



## FINANCE AUTOMATION – EXCEL LAMBDA FUNCTIONS

FUNCTIONS	PURPOSE	USE CASE	LAMBDA
1 Remove_Duplicates	Combines up to 4 arrays and returns a unique list of non-blank values.	Ideal for deduplicating data like customer emails, product SKUs, or vendor names collected from multiple sources.	=LAMBDA(Array1, Array2, Array3, Array4, UNIQUE(FILTER(VSTACK(Array1, IF(ISOMITTED(Array2), "", Array2), IF(ISOMITTED(Array3), "", Array3), IF(ISOMITTED(Array4), "", Array4)), VSTACK(Array1, IF(ISOMITTED(Array2), "", Array2), IF(ISOMITTED(Array3), "", Array3), IF(ISOMITTED(Array4), "", Array4)) <> "")))
2 Extract_Text	Extracts a specific number of characters from a cell starting from a defined search text using MID and SEARCH	Helpful when you need to extract codes, IDs, or values that appear after a specific keyword or pattern in unstructured text—such as pulling invoice numbers after "INV", or GSTIN codes from notes.	=LAMBDA(Cell_Ref, Text_to_Search, No_of_Digit_req, MID(Cell_Ref, SEARCH(Text_to_Search, Cell_Ref), No_of_Digit_req))
3 Vlookup_Auto	Automates VLOOKUP by dynamically determining the column number using MATCH, allowing flexible lookups based on column headers.	Ideal when working with tables where the column positions may change but headers remain fixed—such as pulling sales, quantity, or date data by header name instead of hardcoding column numbers.	=LAMBDA(Lookup_Value1, Lookup_Array, Column_Value, Column_Range, IFERROR(VLOOKUP(Lookup_Value1, Lookup_Array, MATCH(Column_Value, Column_Range, 0), 0), 0))
4 Dual_Lookup	Performs a VLOOKUP using a combination of two lookup values, useful when a unique match depends on two keys instead of one.	Ideal in cases where you need to fetch data based on a composite key, like finding the price of a product for a specific region (Product + Region), or employee info by (Employee ID + Department).	=LAMBDA(Lookup_Value1, Lookup_Value2, Lookup_Array, Column_Value, Column_Range, IFERROR(VLOOKUP(Lookup_Value1 & Lookup_Value2, Lookup_Array, MATCH(Column_Value, Column_Range, 0), 0), 0))
5 Xlookup_Auto	Performs a dynamic two-way lookup using XLOOKUP, allowing retrieval of data at the intersection of a row and column based on their labels—like a matrix lookup.	Perfect for cross-referencing data in a matrix table where you want to fetch values based on both a row label (e.g., product name) and a column header (e.g., month or region).	=LAMBDA(Row_Lookup, Row_Array, Return_Array, Col_Lookup, Col_Array, XLOOKUP(Row_Lookup, Row_Array, XLOOKUP(Col_Lookup, Col_Array, Return_Array, "Column Heading Not Found", 0), "Not Found Data", 0))
6 Xlookup_Multiples	Performs a dual-condition XLOOKUP both for rows and columns, allowing lookups with up to two keys for both dimensions—row and column—making it a powerful two-dimensional lookup function.	Useful when you need to find data in a matrix where the lookup depends on two row values (e.g., Product + Region) and two column headers (e.g., Month + Metric), like in multi-dimensional financial reports or sales dashboards.	=LAMBDA(Lookup_Value1, Lookup_Value2, Optional, Lookup_Range1, Lookup_Range2, Optional, Column_Value1, Column_Value2, Optional, Column_Range1, Column_Range2, Optional, Data_Range, XLOOKUP(1, (Lookup_Range1 = Lookup_Value1) * (IF(ISOMITTED(Lookup_Value2, Optional), 1, (Lookup_Range2, Optional = Lookup_Value2, Optional))), XLOOKUP(1, ((Column_Range1 = Column_Value1) * (IF(ISOMITTED(Column_Value2, Optional), 1, (Column_Range2, Optional = Column_Value2, Optional))), Data_Range, "Column Header Not Found", 0), "Data Not Found"))
7 Check_Availability	Checks if a specific value exists in a given array and returns "Available" if found, or "Not Available" if not.	Useful for validating presence of items like SKU codes, employee IDs, customer names, or invoice numbers in master lists or records.	=LAMBDA(Lookup_Value, Lookup_Array, IF(ISNUMBER(MATCH(Lookup_Value, Lookup_Array, 0)), "Available", "Not Available"))
8 Net_Amount	Returns the Debit amount if "D" is selected, otherwise returns the Credit amount—useful for extracting net values from a dual-column entry system.	Ideal in accounting reports where Debit and Credit amounts are recorded in separate columns, and you want to dynamically extract the correct amount based on the transaction type.	=LAMBDA(DrCr, Debit_Amount, Credit_Amount, IF(DrCr = "D", Debit_Amount, Credit_Amount))
9 Number_Extract	Extracts the first occurring number and everything after it from a given text string—useful for isolating embedded numbers in mixed-format data.	Ideal when dealing with invoice labels, reference numbers, or item codes like "Ref#12345X" or "Bill-INV786", where you want to pull out the number part programmatically.	=LAMBDA(Cell_Ref, MID(Cell_Ref, MIN(SEARCH({0,1,2,3,4,5,6,7,8,9}, Cell_Ref & "0123456789")), LEN(Cell_Ref) - MIN(SEARCH({0,1,2,3,4,5,6,7,8,9}, Cell_Ref & "0123456789")) + 1))



