

# Functions Glossary

**W**ithout functions and formulas, spreadsheets are little more than fancy sheets of graph paper, and databases are little more than boxes of index cards. Add functions and formulas, however, and the power of spreadsheets and databases increase exponentially.

---

**Note:** This special online appendix is designed to accompany the physical book *AppleWorks 6: The Missing Manual*, by Jim Elferdink and David Reynolds. For details on the book, visit [www.missingmanual.com](http://www.missingmanual.com).

---

## About Functions

Think of a function as the really smart kid who sat next to you in Algebra who could give you the answer to a problem with very little effort on your part. That's the beauty of functions. They take the bits of data that you feed them, perform some sort of calculation on it, and hand you back the results.

That's not to say, however, that functions are easy. Most of them are intended for advanced users: the mathematically inclined or people who already have experience building complex spreadsheets.

---

**Tip:** The AppleWorks online help includes its own listing of functions, which offers further examples and help. Choose Help→AppleWorks Help; click "Functions for spreadsheets and databases."

---

You can use functions and formulas in two kinds of documents (and one kind of frame):

- **Spreadsheets.** To use a function in a spreadsheet or spreadsheet frame, just type it into the cell where you want the result to appear; as noted in Chapter 5, you must begin a formula with an equal sign (=). Figure C-1 gives an example.

Most functions, by the way, include one or more *arguments* inside parentheses. The argument is the text or number(s) that you feed into the function. For example, if you want to add up cells A1, A2, and A3, you could type, into another cell, this formula: =SUM(A1..A3)

(The SUM function uses two periods to indicate a range of cells.)

- **Databases.** To use a function in a database, create a Calculation field (see page 104) that will hold your formula. To do this, choose the Calculation Field Type in the Define Database Fields window, give it a name, and click Create. You'll be presented with a window where you can construct a formula for your Calculation field. The right side of the window has a list of all of the formulas that you can use in your database.

You may notice that AppleWorks automatically puts single straight quote marks, 'like this', around each field name that you use in a formula. That's a required format; if you don't use those single quotes, the function won't work. (Be careful not to use curly quotes, 'like this'. They'll break your database calculations.)

## **Kinds of Functions**

You'll note that the functions described in this appendix fall into one of several categories:

- **Financial.** These are the functions you'll use if you're trying to calculate loans and investments. You're expected to know such terms as *net present value*, *number of periods*, *rate of return*, and so on.
- **Date and Time.** You can use AppleWorks to calculate elapsed time, round up invoices that haven't been paid in 30 days, and so on. As you read the descriptions in this appendix, you'll quickly realize that AppleWorks has a funny way of thinking of time: It treats any date as a *serial number* that represents the number of days before or after January 1, 1904 (which the Macintosh clock considers the beginning of time). The left side of the number is the number of days; any decimal amount to the right of the decimal point is the fraction of a day (which the functions described in this appendix can convert to hours, minutes, and seconds).

You can use this kind of serial number when creating complex date calculations, such as invoice aging, library-book due dates, and so on. Most of the date and time functions require that you feed them dates in this serial-number format (or convert standard dates into or out of the serial number format).

- **Information.** You can use these functions to manipulate AppleWorks itself: make it beep, display an error message, play a macro or trigger a Button bar button, and so on.

- **Logical.** These advanced functions let you build complex nested functions. You can use the IF, AND, OR, and NOT functions, for example, to direct your spreadsheet to “Add this number to column A if it’s *either* more than 2000 *or* evenly divisible by 7,” for example.
- **Numeric.** These functions are the bread-and-butter math operations for spreadsheets. They can calculate integers, random numbers, square roots, absolute value, and so on, and operate only on number cells (or number fields).
- **Statistical.** The ever-popular Sum, Average, Count, and Standard Deviation functions are among those on this list; all of these, too, operate exclusively on cells or fields containing numbers.
- **Text.** These functions perform various transformations to *text* cells or fields. For example, they can combine the First Name and Last Name fields into one combined Full Name field, capitalize or un-capitalize some text, extract the middle letters from a field, tell you how many characters are in a field (see Figure C-1), and so on.
- **Trigonometric.** Once again, these functions require that you feed them numbers only. They include such trig favorites as Arc Cosine, Arc Tangent, Cosine, Degrees, Radians, Sine, and so on.

Here’s the complete list of functions, what they should look like, and what they do. In the examples here, *italics* represent arguments that you should replace with your own cell or field references. The title of each function shows both its abbreviation, such as ABS (which is what you actually type into your spreadsheet or choose from the list of database functions), as well as its more helpful full name in parentheses, such as (Absolute Value).

**Figure C-1:**

*Every time you type a state’s name, the spreadsheet manages to tell you exactly how many letters are in its name (top). How does it know? If you ask AppleWorks to reveal the behind-the-scenes formulas responsible for this auto-calculation (bottom), all becomes clear. Each AppleWorks cell in the right-hand column contains a function (in this case, the LEN function) to process the cells in the left-hand column.*

State	Number of Letters
Ohio	4
Mississippi	11
Florida	7
North Carolina	14

State	Number of Letters
Ohio	=LEN(C13)
Mississippi	=LEN(C14)
Florida	=LEN(C15)
North Carolina	=LEN(C16)

---

**Tip:** In the language of functions, you say that a function *returns* a value. For example, in Figure C-1, you'd say that the first "Number of Letters" cell *returns* the value 4, after processing the *argument* in cell C13 (which is "Ohio"). In the next cell down, the argument is "Mississippi," and the LEN function returns the value 11.

---

## ABS (Absolute Value)

Returns the *absolute value* of the number in the parentheses. This function, in other words, turns any number (except zero) into a positive one, even if it's a negative number. (If the cell or field contains 0 or is empty, this function returns 0.) Use ABS to calculate the absolute value of a spreadsheet cell address, a number, or a database field, ignoring the number's sign. Use this function to reveal the magnitude of change, whether up or down.

### Format

ABS(*number*)

### Spreadsheet example

Formula	Result
=ABS(-6)	6
=ABS(8.2-8.2)	0
=ABS(22-33)	11

### Database example

Formula	Result
ABS('Weight before'-'Weight after')	The amount of change between the two Weight fields. (The result doesn't tell you the direction of the change, up or down—just the difference.)

## ACOS (Arc Cosine)

Returns the arc cosine of the number in parentheses (which must be between 1 and -1) as an angle in radians.

### Format

ACOS(*number*)

### Spreadsheet example

Formula	Result
=ACOS(0.2)	1.369438406
=ACOS(1)	0
=ACOS(-0.7)	2.3461938234

## Database example

### Formula

ACOS('Number 1')

### Result

Calculates the Arc Cosine of the number entered in the field 'Number 1'.

## ALERT (Alert)

Using this spreadsheet-only function, you can make your spreadsheet behave more like software written by a software company. It makes a dialog box appear, containing one of three sets of buttons; the function returns which button was clicked. The message appears when you first create (or subsequently modify) the Alert formula, call it from another function (such as an IF function), choose Calculate→Calculate Now, or open the spreadsheet on a different kind of computer (on a Windows machine if you created the file on a Mac, for example).

Put the message you want between quotes; the *type* (which can be the numerals 1, 2, or 3) determines what set of buttons (Cancel, Yes, No, or OK) will be displayed in the dialog box. (The table below shows what value the function returns when you, or whoever's using your AppleWorks document, clicks one of these buttons.) The Type is optional—if it's left out, you get a simple OK button and the function returns the text of the message.

Type	Buttons	Returns
1	OK (default) Cancel	0 if Cancel is clicked 1 if OK is clicked
2	Yes (default) No	0 if No is clicked 1 if Yes is clicked
3	OK Cancel (default)	0 if Cancel is clicked 1 if OK is clicked

### Format

ALERT(*message*, *type*)

## Spreadsheet example

### Formula

=IF(A5<20, ALERT("Too low!", 3), A5)

### Result

Pops up a dialog box that reads, "Too low!" with two buttons—OK (which is highlighted) and Cancel. If you click OK, the function returns 1; if you click Cancel, you get 0. And if the value in cell A5 *isn't* too low, nothing happens; the value you typed stays there.

## AND (And)

This logical expression tests one or more values together, testing to see if all of the statements are true.

It works two different ways. If the logical expressions are equations (such as  $1+1=2$ ), it returns TRUE if *all* expressions in the parentheses are true; if the logical expressions are arithmetic operations (such as  $1+1$ ), then if *any* operation is equal to zero, it returns FALSE; otherwise, it returns TRUE. In databases, the return value is a little different—it returns a 0 for FALSE or a 1 for TRUE.

---

**Tip:** In some of the examples below, you'll note that spaces appear after each argument in a list (such as *Sum1, Sum2*); yet in the AppleWorks Help, spaces never appear after the commas in functions (*Sum1, Sum2*). As it turns out, the space is optional; if you find that the spaces make your functions easier to read, use them. The functions work the same way.

---

## Format

AND(*statement1, statement2, ...*)

## Spreadsheet example

Formula	Result
=AND(3+5=8)	TRUE
=AND(3-3, 1+7)	FALSE
=AND(3+2-5)	FALSE

## Database example

Formula	Result
AND('Sum1', 'Sum2')	If both fields Sum1 and Sum2 contain values other than zero, then AND('Sum1', 'Sum2') will return a 1. If either Sum1 or Sum2 are zero, then AND('Sum1', 'Sum2') will return a zero.

