The Ultimate Guide to the Top 100 Excel Functions

MAKE YOUR DATA WORK FOR YOU

POOJA PAWAR

1. Mathematical and Trigonometric Functions

1. **SUM**
    - **Syntax:** SUM(number1, [number2], …)
    - Adds all the numbers in a range of cells.
    - **Example:** SUM(A1:A5) adds the values in cells A1 to A5.
2. **AVERAGE**
    - **Syntax:** AVERAGE(number1, [number2], …)
    - Returns the arithmetic mean of the numbers provided.
    - **Example:** AVERAGE(B1:B5) calculates the average of the values in cells B1 to B5.
3. **SUMIF**
    - **Syntax:** SUMIF(range, criteria, [sum_range])
    - Adds cells specified by a given condition or criteria.
    - **Example:** SUMIF(A1:A5, ">10") adds the values in cells A1 to A5 that are greater than 10.
4. **SUMIFS**
    - **Syntax:** SUMIFS(sum_range, criteria_range1, criteria1, [criteria_range2, criteria2], …)
    - Adds cells that meet multiple criteria.
    - **Example:** SUMIFS(B1:B5, A1:A5, ">10", C1:C5, "<20") adds values in B1 where A1 is greater than 10 and C1 is less than 20.
5. **COUNT**
    - **Syntax:** COUNT(value1, [value2], …)
    - Counts the number of cells that contain numbers.
    - **Example:** COUNT(A1:A10) counts the number of cells in the range A1 to A10 that contain numbers.

6. **COUNTA**
    - **Syntax:** COUNTA(value1, [value2], …)
    - Counts the number of cells that are not empty.
    - **Example:** COUNTA(A1:A10) counts the number of cells in the range A1 to A10 that are not empty.
7. **COUNTIF**
    - **Syntax:** COUNTIF(range, criteria)
    - Counts the number of cells that meet a single criterion.
    - **Example:** COUNTIF(A1:A10, ">5") counts the number of cells in the range A1 to A10 that are greater than 5.
8. **COUNTIFS**
    - **Syntax:** COUNTIFS(criteria_range1, criteria1, [criteria_range2, criteria2], …)
    - Counts the number of cells that meet multiple criteria.
    - **Example:** COUNTIFS(A1:A10, ">5", B1:B10, "<10") counts the number of cells where A1 is greater than 5 and B1 is less than 10.
9. **PRODUCT**
    - **Syntax:** PRODUCT(number1, [number2], …)
    - Multiplies all the numbers given as arguments.
    - **Example:** PRODUCT(A1:A5) multiplies the values in cells A1 to A5.
10. **SQRT**
    - **Syntax:** SQRT(number)
    - Returns the square root of a number.
    - **Example:** SQRT(16) returns 4.

11. **MOD**

- **Syntax:** MOD(number, divisor)
- Returns the remainder after a number is divided by a divisor.
- **Example:** MOD(10, 3) returns 1.

12. **ROUND**

- **Syntax:** ROUND(number, num_digits)
- Rounds a number to a specified number of digits.
- **Example:** ROUND(3.14159, 2) returns 3.14.

13. **ROUNDUP**

- **Syntax:** ROUNDUP(number, num_digits)
- Rounds a number up, away from zero, to a specified number of digits.
- **Example:** ROUNDUP(3.14159, 2) returns 3.15.

14. **ROUNDDOWN**

- **Syntax:** ROUNDDOWN(number, num_digits)
- Rounds a number down, toward zero, to a specified number of digits.
- **Example:** ROUNDDOWN(3.14159, 2) returns 3.14.

15. **INT**

- **Syntax:** INT(number)
- Rounds a number down to the nearest integer.
- **Example:** INT(5.9) returns 5.

16. **ABS**

- **Syntax:** ABS(number)
- Returns the absolute value of a number.
- **Example:** ABS(-5) returns 5.

