

VARIABLES

```
VAR_GLOBAL
GDivisor : REAL; (* Valore divisore (Globale) *)
GValue : REAL; (* Valore (Globale) *)
END_VAR
```

	Project : FunctionHowTo	
	VARIABLES :	
	Release :	Ver :1.00
	Author :	Date:05/05/2015
	Note :	Page:1 of 1

```
VAR
Result : ARRAY[ 0..2 ] OF REAL; (* Risultato *)
Value : REAL := 100.0; (* Dividendo *)
FBUse : MyFB; (* Istanza FB *)
Compare : BOOL; (* Risultato comparazione *)
END_VAR
```

```
1 (* ***** *)
2 (* PROGRAMMA DI ESEMPIO *)
3 (* ***** *)
4 (* Esempio di utilizzo funzioni e FB. *)
5 (* ----- *)
6 (* Chiamata funzione. *)
7
8     GDivisor:=100.0; (* Valore divisore (Globale) *)
9     Result[0]:=MyFun(Value, 10.0); (* Risultato *)
10    Result[1]:=MyFun(Value, GDivisor); (* Risultato *)
11
12 (* Chiamata blocco funzione. *)
13
14    GDivisor:=1000.0; (* Valore divisore (Globale) *)
15    FBUse.Dividend:=500.0; (* Dividendo *)
16    FBUse(); (* Esegua FB *)
17    Result[2]:=FBUse.Result; (* Risultato *)
18
19 (* Chiamata funzione comparazione. *)
20
21    IF (SysFirstLoop) THEN GValue:=100.0; END_IF; (* Valore (Globale) *)
22    Compare:=MyFunPtr(ADR(GValue), 50.0); (* Risultato comparazione *)
23
24 (* [End of file] *)
25
```

	Project : FunctionHowTo	
	PROGRAM : ST	
	Release : FunctionHo	Ver :1.00
	Author :	Date:05/05/2015
	Note :	Page:1 of 1

```
VAR_INPUT
Dividend : REAL; (* Dividendo *)
END_VAR
```

```
VAR_OUTPUT
Result : REAL; (* Risultato *)
END_VAR
```

```
VAR_EXTERNAL
GDivisor : REAL; (* Valore divisore (Globale) *)
END_VAR
```

```
1 (* ***** *)
2 (* FUNCTION BLOCK "MyFB" *)
3 (* ***** *)
4 (* Questo blocco funzione esegue la divisione tra due numeri. *)
5 (* ----- *)
6 (* Eseguo divisione. *)
7
8     Result:=Dividend/GDivisor; (* Risultato *)
9
10 (* [End of file] *)
11
```

Project : FunctionHowTo

FUNCTION_BLOCK : MyFB

Release : FunctionHo

Ver :1.00

Author :

Date:05/05/2015

Note :

Page:1 of 1

```
VAR_INPUT
Dividend : REAL; (* Dividendo *)
Divisor  : REAL; (* Divisore *)
END_VAR
```

```
1 (* ***** *)
2 (* FUNCTION "REAL MyFun(REAL Dividend, REAL Divisor)" *)
3 (* ***** *)
4 (* Questa funzione esegue la divisione tra due numeri. *)
5 (* ----- *)
6 (* Eseguo divisione. *)
7
8     MyFun:=Dividend/Divisor; (* Function result *)
9
10 (* [End of file] *)
11
```

	Project : FunctionHowTo	
	FUNCTION : MyFun	
	Release : FunctionHo	Ver :1.00
	Author :	Date:05/05/2015
	Note :	Page:1 of 1

FUNCTION MyFunPtr

```
VAR_INPUT
Value : @REAL; (* Valore *)
Threshold : REAL; (* Soglia *)
END_VAR
```

```
1 (* ***** *)
2 (* FUNCTION "BOOL MyFunPtr(@REAL Value, REAL Threshold)" *)
3 (* ***** *)
4 (* Questa funzione esegue la comparazione tra due numeri. *)
5 (* ----- *)
6 (* Eseguo comparazione. *)
7
8     MyFunPtr:=FALSE; (* Function result *)
9
10    IF (@Value > Threshold) THEN
11        MyFunPtr:=TRUE; (* Function result *)
12    END_IF;
13
14 (* [End of file] *)
15
```

Project : FunctionHowTo

FUNCTION : MyFunPtr

Release : FunctionHo

Ver :1.00

Author :

Date:05/05/2015

Note :

Page:1 of 1