## ASIN (Arc Sine)

Returns the arc sine of the number in parentheses (which must be between 1 and -1) as an angle in radians.

## Format

ASIN(*number*)

## Spreadsheet example

Formula	Result
=ASIN(1)	1.5707963268
=ASIN(-0.3)	-0.30469265402

## Database example

Formula	Result
ASIN('Number 1')	If the number in the field Number 1 is 0.83, then ASIN('Number 1') will fill the calculation field with 0.97910768437.

## ATAN (Arc Tangent)

Returns the arc tangent of the number in parentheses as an angle in radians.

### Format

=ATAN(*number*)

### Spreadsheet example

**Formula**

=ATAN(0.25)

=ATAN(3)

**Result**

0.24497866313

1.2490457724

### Database example

**Formula**

ATAN('Number 1')

**Result**

If the number in the field Number 1 is 0.5, then ATAN('Number 1') will fill the calculation field with 0.463647609.

## ATAN2 (Arc Tangent 2)

Returns the angle (in radians) of the line that starts at coordinates 0, 0 and goes through the coordinate specified by x and y in parentheses.

### Format

ATAN2(*x*, *y*)

### Spreadsheet example

**Formula**

=ATAN2(35, -22)

=ATAN2(3, 0)

**Result**

-0.56116341229

0

### Database example

**Formula**

ATAN2('Number 1', 'Number 2')

**Result**

If the number in the field Number 1 is -25 and the number in the field Number 2 is 12, then ATAN2('Number 1', 'Number 2') will fill the calculation field with 2.6940726784.

## AVERAGE (Average)

Returns the arithmetic mean of the values between the parentheses by adding all of the values together and dividing that sum by the number of values. The value in parentheses must be a number, numeric expression, or a cell or cells that contain numbers or numeric expressions.

**Format**

AVERAGE(*value1, value2, ...*)

**Spreadsheet example****Formula**

=AVERAGE(7, 13, 3.43, 2)

**Result**

6.3575

**Database example****Formula**

AVERAGE('Grade1', 'Grade2', 'Grade3')

**Result**

Fills the calculation field with the average of Grade1, Grade2, and Grade3.

**BASETNUM (Base to Number)**

Converts the text (the first item in parentheses) into a base-10 number, using the second item in parentheses as the base. This is useful for converting binary and hexadecimal numbers. A Web master can use this calculation to convert Hex color elements to decimal numbers.

**Format**

BASETNUM(*text, base*)

**Spreadsheet example****Formula**

=BASETNUM("10", 8)

=BASETNUM("FA", 16)

**Result**

8

250

**Database example****Formula**

BASETNUM('StartNumber', 'Base')

**Result**

If the StartNumber text field is 35 and the Base field is 6, then the calculation field will be filled with 24.

**BEEP (Beep)**

Makes your computer beep. This occasionally entertaining function can be useful when embedded in other functions—such as an IF function—to make AppleWorks alert you to something that you should pay attention to (such as an error).

**Format**

BEEP()

## Spreadsheet example

### Formula

=BEEP()  
=IF(A3, "It's true!", BEEP())

### Result

Makes your computer beep.  
If the value in A3 is TRUE, then the function returns the text "It's true!"; otherwise, it makes your computer beep and returns a 0.

## Database example

### Formula

BEEP()

### Result

Beeps once when a new record is created. (Put the BEEP() function into any calculation field to make it work.)

## CHAR (Character)

Returns the ASCII character equivalent to the number in parentheses (which must be between 0 and 255) in the font chosen for this cell or field. The character that's returned depends on the font used, making this function especially useful for finding out which character you need to type in order to produce a certain symbol in, say, the Symbol or Wingdings font.

### Format

CHAR(*number*)

## Spreadsheet example

### Formula

=CHAR(121)  
=CHAR(139+3)

### Result

y  
é

## Database example

### Formula

CHAR('ASCII Number')

### Result

Converts the number entered in the field 'ASCII Number' to its ASCII equivalent and fills the field with that character.

## CHOOSE (Choose)

CHOOSE makes AppleWorks look at a list of values and choose one of them, returning that value. CHOOSE uses the first number as the *index*, which tells the formula *which* of the following values to select and report.

### Format

CHOOSE(*index, value1, value2, ...*)

## Spreadsheet example

**Formula**

```
=CHOOSE(A1, B4, "Bill", 237)
=CHOOSE(1, B4, "Bill", 237)
```

**Result**

Bill (if A1 contains a 2)  
B4

## Database example

**Formula**

```
CHOOSE('Index', 'Phone1',
'Phone2', 'Phone3')
```

**Result**

Uses the number in the field 'Index' to look up the phone number in the fields 'Phone1', 'Phone2', or 'Phone3' and fills the field with that result.

## CODE (Code)

Returns the ASCII code number of the first character in the text in parentheses.

**Format**

CODE(*text*)

## Spreadsheet example

**Formula**

```
=CODE("T")
=CODE("É")
```

**Result**

84  
131

## Database example

**Formula**

```
CODE('Character')
```

**Result**

Fills the calculation field with the ASCII code number for the first character in the field 'Character'.

## COLUMN (Column)

Returns the column number of the cell in the parentheses, as counted starting with column A and moving to the right. It's not available in databases.

**Format**

COLUMN(*cell*)

## Spreadsheet example

**Formula**

```
=COLUMN(B5)
=COLUMN(E17)
```

**Result**

2  
5

## CONCAT (Concatenate)

Returns a string made up of all the text strings in parentheses joined together. To insert literal text, surround it with quotes.

### Format

CONCAT(*text1*, *text2*, ...)

### Spreadsheet example

#### Formula

=CONCAT("Hello", " ", "there!")

#### Result

Hello there!

### Database example

#### Formula

CONCAT('Word1', ", ", 'Word2')

#### Result

Fills the calculation field with the text in the 'Word1' field, a comma and a space, then the text in the 'Word2' field.

## COS (Cosine)

Returns the cosine (a number from -1 to 1) of the number in parentheses. The number in parentheses is an angle, in radians.

### Format

COS(*number*)

### Spreadsheet example

#### Formula

=COS(0.3)  
=COS(-0.732)

#### Result

0.95533648913  
0.74383917362

### Database example

#### Formula

COS('Angle')

#### Result

Calculates the cosine of the number in the Angle field (which is in radians) and fills the field with the result.

## COUNT (Count)

Counts the number of arguments in parentheses and returns that number. If one or more values refer to cells, the cell doesn't get counted if it's empty.

### Format

COUNT(*value1*, *value2*...)

**Spreadsheet example****Formula**

```
=COUNT(2, -17)
=COUNT(32, A5, "Dan")
```

**Result**

2  
3 (However, this will return "2" if cell A5 is empty.)

**Database example****Formula**

```
COUNT('CD1', 'CD2', 'CD3')
```

**Result**

Counts the number of fields ('CD1', 'CD2', 'CD3') that aren't empty and fills the field with that number—in this case, 0, 1, 2, or 3.

**COUNT2 (Count 2)**

Reads the value of the first value in parentheses, and then searches the remaining values for matches to that first value, finally returning the number of matches that it finds.

**Format**

COUNT2(*search value, value1, value2...*)

**Spreadsheet example****Formula**

```
=COUNT2(3, 12, 3, 2, 4, 3, 18, 3)
```

**Result**

3

**Database example****Formula**

```
COUNT2('SearchTerm', 'Item1', 'Item2')
```

**Result**

Takes the item in the 'SearchTerm' field, compares it to the contents of Item1 and Item2, and then fills the field with the number of matches.

**DATE (Date)**

Returns the date serial number for the date in parentheses, as described at the beginning of this appendix.

Use the DATETOTEXT function, described next, to convert the serial number back to human form.

**Format**

DATE(*year, month, day*)

**Spreadsheet example****Formula**

```
=DATE(2000, 3, 10)
```

**Result**

35133

## Database example

**Formula**

DATE('Year', 'Month', 'Day')

**Result**

Calculates a Date serial number using the numbers in the 'Year', 'Month', and 'Day' fields and fills the field with that serial number.

## DATETOTEXT (Date to Text)

Converts a date serial number back into standard date format, according to the format number in parentheses. There are five choices in date format:

**Format code**

0  
1  
2  
3  
4

**How it looks**

1/1/2000  
Jan 1, 2000  
January 1, 2000  
Sat, Jan 1, 2000  
Saturday, January 1, 2000

---

**Tip:** If you're European, fear not: the *sequence* of the day, month, and year in date functions like this one isn't necessarily the American-centric "month, day, year." Instead, it depends on the setting you've made in the Date & Time control panel. The single-digit codes described above specify only the length format (numbers separated by slashes, abbreviated words, written-out words, and so on), not the order.

---

### Format

DATETOTEXT(*serial number*, *format number*)

### Spreadsheet example

**Formula**

=DATETOTEXT(35238, 3)  
=DATETOTEXT(-1176, 4)

**Result**

Fri, Jun 23, 2000  
Friday, October 12, 1900

### Database example

**Formula**

DATETOTEXT('DateSerial', 'Format')

**Result**

Reads the 'DateSerial' field, calculates a date that real humans can read from it, formats it according to the number in the Format field, and fills the calculation field with it.

## DAY (Day)

Returns the day of the month from the date serial number in parentheses.

