )

Disconnect the serial port named "SPS-SLSI" set in serial setting screen.



#### ■GPIB Communication

·Send text data (gpib.send)

Specify the destination address number in Word3 and send the value in Word4. The delimiter is assigned during the sending. Specify Integer type for Word3 and String type for Word4 for calculation.

Example: gpib.send( 4 '\*IDN?' ) Send the "\*IDN?" command to the device set at address 4.

•Receive text data (gpib.receive)

Specify the destination address number in Word3 and assign the received data to Word4 with the delimiter removed. Specify Integer type for Word3 and String type for Word4 for calculation.

Example: gpib .send( 4 '\*IDN?' ) gpib .receive( 4 String01 ) Send the "\*IDN?" command to the device set at address 4 and receive data at String01.

·Send and receive text data (gpib.sendreceive)

Specify the destination address number in Word3 and send the value in Word4. The delimiter is assigned during the sending. Assign the received data to Word5 with the delimiter removed. Specify Integer type for Word3 and String type for Word4 and Word5 for calculation.

Example: gpib.sendreceive( 4 '\*IDN?' String01 ) Send the "\*IDN?" command to the device set at address 4 and receive data at String01.

•Connect to the board (gpib.open) Connect to GPIB board.

Example: gpib.open()

• Disconnect the board (gpib.close) Disconnect the GPIB board.

Example: gpib.close ()



■Ethernet (TCP/IP) communication

Connect to the network (tcpip.connect)

Connect to the network with address number in Word3 and port number in Word4.

Specify String type for Word3 and Integer type for Word4 for calculation.

The connected network must be disconnected by use of "tcpip.disconnect" command until the sequence execution is finished.

Example: tcpip.connect("192.168.0.1" 2020) Connect to the network with address 192.168.0.1 and port number 2020.

Send text data (tcpip.send)

Send the value in Word3. The delimiter is assigned during the sending. Specify String type for Word3 for calculation.

Example: tcpip .send( "Hello World" CRLF )

Sends a string "Hello World" with carriage return line feed to the connected network.

Receive text data (tcpip.receive)

Assign the received data to Word3 with the delimiter removed.

Specify String type for Word3 for calculation.

Example: tcpip .receive( String01, CR )

Assign the received data from the connected network before the carriage return line feed to String01.

·Send and receive text data (tcpip.sendreceive)

Send the value in Word3. The delimiter is assigned during the sending. Assign the received data to Word4 with the delimiter removed. Specify String type for Word3 and Word4 for calculation.

Example: tcpip .sendreceive( "Hello World" String01 LF )

Sends a string "Hello World" with carriage return line feed to the connected network. Assign the received data before the carriage return line feed to String01.

• Disconnect the network (tcpip.disconnect) Disconnect the connected network.

Example: tcpip.disconnect()



#### 4-8. Measurement (Measure)

•Raster alignment (alignment.raster)

Perform auto-alignment in Raster method. The alignment setting number is specified in Word3. The value can be 1 to 20. Once the measurement is finished, the next line of the sequence command is executed.

Example: alignment .raster(1)

Perform auto-alignment in Raster mode with alignment setting number 1.

·Spiral alignment (alignment.)

Perform auto-alignment in Spiral method. The alignment setting number is specified in Word3. The value can be 1 to 20. Once the measurement is finished, the next line of the sequence command is executed.

Example: alignment.spiral(1)

Perform auto-alignment in Spiral mode with alignment setting number 1.

1-axis line alignment (alignment.line1)

Perform auto-alignment in 1-axis line method. The alignment setting number is specified in Word3. The value can be 1 to 20. Once the measurement is finished, the next line of the sequence command is executed.

Example: alignment.line1(1)

Perform auto-alignment in 1-axis line mode with alignment setting number 1.

•2-axis line alignment (alignment.line2)

Perform auto-alignment in 2-axis line method. The alignment setting number is specified in Word3. The value can be 1 to 20. Once the measurement is finished, the next line of the sequence command is executed.