## FINANCE AUTOMATION – EXCEL LAMBDA FUNCTIONS

FUNCTIONS	PURPOSE	USE CASE	LAMBDA
<b>10</b> <b>Min_Value_Lookup</b>	Finds and returns a value from a specified range based on a matching lookup value and the minimum value in another related range (like date, amount, etc.).	Useful in scenarios where you want to retrieve the first occurrence, earliest date, or lowest price related to a specific item, customer, or ID—for example, getting the earliest purchase date or the lowest bid for a vendor.	<code>=LAMBDA(Lookup_Value, Lookup_Range, Min_Value_Range, Return_Range, XLOOKUP(1,(Lookup_Range = Lookup_Value) * (Min_Value_Range = MINIFS(Min_Value_Range, Lookup_Range, Lookup_Value)),Return_Range,0,0))</code>
<b>11</b> <b>Max_Value_lookup</b>	Returns a value from a specified range where the lookup condition matches and the associated value is the maximum within that group—based on a metric like amount, date, or score.	Best used when you need to find the latest transaction, highest sale, or maximum score for a specific entity like a customer, product, or employee.	<code>=LAMBDA(Lookup_Value, Lookup_Range, Max_Value_Range, Return_Range, XLOOKUP(1,(Lookup_Range = Lookup_Value)*(Max_Value_Range = MAXIFS(Max_Value_Range, Lookup_Range, Lookup_Value)),Return_Range,0,0))</code>
<b>12</b> <b>Next_Min_Lookup</b>	Returns a value from a specified range based on a lookup value and the Nth smallest value in a related column, using SMALL() for ranked lookups (e.g., 2nd lowest, 3rd lowest).	Useful when you want to retrieve the second-best offer, next minimum price, or next earliest date for a particular item or category—especially in bidding, inventory, or sales analysis.	<code>=LAMBDA(Lookup_Value, MinValue_Num, Lookup_Range, MinValue_Lookup_Range, Return_Range,XLOOKUP(1,(Lookup_Range = Lookup_Value) * (MinValue_Lookup_Range = SMALL(MinValue_Lookup_Range, MinValue_Num)),Return_Range,0,0))</code>
<b>13</b> <b>Next_Max_Lookup</b>	Returns a value based on a lookup condition and the Nth largest value (e.g., 2nd highest, 3rd highest) from a related range using LARGE().	Best suited for situations where you want to identify the second-highest bid, next top sale, or Nth best performance for a specific category like a customer, product, or region.	<code>=LAMBDA(Lookup_Value, MaxValue_Num, Lookup_Range, MaxValue_Lookup_Range,Return_Range,XLOOKUP(1,(Lookup_Range = Lookup_Value) * (MaxValue_Lookup_Range = LARGE(MaxValue_Lookup_Range, MaxValue_Num)),Return_Range,0,0))</code>