### Format

DAY(*serial number*)

**Spreadsheet example****Formula**

=DAY(27830)

**Result**

12 (27830 is the serial number for March 12, 1980)

**Database example****Formula**

DAY('DateSerial')

**Result**

Calculates the day of the month from the number in the DateSerial number and fills the calculation field with it.

**DAYNAME (Day Name)**

Returns the day of the week from the number in parentheses. The number must range from 1–7, and it starts with Sunday.

**Format**DAYNAME(*number*)**Spreadsheet example****Formula**

=DAYNAME(3)

=DAYNAME(1)

=DAYNAME(7)

**Result**

Tuesday

Sunday

Saturday

**Database example****Formula**

DAYNAME('DayOfWeek')

**Result**

Reads the 'DayOfWeek' field and fills the calculation with the name of the day of the week that the number represents.

**DAYOFYEAR (Day of the Year)**

Returns the day of the year generated from the date serial number in parentheses.

**Format**DAYOFYEAR(*serial number*)**Spreadsheet example****Formula**

=DAYOFYEAR(14483)

**Result**

239 (The date is Friday, August 27, 1943, which is the 239th day of 1943.)

## Database example

**Formula**

DAYOFYEAR('DateSerial')

**Result**

Reads the value in the 'DateSerial' field, calculates the day of the year, and fills the calculation field with that number.

## DEGREES (Degrees)

Converts an angle in radians (the number in parentheses) into an angle in degrees.

**Format**DEGREES(*number in radians*)

## Spreadsheet example

**Formula**

=DEGREES(5.3)

=DEGREES(-2.15)

**Result**

303.66763142

-123.18592595

## Database example

**Formula**

DEGREES('AngleInRadians')

**Result**

Reads the number in the 'AngleInRadians' field, converts it to degrees, and then fills the calculation field with that number.

## ERROR (Error)

Returns #ERROR! Nothing goes in the parentheses; this function just screams "error!" Useful in logical functions, such as the IF function, to tell you if there's been an error condition.

**Format**

ERROR()

## Spreadsheet example

**Formula**

=ERROR()

=IF(AND(A2), "It's true", ERROR())

**Result**

#ERROR!

Fills the cell with the text "It's true" if cell A2 has any number other than 0 in it, #ERROR! if the cell has a 0 in it.

## Database example

**Formula**

IF(ISBLANK('Text1'), "It's Blank!", ERROR())

**Result**

Checks the 'Text1' field to see if it's blank, and if it is, it fills the calculation field with the text "It's Blank!"; otherwise, it fills the field with #ERROR!

## EXACT (Exact)

Returns TRUE if both values in the parentheses are the same in case and spelling. Use it to see if two text items are an exact match.

### Format

EXACT(*text1*, *text2*)

### Spreadsheet example

#### Formula

=EXACT("Brianna", "Cody")  
 =EXACT("Hello", "hello")  
 =EXACT("Cartman", "Cartman")

#### Result

FALSE  
 FALSE  
 TRUE

### Database example

#### Formula

=EXACT('FirstName1', 'FirstName2')

#### Result

If the contents of the text fields 'FirstName1', 'FirstName2' are exactly the same, then the calculation field is filled with a 1 (meaning they match). If they don't match, then the field is filled with a 0 (meaning they don't match).

## EXP (Exponent)

Returns  $e$  to the power of the number in parentheses. For you non-math heads,  $e$  is a constant approximately equal to 2.71828.

### Format

EXP(*number*)

### Spreadsheet example

#### Formula

=EXP(1.7)

#### Result

5.4739473917

### Database example

#### Formula

EXP('Exponent')

#### Result

Calculates  $e$  to the power of the number in the field "Exponent" and then fills the calculation field with the result.

## FACT (Factorial)

Returns the factorial of the number in parentheses. A *factorial*, for the math disinclined, is the product of all positive integers less than or equal to a given number. For example, the factorial of 3 is  $1 \times 2 \times 3$ , or 6. The number in parentheses has to be a positive integer below 171.

## Format

FACT(*number*)

### Spreadsheet example

**Formula**

=FACT(11)

=FACT(4)

**Result**

39916800

24

### Database example

**Formula**

FACT('PositiveInteger')

**Result**

Calculates the factorial of the number in the 'PositiveInteger' field and fills the calculation field with the result.

## FIND (Find)

Searches the second item in parentheses for the first occurrence of the *first* item in parentheses. Returns a number corresponding to the *position*, as counted by characters, of the found text. (Capitals and spaces count.) If you want to look in a specific text string rather than a field or cell, enclose the actual phrase in quotes.

The optional third item lets you begin the search a certain number of characters in.

### Format

FIND(*find text, in text, start offset*)

### Spreadsheet example

**Formula**

=FIND("S", "My Sharona")

=FIND("S", "Miss Sharona")

=FIND("ss", "Mississippi", 5)

=FIND("ss", "Alabama", 5)

**Result**

4

6 (because that's the first *capital S*)

6

0

### Database example

**Formula**

FIND('ThisString', 'SearchedString', 'Offset')

**Result**

Searches through the text in the field 'SearchedString' for the text in 'ThisString' starting at the character position specified by the number in the 'Offset' field and fills the calculation field with the position of the match (if any).

## FRAC (Fraction)

Returns the fractional part (the number to the right of the decimal) of any number. The result is always positive.

**Format**FRAC(*number*)**Spreadsheet example****Formula**

=FRAC(23.5)

=FRAC(-982.36227)

**Result**

0.5

0.36227

**Database example****Formula**

FRAC('FatDecimalNumber')

**Result**

Looks at the number in the field 'FatDecimalNumber' and strips off everything to the left of the decimal point, filling the calculation field with the part of the number to the right of the decimal.

**FV (Future Value)**

Returns an investment's future value based on several parameters. The first value is the interest rate per period; the second value is the number of periods; the third value is the added payment made per period (entered as a negative number); the optional fourth value is the investment's present value (entered as a negative number); and the optional fifth value is when payments are made—0 for payments at the end of the period, and 1 for payments at the beginning of the period.

**Format**FV(*Rate, NumberOfPeriods, ExtraPayment, PresentValue, Type*)**Spreadsheet example****Formula**

=FV(9%, 5, 0, -10000)

**Result**

\$15,386.24—representing an initial investment worth \$10,000 at 9 percent over five years, with payments at the end of the year and no additional payments made beyond those that are regularly scheduled.

**Database example****Formula**

FV('Rate', 'Periods', 'ExtraPmt')

**Result**

Calculates the investment's future value based on the percentage rate in the 'Rate' field, the number of payments to be made over the life of the investment in the 'Periods' field, and any extra payment made in the 'ExtraPmt' field—it leaves out the optional present value and payment timing options. It fills the field with the result.

## HLOOKUP (Horizontal Lookup)

Looks for a specific value in the top row of a cell range—either an exact match or the closest match. If a match is made, then HLOOKUP looks down the column where the match is made and then returns the value of the cell a specified number of rows down. (You indicate the number of rows using a parameter in the formula itself, as indicated below.) HLOOKUP returns an error if there is no match, and it can't be used in databases.

This formula takes a lot of parameters to do its job. Here's what each parameter does:

### Parameter

Lookup value  
Compare range

### What it does

The value you're searching for  
A range of rows used in the search. If there's only one row specified, then HLOOKUP searches through that row; if there two or more rows, then HLOOKUP searches the first row for the lookup value and returns the corresponding value from the second and following rows. *Compare range* values should be in ascending order or descending order to work properly.  
The numeric expression that tells the function how many rows to look down during the search.  
An optional flag that specifies how HLOOKUP goes about its business

Offset

Method

You can specify how the HLOOKUP function works, too:

### Value

0  
1  
-1

### What it does

Returns the lookup value's exact match  
Used when first row's values *ascend* from left to right; returns the largest value less than or equal to the lookup value.  
Used when first row's values *descend* from left to right; returns the smallest value greater than or equal to the lookup value.

## Format

HLOOKUP(*lookup value, compare range, offset, method*)

**Figure C-2:**

Refer to this spreadsheet as you consult the formula examples below.

	A	B	C	D	E	F
1						
2	Rocks!	Pretty cool	All right	Kinda lousy	Horrrific	
3	Best	Better	Good	Bad	Worst	
4						

## Spreadsheet example

Given the sample spreadsheet in Figure C-2, here's how HLOOKUP works.

### Formula

=HLOOKUP(5, A1..E3, 1, 0)  
=HLOOKUP(3,2, A1..E3, 2, -1)  
=HLOOKUP (2, A1..E3, 2, 1)

### Result

Rocks!  
Better  
#N/A (Returns an error because the Method flag is set for rows that ascend from left to right, but our sample cells *descend* from left to right.)

## HOUR (Hour)

Changes the time portion of a date serial number to the hour in military time (0-23).

### Format

HOUR(*date serial number*)

## Spreadsheet example

### Formula

=HOUR(11760.232)  
=HOUR(0.38082)  
=HOUR(25930.9843)

### Result

5  
9  
23

## Database example

### Formula

HOUR('DateSerial')

### Result

Calculates the hour of the day from the number in the 'DateSerial' field and fills the calculation field with that number.

## IF (If)

If you're looking for power in functions, then you've found it here. The IF function evaluates the logical item (the first statement in the parentheses) as true or false, and then returns the second value (if true) or third value (if false). This function can be used to give your spreadsheets real decision-making abilities—see the ERROR function for an example.