17. **POWER**

- **Syntax:** POWER(number, power)
- Returns the result of a number raised to a power.
- **Example:** POWER(2, 3) returns 8.

18. **RAND**

- **Syntax:** RAND()
- Returns a random number between 0 and 1.
- **Example:** RAND() might return 0.543.

19. **RANDBETWEEN**

- **Syntax:** RANDBETWEEN(bottom, top)
- Returns a random integer between the numbers you specify.
- **Example:** RANDBETWEEN(1, 10) returns a random number between 1 and 10.

20. **PI**

- **Syntax:** PI()
- Returns the value of π (Pi).
- **Example:** PI() returns 3.14159265358979.

2. Lookup and Reference Functions

21. **VLOOKUP**

- **Syntax:** VLOOKUP(lookup_value, table_array, col_index_num, [range_lookup])
- Searches for a value in the first column of a table and returns a value in the same row from a specified column.
- **Example:** VLOOKUP("John", A2:C10, 2, FALSE) finds "John" in column A and returns the value from column B.

22. **HLOOKUP**

- **Syntax:** HLOOKUP(lookup_value, table_array, row_index_num, [range_lookup])
- Searches for a value in the first row of a table and returns a value in the same column from a specified row.
- **Example:** HLOOKUP("Q1", A1:E5, 3, FALSE) finds "Q1" in the first row and returns the value from the third row.

23. **INDEX**

- **Syntax:** INDEX(array, row_num, [column_num])
- Returns the value of an element in a table or array, selected by the row and column number.
- **Example:** INDEX(A1:C3, 2, 2) returns the value in the second row and second column of the range A1.

24. **MATCH**

- **Syntax:** MATCH(lookup_value, lookup_array, [match_type])
- Searches for a specified value in a range and returns its relative position.
- **Example:** MATCH(25, A1:A10, 0) finds the position of the value 25 in the range A1.

25. **XLOOKUP**

- **Syntax:** XLOOKUP(lookup_value, lookup_array, return_array, [if_not_found], [match_mode], [search_mode])
- Returns a value from the corresponding position in another range o rararray.
- **Example**: XLOOKUP(“John”,A2:A10,B2:B10,”Not Found).

26. **LOOKUP**

- **Syntax:** LOOKUP(lookup_value, lookup_vector, [result_vector])
- Searches for a value in a vector and returns a value from the same position in another vector.
- **Example:** LOOKUP(4, A1:A5, B1:B5) finds 4 in the range A1 and returns the corresponding value from B1.

27. **OFFSET**

- **Syntax:** OFFSET(reference, rows, cols, [height], [width])
- Returns a reference to a range that is a specified number of rows and columns from a cell or range.
- **Example:** OFFSET(A1, 2, 2) returns the reference to the cell that is two rows down and two columns to the right of A1.

28. **TRANSPOSE**

- **Syntax:** TRANSPOSE(array)
- Changes the orientation of a range of cells (rows become columns, and vice versa).
- **Example:** TRANSPOSE(A1:B2) converts rows to columns and vice versa.

29. **HYPERLINK**

- **Syntax:** HYPERLINK(link_location, [friendly_name])
- Creates a shortcut or jump to another location in the workbook, or an external file.
- **Example:** HYPERLINK("http://example.com", "Visit Example") creates a clickable link to "http://example.com".

30. **ADDRESS**

- **Syntax:** ADDRESS(row_num, column_num, [abs_num], [a1], [sheet_text])
- Creates a cell address as text, given specified row and column numbers.
- **Example:** ADDRESS(1, 1) returns "A1".

