

**String**

COLLABORATORS

	TITLE : String		
ACTION	NAME	DATE	SIGNATURE
WRITTEN BY		July 31, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>String</b>	<b>1</b>
1.1	String . . . . .	1
1.2	asc . . . . .	1
1.3	findstring . . . . .	1
1.4	val . . . . .	2
1.5	striplead . . . . .	2
1.6	striptrail . . . . .	2
1.7	len . . . . .	2
1.8	mid . . . . .	3
1.9	chr . . . . .	3
1.10	right . . . . .	3
1.11	left . . . . .	3
1.12	str . . . . .	3
1.13	ucase . . . . .	4
1.14	lcase . . . . .	4
1.15	hex . . . . .	4
1.16	bin . . . . .	4

# Chapter 1

## String

### 1.1 String

PureBasic - String library

String is the way to store a list of characters. With the commands supplied in this library, you can perform some essentials actions on the strings

Commands summary:

- Asc
- Bin
- Chr
- FindString
- Hex
- LCase
- Left
- Len
- Mid
- Right
- Str
- StripLead
- StripTrail
- UCase
- Val

### 1.2 asc

SYNTAX

Ascii = Asc(String\$)

FUNCTION

Return the ascii value of the first string letter.

### 1.3 findstring

---

**SYNTAX**

```
Position = FindString(String$, StringToSearch$, StartPosition.w)
```

**FUNCTION**

Try to find the StringToSearch\$ into the given String\$ starting from the given position. If the string is found, so its position is returned (in character, starting from 1). If the string is not found, it returns NULL.

## 1.4 val

**SYNTAX**

```
Result.l = Val(String$)
```

**FUNCTION**

Transform a string into a numeric value. The string must be an integer in decimal format. ↔

Exemple:

```
Result = Val("1024") ; Result will be filled with 1024.
```

## 1.5 striplead

**SYNTAX**

```
Result$ = StripLead(String$)
```

**FUNCTION**

Remove all the 'space' characters located in front of a string.

## 1.6 striptrail

**SYNTAX**

```
Result$ = StripTrail(String$)
```

**FUNCTION**

Remove all the 'space' characters located at the end a string.

## 1.7 len

**SYNTAX**

```
length = Len(String$)
```

**FUNCTION**

Return the caracter length of the string.

---

## 1.8 mid

### SYNTAX

Result\$ = Mid(String\$, StartPosition, Length)

### FUNCTION

Extract a string of the given Length from the given 'String\$'.

## 1.9 chr

### SYNTAX

Text\$ = Chr(ASCII Value)

### FUNCTION

Return the letter associated to the given ASCII value.

## 1.10 right

### SYNTAX

Result\$ = Righth(String\$, Length)

### FUNCTION

Return the characters from right of the string with the given length. This function doesn't crash if you give incorrect value for the length parameter, it will return the best matching result.

## 1.11 left

### SYNTAX

Result\$ = Left(String\$, Length)

### FUNCTION

Return the characters from left of the string with the given length. This function doesn't crash if you give incorrect value for the length parameter, it will return the best matching result.

## 1.12 str

### SYNTAX

Result\$ = Str(Value)

### FUNCTION

Convert a numeric number into a string.

---

## 1.13 ucase

### SYNTAX

```
Result$ = UCase(String$)
```

### FUNCTION

Return the original string converted in Upper Case characters (if possible). This command also support accent letter, so if an 'é' is found, it will be tranformed into Upper 'É'.

## 1.14 lcase

### SYNTAX

```
Result$ = LCase(String$)
```

### FUNCTION

Return the original string converted in Lower Case characters (if possible). This command also support accent letter, so if an Upper 'É' is found, it will be tranformed into 'é'.

## 1.15 hex

### SYNTAX

```
Result$ = Hex(Number)
```

### FUNCTION

Return a 8 characters string which represent the hexadecimal value of the given number.

## 1.16 bin

### SYNTAX

```
Result$ = Bin(Number)
```

### FUNCTION

Return a 32 characters string which represent the binary value of the given number.

---