### Format

IF(*statement, true value, false value*)

## Spreadsheet example

### Formula

=IF(E17, "It's true!", "You lie!")

### Result

Examines cell E17. If the statement in E17 is true (such as  $1+2=3$ ), then the function returns the text "It's true!"; if E17 is false (such as  $3+3=11$ ), then the function returns the text "You lie!".

=IF(D3=30, ALERT("Matches 30!"), D3) Returns the text "Matches 30!" if the cell D3 equals 30; if D3 doesn't equal 30, returns what is in D3.

## Database example

### Formula

IF("TrueOrFalse", 'TrueValue', 'FalseValue')

### Result

Examines the value in the field TrueOrFalse, and fills the calculation field with the contents of the 'TrueValue' field if it resolves to TRUE; if it resolves to FALSE, then the calculation field is filled with the value in the 'FalseValue' field.

## INDEX (Index)

Returns the value of a single cell from a range of cells. The range value specifies what group of cells is to be searched by the function, the row value specifies what row to look at, and the column specifies what column to look at—together, they specify a single cell. It's not available in databases.

### Format

INDEX(*range*, *row*, *column*)

## Spreadsheet example

See Figure C-3.

**Figure C-3:**  
The examples below refer to this example spreadsheet snippet.

	A	B	C	D	E
1	1	2	3	4	
2	2	4	6	8	
3	3	6	9	12	
4	4	8	12	16	
5	5	10	15	20	
6					

### Formula

=INDEX(A1..D5, 3, 2)

=INDEX(B1..D4, 3, 2)

### Result

6

9

## INT (Int)

Returns the largest integer less than or equal to the argument.

### Format

INT(*number*)

## Spreadsheet example

### Formula

=INT(3.1415)  
 =INT(-3.1415)  
 =INT(18.9999)

### Result

3  
 -4  
 18

## Database example

### Formula

INT('Dollars')

### Result

Looks at the number in the field 'Dollars' and fills the calculation field with the largest integer less than or equal to the one in the field 'Dollars'.

## IRR (Internal Rate of Return)

Returns an internal rate of return based on an investment and series of cash flows entered in a particular range of cells (which you specify in the first argument). The first cell should include the initial investment, and the subsequent cells should include cash flow amounts received or paid.

The second argument (which is optional) represents a guess, an estimated internal rate of return, which defaults to 10 percent unless you specify otherwise. The *guess* means “number of iterations”—how many times this function will evaluate the calculation in an attempt to converge to a single value. (If it can't arrive at a value in 20 iterations, IRR returns a “#NUM!” error.)

(This function can't be used in databases.)

### Format

IRR(*range*, *guess*)

## Spreadsheet example

Refer to Figure C-4 for the following examples. (The cell containing the IRR function uses the Percent cell format.)

	A	B	C	D	E	F
1	-10000	200	1000	5000	5000	
2						

**Figure C-4:**

*This sample spreadsheet shows the arguments for one sample rate-of-return calculation*

### Formula

=IRR(A1..E1)  
 =IRR(A1..D1)

### Result

3.48%  
 15.73%

## ISBLANK (Is Blank)

Returns TRUE if the value in parentheses (cell or cells, or a text or numeric expression) is empty, or FALSE if the value is either a number or text.

### Format

ISBLANK(*value*)

### Spreadsheet example

#### Formula

=ISBLANK(A3)

#### Result

TRUE if cell A3 is empty; FALSE if cell A3 contains a value).

### Database example

#### Formula

ISBLANK('WatchedField')

#### Result

Looks at the field 'WatchedField.' If it's blank, fills the calculation field with a 1 (meaning TRUE); otherwise, fills the calculation field with a 0 (meaning FALSE).

## ISERROR (Is Error)

Checks if the value (of a cell or cells, or a text or numeric expression) is an error. You specify an error *type* as the second item in parentheses. Returns TRUE if there's an error, FALSE if not.

By specifying the kind of error you're looking for in the second item in parentheses, you can filter for the error type you're interested in and ignore the others. Here's a list of the potential error types you can filter for:

#### Error

#ARG!

#DATE!

#DIV/0!

#ERROR!

#N/A!

#NUM!

#REF!

#TIME!

#USER!

#VALUE!

#### What it is

Wrong argument type/value or wrong number of arguments

Invalid date

Math error—division by zero

General error

Not available

Invalid number or numeric over/underflow

Wrong cell is referenced

Invalid time

An error you define

Invalid value

### Format

ISERROR(*value, error type*)

## Spreadsheet example

**Formula**

```
=ISERROR(D18)
```

```
=ISERROR(B3, #VALUE!)
```

**Result**

TRUE if cell D18 contains an error; FALSE if there's no error in D18.

TRUE if cell B3 has a #VALUE! error; FALSE if there's any other error (or no error at all) in cell B3.

## Database example

**Formula**

```
ISERROR('ErrorProneField')
```

**Result**

Examines the content of the field 'ErrorProneField' and fills the calculation field with a 1 (indicating TRUE) if there's an error; it fills the calculation field with a 0 if there's no error.

## ISLOGICAL (Is Logical)

Checks if the value in the parentheses (which can be a cell or cells, an expression, number, text, or function) is a Boolean (true-or-false) expression. If so, it returns TRUE; if not, it returns FALSE.

**Format**

ISLOGICAL(*value*)

## Spreadsheet example

**Formula**

```
=ISLOGICAL(R22)
```

**Result**

Returns TRUE if the value in cell R22 is a logical value; returns FALSE if it's not.

## Database example

**Formula**

```
ISLOGICAL('Age'>'1.Q.')
```

**Result**

Fills the calculation field with 1 (which indicates TRUE) regardless of whether or not the 'Age' field is greater than the '1.Q.' field, because 'Age'>'1.Q.' is a logical expression.

## ISNA (Is Not Available)

Checks if the value in the parentheses (which can be a cell or cells, number, text, or function) has a #N/A! (not available) error. If so, it returns TRUE; if not, it returns FALSE.

ISNA can't be used in databases.

**Format**

ISNA(*value*)

## Spreadsheet example

**Formula**

```
=ISNA(C3)
```

**Result**

Returns TRUE if the value of cell C3 contains a #N/A! error; returns FALSE if it contains any other error (or no error at all).

## ISNUMBER (Is Number)

Checks if the value in parentheses (a cell or cells, number, text, or function that evaluates as a number) is a number. If so, it returns TRUE; if not, it returns FALSE.

**Format**

ISNUMBER(*value*)

## Spreadsheet example

**Formula**

```
=ISNUMBER(A7)
```

**Result**

Returns TRUE if cell A7 contains a number; returns FALSE if cell A7 contains anything else (or nothing).

## Database example

**Formula**

```
ISNUMBER('AccountBalance')
```

**Result**

Fills the calculation field with a 1 (for TRUE) if the field 'AccountBalance' contains a number (and is formatted as a number field); otherwise, it fills the calculation field with a 0 (for FALSE), which will also happen if the 'AccountBalance' field is blank.

## ISTEXT (Is Text)

Checks if the value in parentheses (a cell or cells, number, text, or function that evaluates as a number) is text. If so, it returns TRUE; if not, it returns FALSE.

**Format**

ISTEXT(*value*)

## Spreadsheet example

**Formula**

```
=ISTEXT(B2)
```

**Result**

Returns TRUE if cell B2 contains text; Returns FALSE if it contains anything else (or nothing at all).

## Database example

### Formula

ISTEXT('TheWord')

### Result

Fills the calculation field with a 1 (for TRUE) if the field 'TheWord' contains text (and is formatted as a text field); otherwise, it fills the calculation field with a 0 (for FALSE), which will also happen if the 'TheWord' field is blank.

## LEFT (Left)

Starts at the left of the text in the first item in parentheses and returns the number of characters specified by the second item. For example =LEFT("Pavilion", 3) returns *Pav*.

### Format

LEFT(*text*, *number of characters*)

## Spreadsheet example

### Formula

=LEFT("Pavilion", 3)

=LEFT("Pavilion", 0)

=LEFT(A3, B3)

### Result

Pav

#ERROR!—LEFT must return at least one character.

Returns, from the left, the number of characters in cell A3 specified in cell B3.

## Database example

### Formula

LEFT('LastName', 'CharacterNumber')

### Result

Examines the field 'LastName' and fills the calculation field with the number of characters specified in the 'CharacterNumber' field, starting at the beginning of the text and working toward the right.

## LEN (Length)

Returns the number of characters of the text in parentheses. Good for figuring out how many characters are in a given cell.

### Format

LEN(*value*)

## Spreadsheet example

### Formula

=LEN("Capitalist Tool")

### Result

15

=LEN(A23)

Counts the number of characters in A23 and returns that number. If the value in A23 is a number, returns the number of *digits*: a two-digit number will return 2.

### Database example

#### Formula

LEN('LongestWord')

#### Result

Counts the number of characters in the field 'LongestWord' and fills the calculation field with that number.

## LN (Natural Log)

Returns the natural logarithm (log base e, see above under EXP) of the number in parentheses.

### Format

LN(*number*)

### Spreadsheet example

#### Formula

=LN(3)

=LN(1)

#### Result

1.0986122887

0

### Database example

#### Formula

LN('Number')

#### Result

Fills the calculation field with the natural log of the number in the 'Number' field.