3. Text Functions

31. **CONCAT**
    - **Syntax:** CONCAT(text1, [text2], …)
    - Joins several text strings into one string.
    - **Example:** CONCAT("Hello", " ", "World") returns "Hello World".
32. **TEXT**
    - **Syntax:** TEXT(value, format_text)
    - Formats a number and converts it to text.
    - **Example:** TEXT(1234.56, "$#,##0.00") returns "$1,234.56".
33. **LEFT**
    - **Syntax:** LEFT(text, [num_chars])
    - Extracts the leftmost characters from a text string.
    - **Example:** LEFT("Excel", 2) returns "Ex".
34. **RIGHT**
    - **Syntax:** RIGHT(text, [num_chars])
    - Extracts the rightmost characters from a text string.
    - **Example:** RIGHT("Excel", 2) returns "el".
35. **MID**
    - **Syntax:** MID(text, start_num, num_chars)
    - Extracts characters from the middle of a text string.
    - **Example:** MID("Excel", 2, 3) returns "xce".

36. **LEN**

- **Syntax:** LEN(text)
- Returns the number of characters in a text string.
- **Example:** LEN("Excel") returns 5.

37. **FIND**

- **Syntax:** FIND(find_text, within_text, [start_num])
- Finds the position of a substring within another text string.
- **Example:** FIND("e", "Excel") returns 1.

38. **SEARCH**

- **Syntax:** SEARCH(find_text, within_text, [start_num])
- Similar to FIND, but not case-sensitive.
- **Example:** SEARCH("e", "Excel") returns 1.

39. **TRIM**

- **Syntax:** TRIM(text)
- Removes extra spaces from text, leaving only single spaces between words.
- **Example:** TRIM(" Excel ") returns "Excel".

40. **UPPER**

- **Syntax:** UPPER(text)
- Converts text to uppercase.
- **Example:** UPPER("Excel") returns "EXCEL".

41. **LOWER**

- **Syntax:** LOWER(text)
- Converts text to lowercase.
- **Example:** LOWER("Excel") returns "excel".

42. **PROPER**

- **Syntax:** PROPER(text)
- Capitalizes the first letter of each word in a text string.
- **Example:** PROPER("hello world") returns "Hello World".

43. **REPLACE**

- **Syntax:** REPLACE(old_text, start_num, num_chars, new_text)
- Replaces part of a text string with a different text string.
- **Example:** REPLACE("Excel", 1, 2, "Ax") returns "Axcel".

44. **SUBSTITUTE**

- **Syntax:** SUBSTITUTE(text, old_text, new_text, [instance_num])
- Substitutes new text for old text in a string.
- **Example:** SUBSTITUTE("Excel", "e", "a") returns "Excal".

4. Date and Time Functions

45. **TODAY**
    - **Syntax:** TODAY()
    - Returns the current date.
    - **Example:** TODAY() returns today's date.
46. **NOW**
    - **Syntax:** NOW()
    - Returns the current date and time.
    - **Example:** NOW() returns the current date and time.
47. **DATE**
    - **Syntax:** DATE(year, month, day)
    - Returns the serial number of a specific date.
    - **Example:** DATE(2024, 9, 17) returns the serial number representing that date.
48. **DATEVALUE**
    - **Syntax:** DATEVALUE(date_text)
    - Converts a date in text format to a serial number.
    - **Example:** DATEVALUE("1/1/2024") returns the serial number of the date.
49. **DAY**
    - **Syntax:** DAY(serial_number)
    - Returns the day of a date, as a number from 1 to 31.
    - **Example:** DAY("1/1/2024") returns 1.

50. **MONTH**

- **Syntax:** MONTH(serial_number)
- Returns the month of a date, as a number from 1 (January) to 12 (December).
- **Example:** MONTH("1/1/2024") returns 1.

51. **YEAR**

- **Syntax:** YEAR(serial_number)
- Returns the year of a date.
- **Example:** YEAR("1/1/2024") returns 2024.

52. **HOUR**

- **Syntax:** HOUR(serial_number)
- Returns the hour of a time value, as a number from 0 (12:00 AM) to 23 (11:00 PM).
- **Example:** HOUR("2:30 PM") returns 14.

