@ Special Functions

#NOT# arg
This is the logical NOT function, but it is not used the same way as the logical AND and OR. For example, #NOT# 1 will equal 0, while #NOT# 0 will equal 1.
A3 = +("Blue"="Red") , which returns 0 #NOT#A3 = 1 : The formula in cell A3 asks "is the label 'Blue' = the label 'Red'? The answer is 0 (false). The #NOT# function inverts this returning the answer 1 (true)

#NOT#1 = 0 (false)

@@(string)
Evaluates string as a cell address and returns that address.
A5 = Cheers, B3 = A3, @@(B3) = Cheers

A1 = 10, A2 = 20, A3 = 30, B1 = A1:A3, @SUM(@@(B1)) = 60

@BEEPIF(cond)
Beeps if the condition is true and returns 1, 0 if false.
C7 = 100 @BEEPIF(C7<200) : This formula will beep one time.

@BEEPIF(@rand>0.5) : This formula will generate a random number between 0 and 1. If the number is > 0.5, the it beeps and returns 1. Otherwise it returns 0.

@CELL(atrb,adr)
Returns information about a the cell at adr. atrb defines the type of information requested.
@CELL("address",U7) = U7
@CELL("row",U7) = 7
@CELL("col",U7) = 21
U7 = @PI, @CELL("contents",U7) = 3.141593
ADDRESS - returns the address of the cell
ROW - returns the row of the cell
COL - returns the column of the cell
CONTENTS - returns the contents of the cell
FORMAT - returns the format of the cell. Formats are:

- G - General
- Fn - Fixed Decimal n places
- Sn - Scientific Format n decimal places
- Cn - Currency format n decimal places
- ,n - Comma format n decimal places
- + - Chart Format
- Pn - Percent Format n decimal places
- T - Text Format
- H - Hidden Format
- D1 - Day Month Year Format
- D2 - Day Month Format
- D3 - Month Year Format

D6 - Hours Minutes Seconds Format
D7 - Hours Minutes Format
D4 - Int'l Date Format 1
D5 - Int'l Date Format 2
D8 - Int'l Time Format 1
D9 - Int'l Time Format 2
PREFIX - The string prefix in Lotus Format
PROTECT - Returns 0 if the cell is Unprotected, 1 otherwise
TYPE - b - blank cell, v - number value, l - string value
WIDTH - Returns the width of the column in characters
RANGE - Converts a single address to a range

@CELLPOINTER(atrb)
Returns the value for the attribute of the current cell. This is valuable for macros. See CELL() or @CELL() for information about attributes.

@CHOOSE(num,item1[,item2[,...])
Chooses the num item from the list.
@CHOOSE(3,"first","second","third","fourth") = third

@CHOOSE(@MOD(@INT(@NOW)),7),"Thu","Fri","Sat",Sun","Mon","Tue","Wed") : This function returns the current day using @NOW to provide the current date.

@COLS(range)
Returns the number of columns in a range.
@COLS(A1:H20) = 8

If RANGE1 is the name assigned to the range B1:CC200, @COLS(RANGE1) = 80

@HLOOKUP(code,table,offset)
Looks up code in the horizontal table. Returns the cell offset cells from the top of the first match. (See "The @HLOOKUP Function" on page55.)
If the first row contains numbers and code is less than the first value in the lookup row, !ERR is returned. If code is greater than the first value but no exact match is found, it returns the value from the last column.
If the first row contains only strings and no exact match is found, it returns !ERR.

A1 = 1 A2 = 2 A3 = 3
B1 = $200 B2 = $400 B3 = $500

@HLOOKUP(2,A1:B3,1) = $400
@HLOOKUP(0.5,A1:B3,1) = !ERR
@HLOOKUP(5,A1:B3,1) = $500

@IF(exp,value1[,value2])
Evaluates exp. If it is a non-zero value, the result is value1. If it is a zero or a string, optional value2 is the result; otherwise, a blank is the result.

@IF(5>10,"Something is wrong!","Everything is normal.") = Everything is normal.

A1 = 3400, A2 = 4000, A3 = 5000, @IF(@SUM(A1:A3)>12400,"Full Capacity") = Full Capacity

@INDEX(range,col,row)
Returns the cell that is row rows and col columns offset from the upper left corner of range.
A1 = 10, B1 = 20, C1 = 30
A2 = 3942, B2 = 4932, C2 = 5929
A3 = 4920, B3 = 5929, C3 = 9294
@INDEX(A1:C3,2,2) = 9294

A10 = 0, A11 = 1, @INDEX(A1:C3,A10,A11) = 3942

@ISERR(exp)
Returns 1 if exp returns an error other than @NA. Returns 0 otherwise.
@ISERR(@ERR) = 1 (true)

@ISERR(@VALUE("ABCD") = 1 (true) : This formula tried to convert "ABCD" to a value using @VALUE(). This returns !ERR, which causes @ISERR() to returns 1.