## LOG (Log)

Returns the logarithm of the first number in parentheses to the base specified by the second number in parentheses. The second number is optional; if you leave it out, the function assumes a base of 10.

### Format

LOG(*number, base*)

### Spreadsheet example

#### Formula

=LOG(5, 3)

#### Result

1.4649735207

## Database example

### Formula

LOG('OriginalNumber', 'BaseNumber')

### Result

Fills the calculation field with the log of the number in the 'OriginalNumber' field to the number in the 'BaseNumber' field.

## LOG10 (Log to Base 10)

Returns the logarithm of the number in parentheses to base 10.

### Format

LOG10(*number*)

## Spreadsheet example

### Formula

=LOG10(13)

### Result

1.1139433523

## Database example

### Formula

LOG10('OriginalNumber')

### Result

Fills the calculation field with the log to base 10 of the number in the 'Original Number' field.

## LOOKUP (Lookup)

Looks for a value within a range of cells and, if it finds a match, returns a value from a corresponding cell in a second range of cells. LOOKUP returns an error if there is no match, and it can't be used in databases.

This formula takes a lot of parameters to do its job:

### Parameter

Lookup value

Compare range

Result range

Method

### What it does

The value you're searching for

A range of cells used in the search. Compare range values should be in strict ascending order or descending order to work properly.

A range of cells from which Lookup draws its results

An optional flag that specifies how Lookup goes about its business

You can use any of these lookup schemes; specify the corresponding number as the final argument in the parentheses:

### Value

0

1

### What it does

Returns the lookup value's exact match

Used when first row's values *ascend* from left to right; returns the largest value less than or equal to the lookup value.

Used when first row's values *descend* from left to right; returns the smallest value greater than or equal to the lookup value.

## Format

LOOKUP(*lookup value, compare range, result range, method*)

## Spreadsheet example

Refer to Figure C-5.

### Figure C-5:

This spreadsheet shows how you might use the LOOKUP function (see the examples below).

	A	B	C	D
1	1	2	3	
2	4	5	6	
3	7	8	9	
4	10	11	12	
5	Aoede	Calliope	Clio	
6	Erato	Euterpe	Melete	
7	Melpomene	Mneme	Polymnia	
8	Terpsichore	Thalia	Urania	
9				

### Formula

=LOOKUP(7.3, A1..C4, A5..C8, 1)

=LOOKUP(3, A1..C4, A5..C8, 0)

=LOOKUP(10, A1..C4, A5..C8, -1)

### Result

Melpomene

Clio

#N/A (Returns an error because the Method flag is set for rows that *descend* from left to right, but our sample cells *ascend* from left to right.)

## LOWER (Lowercase)

Returns the lowercase version of the text in parentheses.

### Format

LOWER(*text*)

## Spreadsheet example

### Formula

=LOWER("FuNkY CaPs")

=LOWER(A12)

=LOWER(33)

### Result

funky caps

Returns the lowercase version of the text in cell A12.

33

## Database example

### Formula

LOWER('AllCaps')

### Result

Converts the text in the field 'AllCaps' to lowercase letters and fills the calculation field with it.

## MACRO (Macro)

Plays previously created macros, runs AppleScripts (or other Open Scripting Architecture-compliant scripts), or executes a button from the Button bar, as though you'd clicked it.

MACRO is a very powerful function that lets AppleWorks do just about anything based on the value in a given cell. MACRO returns a value indicating if the script ran successfully, or if it encountered some problem.

Here's what gets returned when you use this function to trigger macros and buttons:

### What gets returned

0

#N/A!

### What it means

The MACRO function ran the appropriate button or macro.

The MACRO function can't find the appropriate button or macro.

Here's what gets returned when you use this function to trigger AppleScript scripts:

### What gets returned

The script's results

#N/A!

#SCRIPT!

#USER!

### What it means

Returned when everything ran as expected.

The MACRO function can't find the appropriate script.

The MACRO function hit an error while running the script.

The script itself encountered an error.

The MACRO function lets you specify the type of item to be run. Here's a table of what those types are and how to specify them:

### Type

0

1

2

### What it means

Plays a macro.

Executes a button bar button.

Runs an AppleScript script.

## Format

MACRO has two formats: one for buttons and macros, and one for scripts.

For buttons and macros: `MACRO(name, type)`. (Type 0 means a macro; type 1 is a button.)

For scripts: `MACRO(name, type, subroutine, ...)`

## Spreadsheet example

### Formula

`=MACRO("Save Document", 1)`

### Result

Saves the document by "clicking" the Save Document button. To see the names of the Button bar buttons, open the Button bar preferences window.

=IF(B3=90, MACRO("Top"),  
MACRO("Not"))

Checks the cell B3; if it equals 90, then AppleWorks runs the Top macro—otherwise, it runs the Not macro.

## Database example

### Formula

MACRO('Name', 'MacroType',  
'Subroutine')

### Result

Runs the macro, button, or script in the 'Name' field with the type specified in the 'MacroType' field, and, if the script to be run is an AppleScript, executes the optional 'Subroutine' named in the 'Subroutine' field.

## MATCH (Match)

Searches a specified range of cells for a match to a given value. MATCH searches from left to right and top to bottom, scanning each row of cells until the entire range is searched. MATCH can search for an exact value or the nearest match. This formula requires several parameters:

### Parameter

Lookup value  
Compare range

Method

### What it does

The value you're searching for  
A range of cells used in the search. *Compare range* values should be in ascending order or descending order to work properly.  
An optional flag that specifies how Lookup goes about its business.

The Method parameter can be 0, 1, or -1, as follows:

### Value

0

1

-1

### What it does

Returns the lookup value's exact match.  
Used when first row's values ascend from left to right and returns the largest value less than or equal to the lookup value.  
Used when first row's values descend from left to right and returns the smallest value greater than or equal to the lookup value.

## Format

MATCH(*lookup value, compare range, method*)

**Figure C-6:**

*This spreadsheet illustrates the MATCH function, in conjunction with the following examples.*

	A	B	C	D	E
1	1	2	3	4	
2	10	20	30	40	
3	100	200	300	400	
4	1000	2000	3	4000	
5					

## Spreadsheet example

Refer to Figure C-6 for these examples.

**Formula**

=MATCH(20, A1..D4, 1)

=MATCH(3, A1..D4, 1)

**Result**

6 (MATCH finds the value 20 in the sixth cell. It counts the first row, and when it doesn't find "20" there, it drops down to the second row, where it finds a match in the sixth cell.)

3 (Even though there's a second match in the fourth row, MATCH stops at the first one it finds, which emphasizes why the values should be in strict ascending or descending order.)

## MAX (Maximum)

Returns the largest number in the list of numbers between parentheses. Can contain numbers or cell references that contain numbers.

**Format**

MAX(*number1, number2, ...*)

## Spreadsheet example

**Formula**

=MAX(4, -3, 10)

=MAX(A1..C13)

**Result**

10

Finds the largest number in the cell range A1 through C13 and returns the result.

## Database example

**Formula**

MAX('Score1', 'Score2', 'Score3')

**Result**

Looks at the values in the fields and returns the largest number it finds, filling the calculation field with that number.

## MERGEFIELD (Merge Field)

Brings a spreadsheet document's data into a database document, which is great for bringing those spreadsheet documents into a searchable, more customizable database format.

The open Mail Merge window lets you select the database to merge. Enter the MERGEFIELD formula by selecting a cell and then selecting a field from a list. (This function works in spreadsheets only.)

**Format**

MERGEFIELD(*value*)

## Spreadsheet example

### Formula

=MERGEFIELD('Address')

=MERGEFIELD(A10)

=IF((D5="Home"),  
MERGEFIELD("Home phone"),  
MERGEFIELD("Work phone"))

### Result

Merges the data from the 'Address' field in the database selected in the Mail Merge window.

Takes the field named in cell A10 and merges the data from the database selected in the Mail Merge window.

Looks at the field D5, and if it contains the text "Home ", merges the data in the 'Home phone' field in the selected database; otherwise, it merges the data from the 'Work phone' field.

## MID (Middle)

Returns the characters in the text in parentheses, starting with the character in whatever position you specify with the second argument and counting characters to the right for as many characters as you specify in the *third* argument.

### Format

MID(*text*, *start position*, *number of characters*)

## Spreadsheet example

### Formula

=MID("Timothy", 3, 4)

=MID("Best", 1, 5)

=MID(A3, 2, 2)

### Result

moth

#ERROR! because the word "Best" is only four characters long.

If the text in cell A3 is the word FALSE (from a logical function), MID will return "AL".

## Database example

### Formula

MID('SearchText', 'Start', 'Length')

### Result

Examines the search text and returns the characters from it starting at the position defined by the number in the 'Start' field for the length of the number in the 'Length' field.

## MIN (Minimum)

Returns the smallest number in the list of numbers between parentheses, which may include numbers or cell references that contain numbers.

### Format

MIN(*value1*, *value2*...)

**Spreadsheet example****Formula**

=MIN(3, -6, 0)

=MIN(A1..B3)

**Result**

-6

The smallest number in the cell range from A1 through B3.

**Database example****Formula**

MIN('Par1', 'Par2', 'Par3', 'Par4')

**Result**

Fills the calculation field with the smallest number in the fields 'Par1', 'Par2', 'Par3', and 'Par4'—for your finest time on the greens, for example.

**MINUTE (Minute)**

Returns the minutes from a date serial number, discarding the hours.