53. **MINUTE**

- **Syntax:** MINUTE(serial_number)
- Returns the minute of a time value, as a number from 0 to 59.
- **Example:** MINUTE("2:30 PM") returns 30.

54. **SECOND**

- **Syntax:** SECOND(serial_number)
- Returns the second of a time value, as a number from 0 to 59.
- **Example:** SECOND("2:30:15 PM") returns 15.

55. **WEEKDAY**

- **Syntax:** WEEKDAY(serial_number, [return_type])
- Returns the day of the week corresponding to a date.
- **Example:** WEEKDAY("1/1/2024") returns 2 (Monday).

56. **WORKDAY**

- **Syntax:** WORKDAY(start_date, days, [holidays])
- Returns the date before or after a specified number of workdays.
- **Example:** WORKDAY("1/1/2024", 10) returns the date 10 workdays after January 1, 2024.

57. **NETWORKDAYS**

- **Syntax:** NETWORKDAYS(start_date, end_date, [holidays])
- Returns the number of whole workdays between two dates.
- **Example:** NETWORKDAYS("1/1/2024", "1/10/2024") returns the number of workdays between the two dates.

5. Logical Functions

58. **IF**

- **Syntax:** IF(logical_test, value_if_true, value_if_false)
- Returns one value if a condition is TRUE and another value if it is FALSE.
- **Example:** IF(A1 > 10, "Yes", "No") returns "Yes" if A1 is greater than 10, otherwise "No".

59. **IFERROR**

- **Syntax:** IFERROR(value, value_if_error)
- Returns a value you specify if a formula evaluates to an error; otherwise, it returns the result of the formula.
- **Example:** IFERROR(A1/B1, "Error") returns "Error" if B1 is 0.

60. **AND**

- **Syntax:** AND(logical1, [logical2], …)
- Returns TRUE if all arguments are TRUE.
- **Example:** AND(A1 > 10, B1 < 5) returns TRUE if both conditions are met.

61. **OR**

- **Syntax:** OR(logical1, [logical2], …)
- Returns TRUE if any argument is TRUE.
- **Example:** OR(A1 > 10, B1 < 5) returns TRUE if either condition is met.

62. **NOT**

- **Syntax:** NOT(logical)
- Reverses the logical value of its argument.
- **Example:** NOT(A1 > 10) returns TRUE if A1 is not greater than 10.

6. Financial Functions

63. **PMT**

- **Syntax:** PMT(rate, nper, pv, [fv], [type])
- Returns the payment for a loan based on constant payments and a constant interest rate.
- **Example:** PMT(5%/12, 60, 10000) returns the monthly payment for a loan of $10,000 over 60 months at 5% annual interest.

64. **PV**

- **Syntax:** PV(rate, nper, pmt, [fv], [type])
- Returns the present value of an investment.
- **Example:** PV(5%/12, 60, -200) returns the present value of receiving $200 monthly for 60 months at 5% annual interest.

65. **FV**

- **Syntax:** FV(rate, nper, pmt, [pv], [type])
- Returns the future value of an investment.
- **Example:** FV(5%/12, 60, -200) returns the future value of receiving $200 monthly for 60 months at 5% annual interest.

66. **RATE**

- **Syntax:** RATE(nper, pmt, pv, [fv], [type], [guess])
- Returns the interest rate per period of an annuity.
- **Example:** RATE(60, -200, 10000) calculates the monthly interest rate of a loan of $10,000 with a $200 payment over 60 months.

67. **NPV**

- **Syntax:** NPV(rate, value1, [value2], …)
- Returns the net present value of an investment based on a series of periodic cash flows and a discount rate.
- **Example:** NPV(5%, -10000, 2000, 3000, 4000) calculates the NPV of cash flows.

68. **IRR**

- **Syntax:** IRR(values, [guess])
- Returns the internal rate of return for a series of cash flows.
- **Example:** IRR(A1:A5) returns the IRR for the cash flows in cells A1 to A5.