Example: alignment .line2(1) Perform auto-alignment in 2-axis line mode with alignment setting number 1.

•3-axis line alignment (alignment.line3)

Perform auto-alignment in 3-axis line method. The alignment setting number is specified in Word3. The value can be 1 to 20. Once the measurement is finished, the next line of the sequence command is executed.

Example: alignment.line3(1)

Perform auto-alignment in 3-axis line mode with alignment setting number 1.



Neldermead alignment (alignment.neldermead)

Perform auto-alignment in Neldermead method. The alignment setting number is specified in Word3. The value can be 1 to 20. Once the measurement is finished, the next line of the sequence command is executed.

Example: alignment .neldermead(1)

Perform auto-alignment in Neldermead mode with alignment setting number 1.

•Acquisition of minimum value (alignment.min)

The minimum value among all of the measured data is obtained and assigned to Word5. The coordinate of the minimum value location are assigned to Word3 (for 1Axis) and Word4 (for 2Axis). Specify Floating type for Word3, Word4 and Word5 for calculation.

Example: alignment .min( Floating01 Floating02 Floating03 )

Assign the minimum value of the measurement result to Floating03, the coordinate of 1Axis to Floating01, and the coordinate of 2Axis to Floating02.

·Acquisition of maximum value (alignment.max)

The maximum value among all of the measured data is obtained and assigned to Word5. The coordinate of the maximum value location are assigned to Word3 (for 1Axis) and Word4 (for 2Axis). Specify Floating type for Word3, Word4 and Word5 for calculation.

Example: alignment .max( Floating01 Floating02 Floating03 )

Assign the maximum value of the measurement result to Floating03, the coordinate of 1Axis to Floating01, and the coordinate of 2Axis to Floating02.

Acquisition of barycenter (alignment.moment)

The barycenter of all measured data is obtained and assigned to Word5. The coordinate of the barycenter location are assigned to Word3 (for 1Axis) and Word4 (for 2Axis). Specify Floating type for Word3, Word4 and Word5 for calculation.

Example: alignment .moment( Floating01 Floating02 Floating03 )

Assign the barycenter of the measurement result to Floating03, the coordinate of 1Axis to Floating01, and the coordinate of 2Axis to Floating02.



4-9. File (File)

■Folder Operation

·Obtain folder name (path.dirname)

Obtain the folder name specified in Word5. Assign the obtained string to Word1, or empty string to Word1 if the folder name cannot be obtained. Specify String type for Word1 and Word5 for calculation.

Example: String01 = path .dirname( "c:¥windows¥explorer.exe" ) Assign "c:¥windows" to String01.

Example: String01 = path .dirname( "c:¥windows¥system" ) Assign "c:¥windows" to String01.

·Confirm existence of folder (path.isdir)

Confirm the existence of the folder whose name is specified in Word5. Assign True to Word1 if the folder exists. Otherwise, assign False to Word1. Specify Boolean type for Word1 and String type for Word5 for calculation.

Example: Boolean01 = path .isdir( "c:¥windows¥system" )

Assign True to Boolean01 if the folder exists.

Assign False to Boolean01 if the folder does not exist.

Obtain working folder name (path.getcwd)

Obtain the name of the folder containing the name of the sequence which is being executed. Assign the obtained string to Word1. Specify String type for Word1 for calculation.

Example: String01 = path .getcwd() Assign "C:¥Users¥(User Name)¥Documents¥OptoSigma¥SGALIGN¥sequence" to String01.

• Create folder (path.makedirs) Create new folder with the name in Word3. Specify String type for Word3 for calculation.

Example: String01 = path .getcwd() String01 = String01 + "¥result" path .makedirs( String01 ) Create "C:¥Users¥(User Name)¥Documents¥OptoSigma¥SGALIGN¥sequence¥result" folder

Delete folder (path.rmtree)

Delete the folder whose name is specified in Word3 and all files inside the folder. The deleted files are not moved to the Recycle Bin. Specify String type for Word3 for calculation.