**Format**MINUTE(*date serial number*)**Spreadsheet example****Formula**

=MINUTE(19233.223)

=MINUTE(0.6793)

**Result**

21

18

**Database example****Formula**

MINUTE('MyBirthdate')

**Result**

Calculates the minute of the date from the date serial number in 'MyBirthdate' and fills the calculation field with it.

**MIRR (Modified Internal Rate of Return)**

Returns the modified internal rate of return based on the safe and risk investment rates and a series of numbers representing cash flows, entered in a range of cells.

**Format**MIRR(*safe, risk, value1, value2...*)**Spreadsheet example****Formula**

=MIRR(0.12, 0.15, -100, 200, -50)

**Result**

28.24%

## Database example

### Formula

MIRR('Safe', 'Risk', 'Flow1', 'Flow2', 'Flow3') Figures the modified rate of return from the rates and cash flow numbers in the 'Safe', 'Risk', 'Flow1', 'Flow2', and 'Flow3' fields.

### Result

## MOD (Modulo)

Returns the remainder when the first number in parentheses is divided by the second. The number returned has the same sign as the number being divided.

### Format

MODULO(*number, divisor number*)

## Spreadsheet example

### Formula

=MOD(3, 2)

=MOD(2, 3)

=MOD(-13, 5)

### Result

1

2

-3

## Database example

### Formula

MOD('FractionTop', 'FractionBottom')

### Result

Calculates the remainder if the number in the 'FractionTop' field is divided by the number in the 'FractionBottom' field and fills the calculation field with that number.

## MONTH (Month)

Returns the number of the month (1–12) calculated from a date serial number.

### Format

MONTH(*date serial number*)

## Spreadsheet example

### Formula

=MONTH(27389.34)

=MONTH(30000)

### Result

12

2

## Database example

### Formula

Month('Anniversary')

### Result

Calculates the number of the month from the date serial number in the field 'Anniversary' and fills the calculation field with that number.

## MONTHNAME (Month Name)

Returns the name of the month from the number in parentheses (which must be an integer, 1–12). Handy for decoding the result of a MONTH operation.

### Format

MONTHNAME(*Number*)

### Spreadsheet example

**Formula**

=MONTHNAME(6)  
=MONTHNAME(MONTH(30000))

**Result**

June  
February. Takes the number of the month returned by the MONTH function and converts it into a month name.

### Database example

**Formula**

MONTHNAME("Easter")

**Result**

Calculates the name of the month from the number in the 'Easter' field and fills the calculation field with that name.

## NA (Not Available)

Returns the error message "#N/A!" Useful when nested in other functions, such as IF. It can't be used in databases. (Don't put anything in the parentheses.)

### Format

NA()

### Spreadsheet example

**Formula**

=IF(ISNUMBER(B3), "Number!", NA())

**Result**

If the value in cell B3 is anything other than a number (even if it's empty), then this function returns the #N/A! error; otherwise, it returns the word "Number!"

## NOT (Not)

Evaluates the number or equation in the parentheses and returns the opposite value. It works in two ways:

- If the argument in parentheses is a number (or a number expression, such as "2+2"), this function returns TRUE if the number comes out to 0, or FALSE otherwise.
- If the argument in parentheses is an equation, you get TRUE if any argument is false and FALSE if all arguments are true.

**Format**

NOT(*statement*)

**Spreadsheet example****Formula**

=NOT(6+3)

=NOT(AND(6+3=3))

**Result**

FALSE (because 6+3 evaluates as TRUE).

TRUE (because AND(6+3) evaluates as FALSE).

**Database example****Formula**

NOT('Weight')

**Result**

Returns a 1 (which stands for TRUE) if any value other than 0 fills the 'Weight' field.

**NOW (Now)**

Returns the current date and time (or serial number of the current date and time, depending on the cell's formatting) from your system clock when the spreadsheet is recalculated.

**Format**

NOW()

**Spreadsheet example****Formula**

=NOW()

**Result**

Returns the date serial number 35134.61998 if the computer's clock reads the date as March 11, 2000 and the time as 2:52:46 p.m.

**Database example****Formula**

NOW()

**Result**

Fills the field with the current date and time serial number.

**NPER (Number of Periods)**

Returns the number of payment periods based on a given interest rate, payment amount made per period, the investment's present value, the investment's future value (optional—defaults to 0), and the type of payment scheme (also optional). A payment scheme number of 0 is used for payments at the end of the period, and a 1 is used for payments at the beginning of the period.

**Format**

NPER(*rate, pmt, pv, fv, type*)

## Spreadsheet example

**Formula**

=NPER(10%/12, -250, 10000)

**Result**

48.86 (rounded to two digits)—48 monthly payments at 10 percent annual interest, and payments of \$250 every month with a present value of \$10,000.

## Database example

**Formula**

NPER('Percent', 'Payment', 'Value')

**Result**

Fills the calculation field with the number of payments based on the percentage rate in the 'Percent' field, the payment amount in the 'Payment' field, and the present value in the 'Value' field.

## NPV (Net Present Value)

Returns the net present value of an investment based on the interest rate and the series of payments in parentheses.

**Format**

NPV(*interest rate, payment1, payment2...*)

## Spreadsheet example

**Formula**

=NPV(6%, -1000, 100, 500, 1000, 1000)

**Result**

1104.76 (rounded to two decimal places)

## Database example

**Formula**

NPV('Rate', 'Pay1', 'Pay2', 'Pay3', 'Pay4')

**Result**

Calculates an investment's net present value based on the percentage rate in the field 'Rate' and the payments in the fields 'Pay1', 'Pay2', 'Pay3', 'Pay4', and 'Pay5'.

## NUMTOBASE (Number to Base)

Changes a base 10 number (the first number in parentheses) into a number of another base (the second number in parentheses). The optional third number in parentheses specifies the minimum number of digits returned. (This function is the reverse of Base to Number, and it's useful to Web masters, programmers, and other numbery sorts.)

**Format**

=NUMTOBASE(*number, base, minimum digits*)

## Spreadsheet example

### Formula

=NUMTOBASE(251, 16)

=NUMTOBASE(6, 2)

### Result

FB (hexadecimal)

110 (binary)

## Database example

### Formula

NUMTOBASE('Number', 'Base')

### Result

Converts the number in the 'Number' field to the base in the 'Base' field and fills the calculation field with that number.

## NUMTOTEXT (Number to Text)

Converts the number (or cell reference, or field) in parentheses to its textual version. Used to convert a number to be used in a text field, or to prevent a number from being included when you create a chart from your data.

### Format

NUMTOTEXT(*number*)

## Spreadsheet example

### Formula

=NUMTOTEXT(23)

### Result

23—It may *look* the same, but the item returned is actually text and can be used in text-based functions.

## Database example

### Formula

NUMTOTEXT('SourceNumber')

### Result

Converts the number in the field 'SourceNumber' to text and fills the calculation field with that text. It makes a number usable in a text-only function.

## OR (Or)

Evaluates the number or equation in the parentheses. It works in two ways:

- If the argument in parentheses is a number (or a number expression, such as "2+2"), this function returns FALSE if the number comes out to 0, or TRUE otherwise.
- If the argument in parentheses is an equation or set of them, you get TRUE if *any* one of them is true and FALSE if they're all false.

### Format

OR(*logical1, logical2...*)

## Spreadsheet example

**Formula**

```
=OR(3+5=8, 2+2=4)
=OR(3+5=8, 2+2=5)
=OR(3+4=8, 2+2=5)
=OR(13-13, 8-8)
```

**Result**

```
TRUE
TRUE
FALSE
FALSE
```

## Database example

**Formula**

```
OR('Account1', 'Account2')
```

**Result**

Returns 0 (for FALSE) if both 'Account1' and 'Account2' are blank—if either contains any value, then it returns 1 (for TRUE).

## PI (Pi)

Returns the value of Pi.

**Format**

```
PI()
```

## Spreadsheet example

**Formula**

```
=PI()
```

**Result**

```
3.1415926536
```

## Database example

**Formula**

```
PI()
```

**Result**

Fills the calculation field with the value of Pi.

## PMT (Payment)

Returns the payment amount per period calculated from the interest rate, number of payment periods, the present value, the investment's future value (optional—defaults to 0), and the type of payment scheme (also optional) given in parentheses. The payment scheme can be either 0 for payments at the end of the period or 1 for payments at the beginning of the period.

**Format**

```
=PMT(rate, nper, pv, fv, type)
```

## Spreadsheet example

**Formula**

```
=PMT(5%/12, 60, 10000, 12000, 0)
```

**Result**

365.17 (when rounded to two decimal places). Payments will be \$365.17 per monthly payment over 60 months if the present value is \$10,000, the future value will be \$12,000, and payments are made at the end of the month.

**Database example****Formula**

PMT('Rate', 'Payments', 'Value')

**Result**

Calculates payment amounts using the percentage rate in the 'Rate' field, the number of payments in the 'Payments' field, and the investment's value in the 'Value' field.

**PRODUCT (Product)**

Returns the product of the numbers (or cells) in parentheses.

**Format**

PRODUCT(*number1*, *number2*, ...)

**Spreadsheet example****Formula**

=PRODUCT(3, 6, 10, 33)

=PRODUCT(B11..B15)

**Result**

5940

Multiplies the numbers in the cells B11 through B15 and returns the result.