@ISNA(exp)
Returns 1 if exp returns an @NA error. Returns 0 otherwise.
@ISNA(@NA) = 1 (true)

A1 = 34, A2 = 49, A3 = @NA, @ISNA(@SUM(A1:A3)) = 1 : This formula returns 1 (true) because cell A3 contains @NA, which causes @SUM(A1:A3) to evaluate to @NA also.

@ISNUMBER(exp)
Returns 1 if exp returns a numeric result. Returns 0 otherwise.
@ISNUMBER(@STRING(300,0)) = 0 : This formula converts the label "300" to a string using @STRING(). It then uses @ISNUMBER() to test if that result is a value and returns 0 (false).

@ISNUMBER(34) = 1

A3 = +394 - 100, @ISNUMBER(A3) = 1

@ISREF(exp)
Returns 1 if exp is a cell or range reference. Returns 0 otherwise.
@ISREF(A1) = 1 (true)

If RANGE1 has been assigned A1:C10, @ISREF(RANGE1) = 1

@ISSTRING(exp)
Returns 1 is exp results in a string. Returns 0 otherwise.
@ISSTRING(@VALUE("340")) = 0
@ISSTRING(@LEFT("Goodbye!",4)) = 1

@ISSTRING("Random text") = 1

@N(exp)
If exp is a number, @N() returns exp, otherwise @N() returns 0. @N() is guaranteed to return a number and is used in functions that require numbers as parameters.
@N(@STRING(300,0)) = 0
@N(34) = 34

@NEXT(value[,inc])
Returns the next logical value after value. For example, if value = "Mon", then @NEXT() returns "Tue". Values for months, days, and quarters are built in. If value is a number, @NEXT() adds 1 or inc to the number. For quarterly information, it will increment the quarter. When the quarter reaches 4, the next quarter is Q1 of the next year. For example, @NEXT("Q4 `92") is "Q1 `93".
@NEXT("Q4 '92") = Q1 '93
@NEXT("January") = February
@NEXT("Region 1") = Region 2

@NOT(num)
If num is zero or not a number, 1 is the result; otherwise, 0 is the result.
@NOT(34) = 0
@NOT("Text") = 1

@ROWS(range)
Returns the number of rows in a range.
@ROWS(A1:H20) = 20
If RANGE1 is the name assigned to the range B1:CC200, @ROWS(RANGE1) = 200

@S(exp)
If exp is a string, returns exp; otherwise it returns a zero length string. This function is used to guarantee that an expression is a string.
@S(343) = "" (blank string)
A1 = bob, A2 = jones, @S(@PROPER(A1&" "&A2)) = Bob Jones : This formula uses & to concatenate cells A1 and A2. @PROPER() is used to capitalize the first letter in each word and @S() insures the result is a string.
A1 = 2000, A2 = Orders @S(A1)&@S(A2) = Orders

@SAME(adr)
Takes the formula in cell adr and executes it as if it had been copied and then pasted into the current cell. This is a powerful and valuable function. You can simply have a master function and a series of functions that are the same. Where you change the master function, the method of calculation used by the other functions is changed. You can use @SAME() in conjunction with INDEX(), @INDEX(), HLOOKUP(), VLOOKUP(), @HLOOKUP(), and @VLOOKUP() to define a formula used in a calculation rather than having a complex @IF() statement. An application for this is an employee pay calculation that depends on an employee type.
A1 = 1, A2 = 2

If B1 = A1 * 5 and B2 = @SAME(B1), then B2 = 10 (cell A2*5).
If B1 = A1/5, then B2 changes to 0.4 (cell A2/5)

If B1= @SIN(A1), B2 changes to 0.9093 (@sin(A2))

@SIGNAL(cond,port,range)
If the condition is true, sends range of cells to the named Mach port.

@VLOOKUP(code,table,offset)
Looks up code in the vertical table. Returns the cell offset cells from the left side of the first match.
If the first column contains only strings and code is a number, it returns the value from the last row. If code is also a string, but doesn't match any of the table entries, !ERR is returned.

A1=A B1=5.0
A2=B B2=4.0
A3=C B3=3.0
A4=D B4=2.0
A5=F B5=0

@VLOOKUP("A",A1:B5,1) = 5.0
@VLOOKUP("D",A1:B5,1) = 2.0
@VLOOKUP("Incomplete",A1:B5,1) = !ERR
@VLOOKUP(3.0,A1:B5,1) = 0