Example: path .rmtree( "C:¥Users¥(User Name)¥Documents¥OptoSigma¥SGALIGN¥sequence¥result" ) Delete "C:¥Users¥(User Name)¥Documents¥OptoSigma¥SGALIGN¥sequence¥result" folder.



#### ■File Operation

•Obtain file name (path.basename)

Obtain the file name specified in Word5. Assign the obtained string to Word1, or empty string to Word1 if the file name cannot be obtained. Specify String type for Word1 and Word5 for calculation.

Example: String01 = path .basename( "c:¥windows¥explorer.exe" ) Assign "explorer.exe" to String01.

Example: String01 = path .basename( "c:¥windows¥system" ) Assign "system" to String01.

\* A empty string "" is returned in the case of "c:¥windows¥system¥".

·Confirm existence of file (path.isfile)

Confirm the existence of the file whose name is specified in Word5. Assign True to Word1 if the file exists. Otherwise, assign False to Word1. Specify Boolean type for Word1 and String type for Word5 for calculation.

Example: String01 = path .isfile( "c:¥windows¥explorer.exe" ) Assign True to Boolean01 if the file exists. Assign False to Boolean01 if the file does not exist.

·Delete a file (path.remove)

Delete the file whose name is specified in Word3. The deleted file is not moved to the Recycle Bin. The file name must be specified as a full path. Specify String type for Word3 for calculation.

Example: path .rmtree( "C:¥Users¥(User Name)¥Documents¥OptoSigma¥SGALIGN¥sequence¥New.json" ) Delete "C:¥Users¥(User Name)¥Documents¥OptoSigma¥SGALIGN¥sequence¥New.json" file.

Copy the file (path.copy)

Copy the contents of a file whose name is specified in Word3 to a new file whose name is specified in Word 4. The file name must be specified as a full path. Specify String type for Word3 and Word4 for calculation.

Example:

String01 = path .getcwd()

String02 = String01 + "¥New.json"

String03 = String01 + "¥Old.json"

path .copy( String02 String03 )

Copy the contents of the file "New.json" to the filer "Old.json".



Join (path.join)

Specify the folder name in Word5 and the file name in Word6. And then join them and assign to Word1. Specify String type for Word1, Word5 and Word6 for calculation.

Example: String01 = "c:¥windows" + "¥explorer.exe" String02 = path .join( "c:¥windows" "explorer.exe" ) Assign "c:¥windows¥explorer.exe" to String01 and String02.

·Load the alignment setting file (settingfile.readalignsettingfile)

Loads and applies the alignment setting file. Used when operating multiple alignment setting files.

The Word3 value should be the full path to the alignment setting file (extension amt).

Example:

String01 = path .getcwd()

Integer01 = String01 .rfind( "¥" )

String02 = String01 0 :: Integer01

String03 = path .join( String02 "parameter"

String04 = path .join( String03 "NoTitle.amt"

Settingfile .readalignsettingfile( String04 )

Generate "C:¥Users¥username¥Documents¥OptoSigma¥SGALIGN¥parametr¥NoTitle.amt" in String04 and load the alignment configuration file "Notitle.amt".



#### ■CSV File

•Open the file (csv.open)

Open the file whose name is specified in Word3. The file name must be specified as a full path. <u>The opened file must be closed by use of "csv.close" command until the sequence execution is finished.</u> The function varies depending on the option specified in Word4.

#### Word4 Options

'w' : Delete the file contents if the file already exists and write new contents to it.Create a new file if the file does not exist.

- 'a' : Append contents to the file if it already exists. Error occurs if the file does not exist.
- 'x' : Error occurs if the file already exists. Create a new file if it does not exist.
- 'r' : Read the contents of the file if it already exists. Cannot write because the file is opened as read-only. Error occurs if the file does not exist.