69. **XIRR**

- **Syntax:** XIRR(values, dates, [guess])
- Returns the internal rate of return for a schedule of cash flows that is not necessarily periodic.
- **Example:** XIRR(A1:A5, B1:B5) calculates the IRR for cash flows in A1 with corresponding dates in B1.

7. Statistical Functions

70. **AVERAGEIF**

- **Syntax:** AVERAGEIF(range, criteria, [average_range])
- Returns the average of the cells that meet a single criterion.
- **Example:** AVERAGEIF(A1:A10, ">10") returns the average of cells in A1 that are greater than 10.

71. **MEDIAN**

- **Syntax:** MEDIAN(number1, [number2], …)
- Returns the median of the given numbers.
- **Example:** MEDIAN(1, 3, 5, 7) returns 4.

72. **MODE.SNGL**

- **Syntax:** MODE.SNGL(number1, [number2], …)
- Returns the most frequently occurring number in a data set.
- **Example:** MODE.SNGL(1, 2, 2, 3) returns 2.

73. **STDEV.P**

- **Syntax:** STDEV.P(number1, [number2], …)
- Calculates standard deviation based on the entire population.
- **Example:** STDEV.P(1, 2, 3, 4) returns the standard deviation of the numbers 1, 2, 3, and 4.

74. **STDEV.S**

- **Syntax:** STDEV.S(number1, [number2], …)
- Calculates standard deviation based on a sample.
- **Example:** STDEV.S(1, 2, 3, 4) returns the standard deviation for the sample data.

75. **VAR.P**

- **Syntax:** VAR.P(number1, [number2], …)

- Calculates variance based on the entire population.
- **Example:** VAR.P(1, 2, 3, 4) returns the variance of the numbers 1, 2, 3, and 4.

76. **VAR.S**

- **Syntax:** VAR.S(number1, [number2], …)
- Calculates variance based on a sample.
- **Example:** VAR.S(1, 2, 3, 4) returns the variance for the sample data.

77. **PERCENTILE.EXC**

- **Syntax:** PERCENTILE.EXC(array, k)
- Returns the k-th percentile of values in a range, excluding the endpoints.
- **Example:** PERCENTILE.EXC(A1:A10, 0.9) returns the 90th percentile of values in A1.

78. **PERCENTILE.INC**

- **Syntax:** PERCENTILE.INC(array, k)
- Returns the k-th percentile of values in a range, including the endpoints.
- **Example:** PERCENTILE.INC(A1:A10, 0.9) returns the 90th percentile of values in A1.

79. **QUARTILE.EXC**

- **Syntax:** QUARTILE.EXC(array, quart)
- Returns the quartile of a data set, based on percentile values, excluding the median.
- **Example:** QUARTILE.EXC(A1:A10, 2) returns the second quartile (median) of A1.

80. **QUARTILE.INC**

- **Syntax:** QUARTILE.INC(array, quart)
- Returns the quartile of a data set, based on percentile values, including the median.
- **Example:** QUARTILE.INC(A1:A10, 2) returns the second quartile (median) of A1.

8. Information Functions

81. **ISNUMBER**
    - **Syntax:** ISNUMBER(value)
    - Checks whether a value is a number.
    - **Example:** ISNUMBER(A1) returns TRUE if A1 contains a number.
82. **ISBLANK**
    - **Syntax:** ISBLANK(value)
    - Checks whether a cell is empty.
    - **Example:** ISBLANK(A1) returns TRUE if A1 is empty.
83. **ISERROR**
    - **Syntax:** ISERROR(value)
    - Checks whether a value is an error.
    - **Example:** ISERROR(A1/B1) returns TRUE if the formula results in an error.
84. **ISEVEN**
    - **Syntax:** ISEVEN(number)
    - Checks whether a number is even.
    - **Example:** ISEVEN(4) returns TRUE.
85. **ISODD**
    - **Syntax:** ISODD(number)
    - Checks whether a number is odd.
    - **Example:** ISODD(3) returns TRUE.
86. **ISTEXT**
    - **Syntax:** ISTEXT(value)
    - Checks whether a value is text.

