format

Synopsis

```
Defun: format(
       String, $str,
       Any, $value )
@{...} String;
```

Description:
Returns a formatted string, using C-style formatted specifications.
The first argument, $str, is a string containing a single C "%" **format** specification. For Integers, use "%xd", where x is any legal C integer **format** specifier (including none). For Numbers, use "%xf" where x is any legal C floating point **format** specifier.
The second argument, $value, is currently restricted to Integers or Numbers.
Use **Format** for formatting only a single number, then concatenate strings together using the '+' operation.

Examples:

```
(String) fNum: format("%f"; myNumber:);
(String) longString: "Total of " + format("%d"; n_items:) + " ordered.";
```

Formatting
Examples:

	Example	Result
Formatting strings	**format**("%8s","aaa")	" aaa"
	format("%-8s","aaa")	"aaa "
Formatting integers	**format**("%8d",123)	" 123"
	format("%08d",123)	"00000123"
	format("%d,123)	"123"
	format("%d",123.456)	"123"
Formatting numbers	**format**("%8.1f",12.345)	" 12.3"
	format("%08.1f",12.345)	"000012.3"
	format("%8.0f",12.345)	" 12"
	format("%#8.0f",12.345)	" 12."
	format("%8.1e",334.3434)	"3.3e+002"
	format("%8.1E",334.3434)	"3.3E+002"
Adding spaces and text	**format**(" %4d",12)	" 12"
	format(" %4d ",12)	" 12 "
	format("X=%6.2f",12.1111)	"X= 12.11"
	format("%5.1f mm",12.1111)	" 12.1 mm"

Input Arguments:
str - Specify the string
value - Specify a value

Returns:
String - A formatted number string

See Also: