Learning to Extract Numbers: A Guide to Removing Non-Numeric Characters in Google Sheets

Authored by

Mohammed looti

June 2, 2026

Welcome to this essential guide on data cleaning within Google Sheets. When dealing with complex datasets, particularly those involving identifiers, codes, or phone numbers, it is often necessary to isolate purely numeric values by eliminating extraneous symbols, letters, or punctuation. This process ensures data integrity and prepares your sheets for accurate numerical analysis or database migration. Fortunately, Google Sheets offers a powerful function, **REGEXREPLACE**, which utilizes the flexibility of Regular Expressions to achieve this transformation efficiently and reliably.

The following formula represents the simplest and most effective method for removing all non-numeric characters from a specified cell. Understanding this structure is fundamental to mastering advanced text manipulation within your spreadsheets.

=REGEXREPLACE(A2,"D+", "")

Specifically, this robust formula is designed to target cell **A2**, identifying and replacing any character that is not a digit with an empty string, thereby leaving only the numerical components of the original data intact. We will delve deeper into the specific components of this expression, including the powerful regular expression pattern `D+`, to demonstrate its utility in various data cleansing scenarios requiring the removal of unwanted textual elements.

Mastering Data Cleaning with REGEXREPLACE

Effective spreadsheet management invariably relies on the ability to standardize data inputs. When imported or manually entered data contains inconsistent formatting--such as dashes, parentheses, currency symbols, or trailing text mixed with numbers--standard filtering or mathematical operations become impossible. The **REGEXREPLACE** function provides a surgical solution, allowing users to define exactly what needs to be removed based on complex patterns, rather than relying solely on simple, single-character matching. This capability is particularly valuable in environments where data consistency is paramount, such as preparing contact lists or financial transaction records for external systems that demand strict numerical formatting.

The primary goal of employing this function is achieving data normalization. By stripping away all **non-numeric characters**, we ensure that every entry in a column meant to contain numerical data conforms strictly to that requirement. For instance, if you are calculating the average duration of a task, but some entries include the unit (e.g., "15 min" or "20 seconds"), removing the letters and spaces ensures only the pure numerical value remains, allowing for accurate statistical calculation. This foundational step of data cleaning dramatically reduces calculation errors downstream, significantly enhancing the reliability of any subsequent analysis performed in Google Sheets.

While simple functions like `SUBSTITUTE` or `REPLACE` might suffice for removing a single, known

character (like a single dash), they quickly become cumbersome when dealing with a variety of unwanted symbols simultaneously. **REGEXREPLACE** handles all such variations--including parentheses, dashes, commas, and letters--in one concise operation through the use of a single, powerful regular expression pattern. This efficiency is the key differentiator and why **REGEXREPLACE** is the preferred tool for professional spreadsheet analysts tasked with large-scale data transformation projects where speed and accuracy are crucial.

Understanding the REGEXREPLACE Syntax and Logic

To fully appreciate the power of the solution, it is vital to dissect the structure and required arguments of the **REGEXREPLACE** function. This function requires three primary arguments to operate successfully, following the structure: `REGEXREPLACE(text, regular_expression, replacement)`. Each component plays a crucial role in defining the source data, the pattern to be matched, and the resulting output after the substitution occurs. Mastery of these components allows for highly customized data manipulation far beyond simple predefined character removal.

The first argument, `text` (represented by `A2` in our initial formula), specifies the cell or string containing the source data you wish to clean. This is the raw material from which the **non-numeric characters** will be extracted. The function reads the entire contents of this cell before attempting to match any patterns defined in the second argument. When applying the formula across an entire column, it is critical that this reference uses relative addressing (like `A2`) so that it adjusts correctly when the formula is dragged down the column to process subsequent rows.

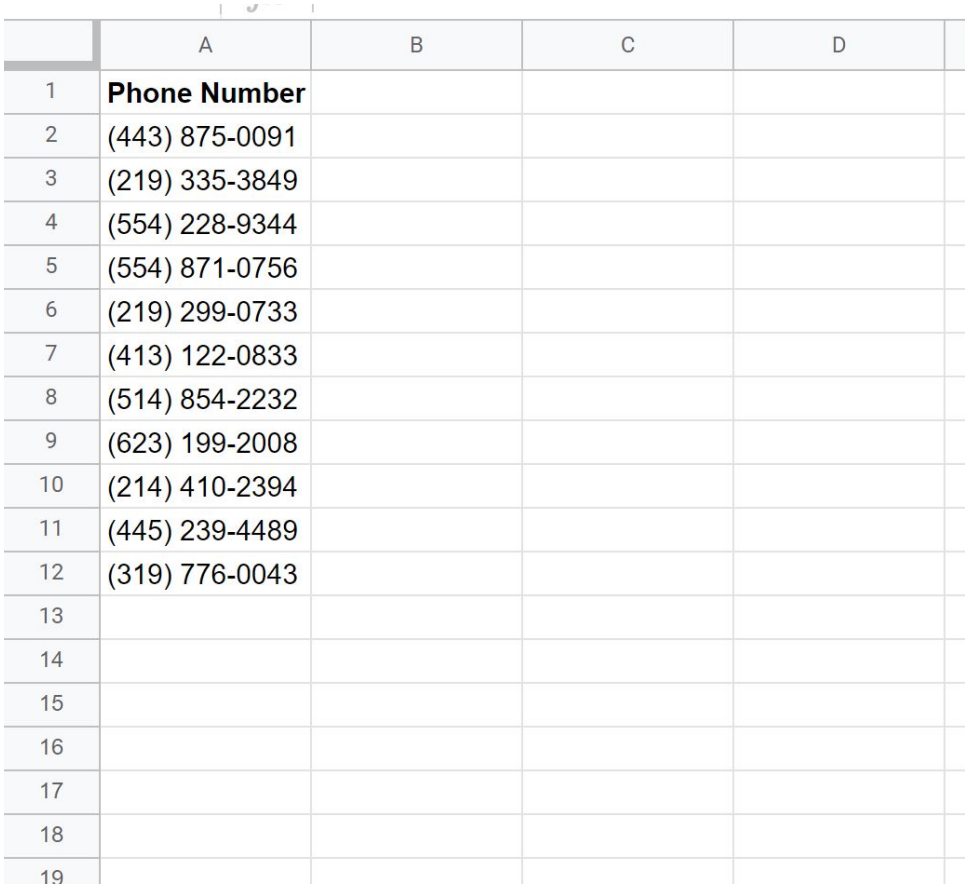
The second argument, the `regular_expression` (represented by `"D+"`), is the core operational component. This pattern defines precisely which characters or sequences should be identified for removal. The specific pattern `D` is a predefined shorthand within Regular Expressions that succinctly stands for "any character that is NOT a digit" (i.e., not 0 through 9). Furthermore, the plus sign (`+`) is a powerful quantifier, indicating that the preceding pattern (`D`) should be matched one or more times consecutively. Together, `D+` ensures that every sequence of non-digit characters, regardless of its length, is captured and treated as a single match, making the substitution process highly efficient.

Finally, the third argument, `replacement` (represented by `""`), dictates what the matched patterns will be replaced with. In the context of data extraction, using an empty string (two double quotes with nothing between them) instructs REGEXREPLACE to effectively delete the matched **non-numeric characters**, leaving behind only the pure digits. If, alternatively, you intended to replace all non-numeric separators with a specific space or dash, you would input that desired character within the quotes instead. However, for achieving the goal of isolating numbers, the empty string is the essential input.

Practical Demonstration: Sanitizing Phone Number Data

To demonstrate this technique practically, let us consider the common challenge of cleaning up a list of phone numbers. Phone number data is frequently inconsistent, often including international symbols, parentheses around area codes, and various types of dashes or spaces used as separators. Before these numbers can be imported into a Customer Relationship Management (CRM) system or used for automated dialing, they must be standardized into a uniform, clean numerical string, typically 10 or 11 digits long.

Suppose we begin with the following list of phone numbers stored in Column A of our Google Sheets document. This example clearly shows the variety of inconsistent formatting applied to the entries, which needs immediate remediation:



	A	B	C	D
1	**Phone Number**			
2	(443) 875-0091			
3	(219) 335-3849			
4	(554) 228-9344			
5	(554) 871-0756			
6	(219) 299-0733			
7	(413) 122-0833			
8	(514) 854-2232			
9	(623) 199-2008			
10	(214) 410-2394			
11	(445) 239-4489			
12	(319) 776-0043			
13				
14				
15				
16				
17				
18				
19				

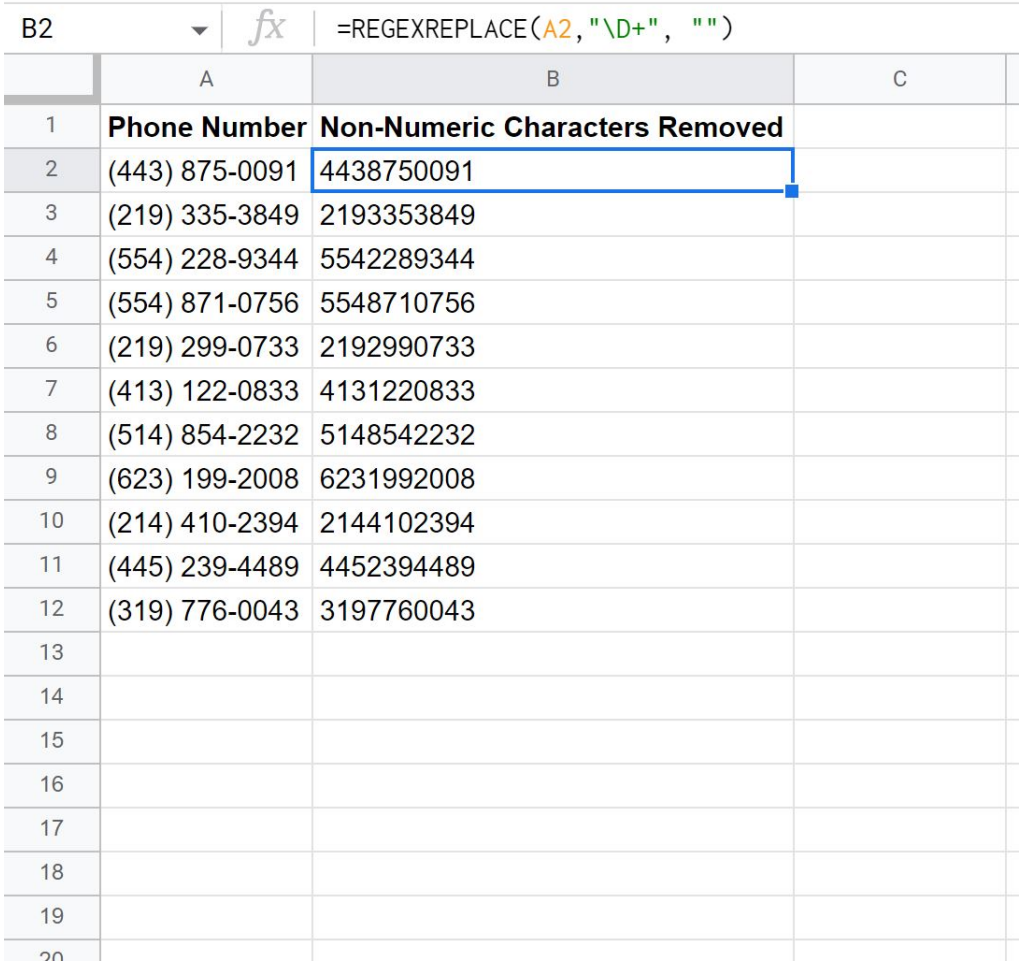
Our objective is straightforward: we need a robust method to systematically remove all extraneous, **non-numeric characters**--including the parentheses, various spaces, and hyphens--from each phone number entry so that we are left exclusively with the underlying numerical digits. This precise task is perfectly suited for the **REGEXREPLACE** function due to its capability to handle multiple, varied character types simultaneously using the generic `D+` pattern, ensuring a complete cleanse.

We initiate the process by applying the formula in the adjacent column, Column B, specifically starting in cell **B2**. The formula targets the corresponding cell in Column A (A2) and implements the removal logic we previously detailed. The consistent application of this formula guarantees uniformity across the entire dataset, regardless of how complex or inconsistent the original formatting may have been across individual entries.

=REGEXREPLACE(A2,"D+", "")

Once this formula is correctly entered into cell **B2**, the crucial final step involves utilizing the fill handle--the small square located at the bottom right corner of the selected cell--and dragging it down through all remaining applicable cells in Column B. This action automatically adjusts the cell reference (progressing from A2 to A3, A4, and so on), applying the sophisticated cleansing logic to every subsequent phone number in the list. This technique is an indispensable practice in spreadsheet operations for efficiently processing large volumes of raw data.

The result of this operation is immediately visible, providing a perfectly standardized dataset in Column B:



	A	B	C
1	**Phone Number**	**Non-Numeric Characters Removed**	
2	(443) 875-0091	4438750091	
3	(219) 335-3849	2193353849	
4	(554) 228-9344	5542289344	
5	(554) 871-0756	5548710756	
6	(219) 299-0733	2192990733	
7	(413) 122-0833	4131220833	
8	(514) 854-2232	5148542232	
9	(623) 199-2008	6231992008	
10	(214) 410-2394	2144102394	
11	(445) 239-4489	4452394489	
12	(319) 776-0043	3197760043	
13			
14			
15			
16			
17			
18			
19			
20			

Interpreting the Results and Expanding Use Cases

Upon reviewing the output in Column B, it is clear that the **REGEXREPLACE** function has successfully executed the necessary data cleaning operation with high precision. Notice specifically that all complex, non-essential characters, such as the parentheses used for the area code and the various dashes separating the number groups, have been entirely eliminated. The resulting values are pure strings of digits, immediately ready for use in any database or system requiring normalized, standardized numerical input. This powerful transformation underscores the critical advantage of using Regular Expressions over more limiting, conventional text functions when dealing with highly variable data formats.

This technique is highly versatile and extends well beyond the simple standardization of phone numbers. It is equally applicable in numerous other data manipulation contexts where the precise isolation of numerical data is required for analysis or transfer. It is crucial for maintaining the integrity of data fields that must strictly contain only numbers.

Consider these specific additional use cases where the `=REGEXREPLACE(A2,"D+", "")` structure proves invaluable:

Extracting Product IDs: If inventory records list product codes with mixed alphanumeric strings, such as "SKU-453987-US" or "P_ID_009901", applying the formula extracts only the core numerical identifier ("453987" or "009901"), facilitating numerical lookups.

Cleaning Measurement Data: Stripping units from technical measurements, effectively converting entries like "50kg," "100ft," or "200ml" into their pure numerical counterparts ("50," "100," and "200" respectively), allowing the data to be used directly in mathematical operations without error.

Standardizing Currency: Removing currency symbols, commas, and other non-numeric separators from financial data, thereby transforming entries such as "$1,234" into "1234." Note that maintaining the decimal point for fractional values requires a slight modification to the pattern to exclude the period from the removal set, which is an advanced adjustment to the core principle.

While `D+` is optimized for removing all **non-numeric characters**, advanced users frequently encounter scenarios where they must preserve specific non-digit symbols, such as a decimal point or a hyphen used as a negative sign. In such situations, the Regular Expression pattern can be modified using a negative character set. For instance, to remove everything except digits and the decimal point, you would use the pattern . Understanding these subtle nuances allows for highly precise and customized control over the data extraction process, cementing **REGEXREPLACE** as an indispensable and versatile tool for complex data preparation tasks within Google Sheets.

Note: It is highly recommended that users wishing to perform more intricate text manipulations

familiarize themselves with the full capabilities of the **REGEXREPLACE** function and the broader syntax of Regular Expressions by consulting the complete official documentation. Detailed information regarding advanced patterns and specific usage examples can be found on the dedicated support pages for Google Sheets functions.

Additional Resources for Google Sheets Mastery

The successful application of regular expressions, such as that demonstrated by **REGEXREPLACE**, unlocks a vast array of possibilities for data transformation and validation in spreadsheets. To continue building expertise in advanced data cleaning techniques and other common operations in Google Sheets, we highly recommend exploring the following relevant tutorials and official documentation links. These resources provide deeper insights into functions that complement text manipulation and help optimize your daily data management workflow.

Learning Advanced Regular Expression Syntax for Complex Text Matching and Extraction.

Tutorial: How to use the `ARRAYFORMULA` function in conjunction with text manipulation functions for column-wide processing.

Complete official documentation for the `REGEXREPLACE` function, including edge cases and advanced pattern definitions.

By integrating these powerful textual manipulation functions into your daily workflow, you can ensure that your data remains consistently clean, standardized, and perfectly ready for accurate analysis, moving your processes beyond cumbersome manual cleanup toward automated, reliable data management practices.