·Write to the specified cell (csv.setcell)

Specify row number in Word3 and column number in Word4. Row and column numbers start from 1. Error occurs if input 0 or less than 0. Specify the value to be written in Word5.

Specify Integer type for Word3 and Word4, and any variable type for Word5 for calculation.

•Write the one-dimensional array (csv.setlinedata)

Specify the row number in Word3 and the decimal point position in Word4. The column number is fixed to 1.

·Write the two-dimensional array (csv.setarraydata)

Specify the row number in Word3 and the decimal point position in Word4. The column number is fixed to 1.

•Get the value in the cell (csv.getcell)

Specify row number in Word5 and column number in Word6. Row and column numbers start from 1.

Error occurs if input 0 or less than 0. Assign the value in the specified cell to Word1.

Specify String type for Word1, and Integer type for Word5 and Word6 for calculation.

•Close the file (csv.close) Close the opened file.



#### •Tips.

An example for saving a CSV file is shown below.

\* The string with the prefix "#" is the comment or the explanation.

Example: Apply alignment and save the measurement result to CSV file.

# Apply Claster alignment and obtain the maximum value.alignment .raster( 1 )alignment .max( Floating01 Floating02 Floating03 )

```
# Create "result.csv" in operation folder.
String01 = path .getcwd()
String02 = path .join( String01 "result.csv" )
csv .open( String02 'x' )
csv .setcell( 1 1 'X Axis' )
csv .setcell( 1 2 'Y Axis' )
csv .setcell( 1 3 'Max' )
csv .setcell( 2 1 Floating01 )
csv .setcell( 2 2 Floating02 )
csv .setcell( 2 3 Floating03 )
csv .setarraydata( 4 1 )
csv .close()
```



#### ∎Text File

•Open the file(text.open)

Open the file whose name is specified in Word3. The file name must be specified as a full path. <u>The opened file must be closed by use of "text.close" command until the sequence execution is finished.</u> The function varies depending on the option specified in Word4.

#### •Word4 option

- 'w' : Delete the file contents if the file already exists and write new contents to it.Create a new file if the file does not exist.
- 'a' :Append contents to the file if it already exists. Error occurs if the file does not exist.
- 'x' : Error occurs if the file already exists. Create a new file if it does not exist.
- 'r' : Read the contents of the file if it already exists. Cannot write because the file is opened as read-only. Error occurs if the file does not exist.

Write (text.write)

Write the value in Word3 to the opened file. Specify String type for Word3 for calculation.

·Write with line feed (text.writeline)

Write the value in Word3 with line feed appended to the opened file. Specify String type for Word3 for calculation.

Read (text.readline)

Read the text in the opened file line-by-line. Assign the read string to Word1.

| <ul> <li>Close the file (text.close</li> </ul> | ) |
|--|---|
| Close the opened file.                         |   |

Example:

Boolean01 = path .isfile( String01 )

if (Boolean01 = False)

text .open( String01 'x' )

text .write( 'Hello ' )

text .writeline( 'World' )

text .writeline( 'Arignment' )

text .close()

Contents of sample.txt to be created.

Hello World

Arignment

end

Saves a new text file. If a file with the same name exists, it is not written.



# 4-10. Comment (Comment)

Indicate comment line. Nothing is executed in program.

Word1 is fixed to "#" and Word2 is comments. Word 1 is fixed to "#".

Comments are to be written in Word 2 through Word 5.

Adding comment line before the def() command enable tooltips to the function defined by def() command. Only the content of Word2 is added.

\* To activate tooltips, click on the save sequence or check button. If an error occurs in doing so, tooltips will remain disabled.

Example: Add Tips of "Initialize" to the function of "Initialize" defined by def () command.

# Initialization

def Initialize()

return

If a line that calls Initialize() is listed, when moving the mouse over the word "Initialize", a Tips of "Initialize" is displayed.