- **Example:** ISTEXT(A1) returns TRUE if A1 contains text.

87. **ISLOGICAL**

- **Syntax:** ISLOGICAL(value)
- Checks whether a value is logical (TRUE or FALSE).
- **Example:** ISLOGICAL(A1) returns TRUE if A1 is logical.

9. Array Functions

88. **ARRAYFORMULA**

- **Syntax:** ARRAYFORMULA(array_formula)
- Allows functions to return multiple values.
- **Example:** ARRAYFORMULA(A1:A10 + B1:B10) adds corresponding elements from two arrays.

89. **FILTER**

- **Syntax:** FILTER(array, include, [if_empty])
- Filters a range of data based on a condition.
- **Example:** FILTER(A1:A10, B1:B10 > 5) returns values in A1 where B1 is greater than 5.

90. **UNIQUE**

- **Syntax:** UNIQUE(array)
- Returns unique values from a range or array.
- **Example:** UNIQUE(A1:A10) returns the unique values from A1 to A10.

91. **SORT**

- **Syntax:** SORT(array, [sort_index], [sort_order])
- Sorts the contents of a range or array.
- **Example:** SORT(A1:A10, 1, TRUE) sorts A1

  in ascending order.

92. **SEQUENCE**

- **Syntax:** SEQUENCE(rows, [columns], [start], [step])
- Generates a sequence of numbers in an array.
- **Example:** SEQUENCE(5) generates a vertical array of 5 numbers (1, 2, 3, 4, 5).

93. **SPILL**

- **Syntax:** SPILL(reference)
- Refers to the range of spilled values from a dynamic array formula.
- **Example:** SUM(SPILL(A1)) sums the dynamic array starting at A1.

10. Database Functions

94. **DAVERAGE**

- **Syntax:** DAVERAGE(database, field, criteria)
- Returns the average of selected database entries.
- **Example:** DAVERAGE(A1:C10, "Sales", E1:E2) returns the average sales value meeting criteria in E1.

95. **DCOUNT**

- **Syntax:** DCOUNT(database, field, criteria)
- Counts the cells containing numbers in a database column that meets specified criteria.
- **Example:** DCOUNT(A1:C10, "Sales", E1:E2) counts the number of sales entries meeting the criteria in E1.

96. **DSUM**

- **Syntax:** DSUM(database, field, criteria)
- Adds the numbers in a database column that meets specified criteria.
- **Example:** DSUM(A1:C10, "Sales", E1:E2) adds the sales values meeting criteria in E1.

97. **DGET**

- **Syntax:** DGET(database, field, criteria)
- Extracts a single value from a database that matches specified criteria.
- **Example:** DGET(A1:C10, "Sales", E1:E2) returns the single sales value meeting criteria in E1.

11. Cube Functions

98. **CUBEVALUE**

- **Syntax:** CUBEVALUE(connection, member_expression1, [member_expression2], …)
- Returns an aggregated value from a cube.
- **Example:** CUBEVALUE("Sales", "[Products].[All Products].[Bikes]") returns the value for bikes from a cube data source.

99. **CUBEMEMBER**

- **Syntax:** CUBEMEMBER(connection, member_expression, [caption])
- Returns a member from a cube.
- **Example:** CUBEMEMBER("Sales", "[Products].[All Products].[Bikes]") returns the "Bikes" member.

100. **CUBERANKEDMEMBER**

- **Syntax:** CUBERANKEDMEMBER(connection, set_expression, rank, [caption])
- Returns the nth or ranked member in a set.
- **Example:** CUBERANKEDMEMBER("Sales", "[Top Products]", 1) returns the top-ranked product.