**Database example****Formula**

PRODUCT('Rabbit1', 'Rabbit2')

**Result**

Multiplies the number in 'Rabbit1' and 'Rabbit2' together, and fills the calculation field with the result.

**PROPER (Proper)**

Returns the text in parentheses after capitalizing the first letter in each word.

**Format**

PROPER(*text*)

**Spreadsheet example****Formula**

=PROPER("itty bitty text")

**Result**

Itty Bitty Text

**Database example****Formula**

PROPER('Title')

**Result**

Capitalizes the first letter of each word in the field 'Title' and fills the calculation string with the result.

## PV (Present Value)

Returns an investment's present value, calculated from the interest rate for each payment period, number of payment periods, the amount of payment to be made per period, the investment's future value (optional—defaults to 0), and the type of payment scheme (also optional) given in parentheses. Use a 0 for payments at the end of the period, and 1 for payments at the beginning of the period.

### Format

PV(*rate*, *nper*, *pmt*, *fv*, *type*)

### Spreadsheet example

#### Formula

=PV(8%, 10, 300, 15000)

#### Result

-\$8960.93 (rounded to two decimal places). To accumulate an investment of \$15,000 in 10 years with payments of \$300 each year at an 8% annual rate, you would have to invest \$8960.93 at the start.

### Database example

#### Formula

PV('Rate', 'Periods', 'Payment')

#### Result

Calculates an investment's present value based on the interest rate in the 'Rate' field, the number of payment periods in the 'Periods' field, and the payment amount per period in the 'Payment' field.

## RADIANS (Radians)

Converts an angle in degrees (the number in parentheses) into an angle in radians.

### Format

RADIANS(*number*)

### Spreadsheet example

#### Formula

=RADIANS(150)

#### Result

2.617993878

### Database example

#### Formula

RADIANS('DegreeAngle')

#### Result

Takes the number (in degrees) in the 'DegreeAngle' field and converts it to radians.

## RAND (Random)

Returns a random number between 1 and the number in parentheses. If the parentheses are empty, the number generated is between zero and one. Use this function to be the “seed” from which you calculate lottery results, gambling simulations, and so on.

### Format

RAND(*number*)

### Spreadsheet example

#### Formula

=RAND(20)

=RAND()

#### Result

A random number between 1 and 20.

A random number between 0 and 1.

### Database example

#### Formula

RAND('UpperLimit')

#### Result

Generates a random number from 1 to the value in the field 'UpperLimit' and fills the calculation field with that result.

## RATE (Rate)

Returns the interest rate per payment period needed to make a given investment grow into a future value over a specified number of payments. (When you make monthly payments, the interest rate that's returned is a monthly interest rate, not an annual interest rate, which is what your investment plan usually reports.)

### Format

RATE(*fv*, *pv*, *term*)

### Spreadsheet example

#### Formula

=RATE(15000, 10000, 10)

#### Result

4.14% (If the cell is formatted in a percentage format).

### Database example

#### Formula

RATE('Goal', 'Start', 'Periods')

#### Result

Computes the percentage rate needed to make the number in the 'Start' field grow to the number in the 'Goal' field, given the number of periods in the 'Periods' field, and fills the calculation field with that number.

## REPLACE (Replace)

Replaces the text in old text with the text in new text, starting at the position specified for the specified number of characters.

### Format

REPLACE(*old text, starting number, number of characters, new text*)

### Spreadsheet example

#### Formula

=REPLACE("Bush", 1, 1, "R")  
=REPLACE("absolve", 3, 5, "stain")

#### Result

R  
abstain

### Database example

#### Formula

REPLACE('Original', 'Position',  
'Number', 'Munge')

#### Result

Replaces the portion of text in the field 'Original' with characters from the field 'Munge' according to the numbers in the 'Position' field and the 'Number' field.

## REPT (Repeat)

Repeats the text in parentheses the number of times specified by the second item.

### Format

REPT(*text, number of times*)

### Spreadsheet example

#### Formula

=REPT("la ", 5)

#### Result

la la la la la. (Notice the space after the "a".)

### Database example

#### Formula

REPT('SenselessPhrase', 'IrritationFactor')

#### Result

Repeats the text in the field 'SenselessPhrase' the number of times specified in the field 'IrritationFactor'.

## RIGHT (Right)

Starts at the right of the text in the first item in parentheses, and returns the number of characters specified by the second item, moving from right to left.

### Format

RIGHT(*text, number of characters*)

## Spreadsheet example

### Formula

=RIGHT("Reserve", 3)  
 =RIGHT("Reserve", 6)  
 =RIGHT("Reserve", 0)

### Result

ve  
 ervice  
 #ERROR! (RIGHT must return at least one character.)

## Database example

### Formula

RIGHT('LastName', 'CharacterNumber')

### Result

Examines the field 'LastName' and fills the calculation field with the number of characters specified in the 'CharacterNumber' field, starting from the end and working toward the beginning.

## ROUND (Round)

Rounds the first number in parentheses to the number of decimal digits specified by the second argument. For those with a real noggin for numbers, the Round function uses the "round half to higher" method rather than the "round to nearest or even" method. Round to nearest even" is a more accurate rounding method, and it's the one generally preferred by those who know about such things. It involves some fairly complicated binary math, so unless you're doing some serious number crunching, this difference probably won't affect you much. If you're interested in this, there's a discussion at <http://www-ee.eng.hawaii.edu/Courses/EE361.S95/Lectures/Lec40/lec40.3.html>.

### Format

ROUND(*number*, *number of digits*)

## Spreadsheet example

### Formula

=ROUND(-6.375, 1)  
 =ROUND(42.1873, 2)

### Result

-6.4  
 42.19

## Database example

### Formula

ROUND('FattyDecimalNumber', 'Limit')

### Result

Rounds the number in the field 'FattyDecimalNumber' to the number of digits specified by 'Limit'.

## ROW (Row)

Returns the row number of the cell in parentheses. If there's no cell specified, returns the row number of the cell that it's in. (Not available in databases.)

**Format**ROW(*cell*)**Spreadsheet example****Formula**

=ROW(D8)

**Result**

8

**SECOND (Second)**

Returns the number of seconds calculated from a date serial number, discarding the hours and minutes.

**Format**=SECOND(*serial number*)**Spreadsheet example****Formula**

=SECOND(0.61998)

=SECOND(32839.23994)

**Result**

46

31

**Database example****Formula**

SECOND('MyBirthdate')

**Result**

Calculates the seconds of the date from the date serial number in 'MyBirthdate' and fills the calculation field with it.

**SIGN (Sign)**

Returns a 1 when the number in parentheses is positive, -1 when the number is negative, and 0 when the number is zero. You can use it to tell whether the cell's contents are negative, positive, or zero.

**Format**SIGN(*number*)**Spreadsheet example****Formula**

=SIGN(-6)

=SIGN(8.23)

=SIGN(0)

**Result**

-1

1

0

**Database example****Formula**

SIGN('CheckingBalance')

**Result**

Examines the number in the field 'CheckingBalance' and returns a 1, 0, or -1 based on the sign of the number in the field.

## SIN (Sine)

Returns the sine (a number between -1 and 1) of the argument (an angle in radians).

### Format

SIN(*number*)

### Spreadsheet example

#### Formula

=SIN(3.2)

=SIN(0)

=SIN(-0.666)

#### Result

-0.05837414343

0

-0.61784574085

### Database example

#### Formula

SIN('Angle')

#### Result

Calculates the sine of the number in the field 'Angle' and returns it as an angle in radians.

## SQRT (Square Root)

Returns the square root of the number in parentheses. It doesn't allow negative numbers as an argument.

### Format

SQRT(*number*)

### Spreadsheet example

#### Formula

=SQRT(4)

=SQRT(2)

#### Result

2

1.4142135624

### Database example

#### Formula

SQRT('TriangleSide')

#### Result

Calculates the square root of the number in the field 'TriangleSide' and fills the calculation field with the result.

## STDEV (Standard Deviation)

Returns the standard deviation of the number sequence given in parentheses. (Note to mathematicians: AppleWorks uses the  $n-1$  method of computing standard deviation.)

### Format

=STDEV(*number 1, number 2, ...*)

## Spreadsheet example

**Formula**

=STDEV(83, 78, 66, 81.3)

**Result**

7.6695827787

## Database example

**Formula**

STDEV('Pop1', 'Pop2', 'Pop3', 'Pop4')

**Result**

Computes the standard deviation of the numbers in fields 'Pop1', 'Pop2', 'Pop3', and 'Pop4' and fills the calculation field with the result.

## SUM (Sum)

Returns the sum of the values in parentheses. The single most popular spreadsheet function in human history.

**Format**

SUM(*value1, value2, ...*)

## Spreadsheet example

**Formula**

=SUM(23, 7, 10, -3)  
=SUM(2, A3..B3, D7)

**Result**

37  
Adds the number 2 to the contents of cells A3 through B3 and D7 together, and returns the result.

## Database example

**Formula**

SUM('Check1', 'Check2', 'Check3')

**Result**

Adds the contents of fields 'Check1', 'Check2', and 'Check3', and fills the calculation field with the results.

## TAN (Tangent)

Returns the *tangent* of the angle in parentheses (an angle in radians). (The tangent is the quotient obtained by dividing the sine by the cosine.)

**Format**

TAN(*number*)

## Spreadsheet example

**Formula**

=TAN(-6)  
=TAN(1)

**Result**

0.29100619138  
1.5574077247

**Database example****Formula**

TAN('LittleNumber')

**Result**

Returns the tangent of the angle in 'LittleNumber' (expressed in radians) and fills the calculation field with that number.

**TEXTTODATE (Text to Date)**

Returns the date in parentheses as a date serial number (the number of days before or after January 1, 1904).

**Format**

TEXTTODATE(date text)

**Spreadsheet example****Formula**

=TEXTTODATE("Jan 4, 2000")

=TEXTTODATE("2/29/1996")

**Result**

35067

33662

**Database example****Formula**

TEXTTODATE('Today'sDate')

**Result**

Converts the date in the field 'Today'sDate' to a date serial number and fills the calculation field with that number.

**TEXTTONUM (Text to Number)**

Returns the text in parentheses as a number. Useful for converting a textual number into a number that can be used in a numeric formula or for extracting the numeric parts of a text string.

**Format**

TEXTTONUM(text)

**Spreadsheet example****Formula**

=TEXTTONUM("23")

=TEXTTONUM("Apartment 302-B")

**Result**

23. Although it looks almost identical, some formulas see a world of difference between "23" and 23.

302. Text to Number extracts just the number 302 from the text string.

## Database example

**Formula**  
TEXTTONUM("SerialNumber")

**Result**  
Converts the textual number in the field 'SerialNumber' to a number and fills the calculation field with the result.

## TEXTTOTIME (Text to Time)

Returns the time in parentheses (starting as text in quotes) as a time serial number. The serial number is a decimal fraction of the number of hours, minutes, and seconds.

### Format

TEXTTOTIME(*time text*)

## Spreadsheet example

**Formula**  
=TEXTTOTIME("1:28:53")

**Result**  
0.06172453704

## Database example

**Formula**  
TEXTTOTIME("CurrentTime")

**Result**  
Calculates a time serial number from the time entered in the field 'CurrentTime' and fills the calculation field with the result.

## TIME (Time)

Returns a time serial number from the hours, minutes, and seconds given in parentheses. This is the version to use when you have actual numbers (not text) for the time you want to work with.

### Format

TIME(*hour, minute, second*)

## Spreadsheet example

**Formula**  
=TIME(14, 52, 46)

**Result**  
0.61997685185

## Database example

**Formula**  
TIME('Hours', 'Minutes', 'Seconds')

**Result**  
Computes a time serial number from the numbers in the 'Hours', 'Minutes', and 'Seconds' fields, and it fills the calculation field with the results.

## TIMETOTEXT (Time to Text)

Returns the text generated by interpreting a time serial number and formatting it according to the second item in parentheses. The serial number is a decimal expression of the number of hours, minutes, and seconds.

The second argument can be one of these four numbers, which specify how you want the resulting time formatted:

Option	Format used
0	1:01 PM
1	1:01:01 PM
2	13:01
3	13:01:01

### Format

TIMETOTEXT(*serial number, format*)

### Spreadsheet example

Formula	Result
=TIMETOTEXT(0.61997685185, 3)	14:52:46
=TIMETOTEXT(0.30822916667, 1)	7:23:51 AM

### Database example

Formula	Result
TIMETOTEXT('TimeSerial', 'Format')	Converts the number in the 'TimeSerial' field to a text format according to the code in the 'Format' field.

## TRIM (Trim)

Removes extra spaces from the text in parentheses, compressing multiple spaces in a row into a single space.

### Format

TRIM(*text*)

### Spreadsheet example

Formula	Result
=TRIM("Lost In Spaces")	Lost In Spaces

### Database example

Formula	Result
TRIM('SpacySentence')	Removes all extraneous spaces from the field 'SpacySentence', leaving one space between words and puts the result in the calculation field.

## TRUNC (Truncate)

Lops off any digits to the right of the decimal point.

### Format

TRUNC(*number*)

### Spreadsheet example

**Formula**

=TRUNC(23.9996)

=TRUNC(-5.213)

**Result**

23

-5

### Database example

**Formula**

TRUNC('FattyDecimal')

**Result**

Removes all of the numbers to the right of the decimal point from the number in the field 'FattyDecimal.'

## TYPE (Type)

Returns the type of data specified in the argument. This information is provided as a number:

**Result**

5

6

7

8

**If the parentheses...**

Are empty.

Contain logical (true/false) statement (such as "1+1=3").

Contain a number.

Contain text.

If the value in parentheses refers to a cell or field, TYPE evaluates the contents of that cell or field in the same way.

### Format

=TYPE(*value*)

### Spreadsheet example

**Formula**

=TYPE(-6)

=TYPE(FALSE)

=TYPE("Hello, world!")

=TYPE(E1)

**Result**

3

2

4

1 (if the cell E1 is blank)

### Database example

**Formula**

TYPE('MysteryField')

**Result**

Returns the type of data in the field 'MysteryField' and fills the calculation field with the result.

## UPPER (Uppercase)

Returns the all-capitalized version of the text in parentheses.

### Format

UPPER(*text*)

### Spreadsheet example

#### Formula

=UPPER("Ye!!")

=UPPER("WaCkYtExT")

#### Result

YELL

WACKYTEXT

### Database example

#### Formula

UPPER('WeightyText')

#### Result

Fills the calculation field with the all-caps version of whatever's in the field 'WeightyText'.

## VAR (Variance)

Returns the variance of the numbers in parentheses.

### Format

VAR(*number1, number2, ...*)

### Spreadsheet example

#### Formula

=VAR(23, 28, 31, 25, 24)

#### Result

10.7

### Database example

#### Formula

VAR('Item1', 'Item2', 'Item3', 'Item4')

#### Result

Measures the variance of the numbers in the 'Item1', 'Item2', 'Item3', 'Item4' fields, and returns the results in the calculation field.

## VLOOKUP (Vertical Lookup)

This function is the flip side of the HLOOKUP function described earlier.

This time, the function looks for a particular value in the leftmost column of a cell range. If it finds a match, then VLOOKUP returns the value of the cell a specified number of columns to the right. VLOOKUP returns an error if there is no match, and it can't be used in databases.

This formula takes a lot of parameters to do its job:

#### Parameter

Lookup value

#### What it does

The value you're searching for

Compare range

A range of columns used in the search. If there's only one column specified, then VLOOKUP searches through that column; if there two or more columns, then VLOOKUP searches the first column for the lookup value and returns the corresponding value from the second and following column. *Compare range* values should be in ascending order or descending order to work properly.

Offset

This number tells the function how many columns to look across during the search.

Method

An optional flag that specifies how VLOOKUP goes about its business, as described next.

You can use the final argument, *method*, to specify how VLOOKUP determines what constitutes a match:

**Value**

0

**What it does**

Return the lookup value's exact match

1

Used when first column's values *ascend* from top to bottom; returns the largest value less than or equal to the lookup value.

-1

Used when first column's values *descend* from top to bottom; returns the smallest value greater than or equal to the lookup value.

**Format**

VLOOKUP(*lookup value, compare range, offset, method*)

**Spreadsheet example**

See Figure C-7 for the following examples.

	A	B	C	D
1	5	Rocks!	Best	
2	4	Pretty cool	Better	
3	3	All right	Good	
4	2	Kinda lousy	Bad	
5	1	Horrific	Worst	
6				

**Figure C-7:**

*This sample spreadsheet provides the data for the following VLOOKUP examples.*

**Formula**

=VLOOKUP(4, A1..C5, 1, 0)  
 =VLOOKUP(1,2, A1..C5, 2, -1)  
 =VLOOKUP(5, A1..C5, 2, 1)

**Result**

Pretty cool  
 Bad  
 #N/A! (Returns an error because the Method flag is set for columns that *ascend* as read from top to bottom, but these sample cells *descend*.)

## WEEKDAY (Weekday)

Returns the day of the week generated from the date serial number in parentheses, where 1 refers to Sunday and 7 means Saturday.

### Format

WEEKDAY(*date serial number*)

### Spreadsheet example

**Formula**

=WEEKDAY(36221)

**Result**

2 (Monday)

### Database example

**Formula**

WEEKDAY('DayNumber')

**Result**

Converts the date serial number in the field 'DayNumber' into a day of the week and fills the calculation field with the result.

## WEEKOFYEAR (Week of Year)

Returns the week of the year generated from a date serial number.

---

**Note:** The WEEKOFYEAR function may not agree with the methods used by many governments or corporations. The AppleWorks version has week number one starting with January 1st, always putting December 31st in week 53. The conventions used by others assume that the week begins on a specific day of the week, and then determines the week-of-year number based on the day of the week for January 1st.

---

### Format

WEEKOFYEAR(*serial number*)

### Spreadsheet example

**Formula**

=WEEKOFYEAR(27830)

=WEEKOFYEAR(35421)

**Result**

11

52

### Database example

**Formula**

WEEKOFYEAR('SpecialDate')

**Result**

Computes the week of the year from the date serial number in the field 'SpecialDate' and fills the calculation field with that value.

## YEAR (Year)

Returns the year generated from a date serial number.

## **Format**

YEAR(*serial number*)

## **Spreadsheet example**

### **Formula**

=YEAR(35533)

### **Result**

2001

## **Database example**

### **Formula**

YEAR('DateSerial')

### **Result**

Computes the year from the date serial number in the field 'DateSerial' and fills the calculation field with the result.