

Generate a Random Password in Excel (With Example)

Authored by
stats writer

November 17, 2025

RECOMMENDED CITATION

stats writer (2025). *Generate a Random Password in Excel (With Example)*.
PSYCHOLOGICAL SCALES. Retrieved from <https://scales.arabpsychology.com/?p=94550>

The Core Formula for Generating Secure Passwords in Excel

Generating robust, randomized passwords is a fundamental aspect of digital security. While dedicated password managers are often the preferred tool, Microsoft Excel offers powerful built-in functions that allow users to create complex, multi-character passwords directly within a spreadsheet. The following formula utilizes a combination of character functions and random number generation to produce an 8-character random password that includes uppercase letters, lowercase letters, and digits.

```
=CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)&CHAR(RANDBETWEEN(97,122))&CHAR(RANDBETWEEN(97,122))&CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)&CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)
```

This extensive formula is simply a concatenation of eight distinct segments, each designed to generate one specific character type. By strategically mixing the functions, we ensure that the resulting password meets typical complexity requirements, featuring variety and unpredictability across its structure. Understanding how each component contributes to the final string is essential for customizing the password generator to specific needs, such as increasing length or adding special characters.

Understanding the Essential Excel Functions

The complex password generation formula relies on two primary Excel functions working in tandem: RANDBETWEEN and CHAR. The RANDBETWEEN function is responsible for selecting an integer uniformly at random between a specified bottom and top boundary. This provides the core element of unpredictability. Conversely, the CHAR function takes a numeric code (specifically an ASCII code) and returns the character corresponding to that code.

When these two are nested--for example, `CHAR(RANDBETWEEN(65,90))`--RANDBETWEEN first selects a random number, and then the CHAR function translates that number into a usable character. For numerical components (0-9), we only need RANDBETWEEN, but we must use the Concatenation operator (the & symbol) to join the resulting character string with the generated digit. This careful mixing ensures the final password is a single, cohesive text string.

The three core components used repeatedly in the comprehensive formula are defined by the ranges they target:

CHAR(RANDBETWEEN(65,90)) generates a random uppercase letter by selecting from the ASCII codes for capital A through Z.

RANDBETWEEN(0,9) generates a random number between 0 and 9, which is implicitly converted

to a string when concatenated.

CHAR(RANDBETWEEN(97,122)) generates a random lowercase letter by targeting the ASCII codes for lowercase a through z.

We then use the & symbol to concatenate (join) each of these random values together into one continuous output string, creating the final password structure.

Decoding the ASCII Codes Used in Password Generation

The key to generating characters rather than just numbers lies in leveraging the standardized ASCII codes. Each character on your keyboard, including letters, numbers, and symbols, corresponds to a unique decimal value. By specifying a range of these decimal values within the RANDBETWEEN function, we instruct Excel to randomly select a code from the desired character set.

For instance, the range 65 through 90 exclusively maps to the 26 uppercase letters of the English alphabet (A=65, B=66, ..., Z=90). Similarly, the range 97 through 122 maps directly to the 26 lowercase letters (a=97, b=98, ..., z=122). This separation is crucial for ensuring character diversity and fulfilling common security requirements that necessitate both cases.

While the formula only uses 0-9 for numbers (which are outputted directly by RANDBETWEEN), if one wished to generate numerical characters using the CHAR function, the corresponding ASCII codes would be 48 through 57. However, the current formula's approach of directly using RANDBETWEEN(0,9) is simpler and achieves the same result when joined via concatenation. Understanding these code ranges allows users to easily expand the formula to include special characters (e.g., symbols like 33-47 for punctuation) if greater complexity is required.

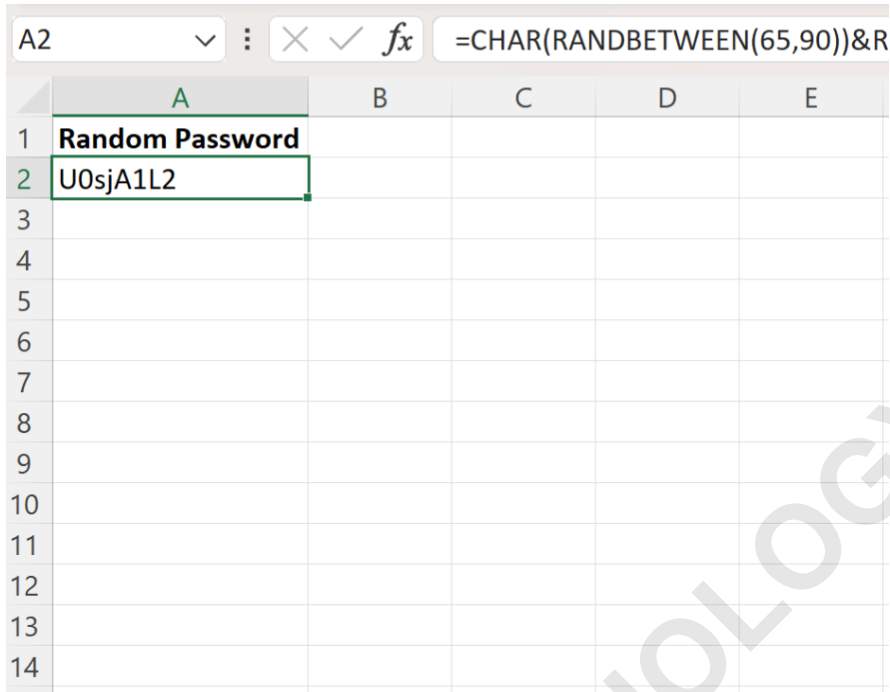
Example: Generate Random Password in Excel

To demonstrate this process in practice, we will use the full 8-character formula designed to create a complex password incorporating uppercase letters, lowercase letters, and numbers. This example showcases the immediate application and the resulting random output.

We begin by selecting an empty cell, such as cell **A2**, and pasting the complete formula. This specific sequence ensures a randomized mix of characters across the 8 positions, maximizing the resulting password's entropy and making it more difficult to guess or crack.

```
=CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)&CHAR(RANDBETWEEN(97,122))&CHAR(RANDBETWEEN(97,122))&CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)&CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)
```

Upon pressing **Enter**, Excel processes the formula, executing the RANDBETWEEN functions eight times and converting the required numerical outputs to their respective characters via the CHAR function. The resulting string appears instantly in cell A2.



| | A | B | C | D | E |
|----|-----------------|---|---|---|---|
| 1 | Random Password | | | | |
| 2 | U0sjA1L2 | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |

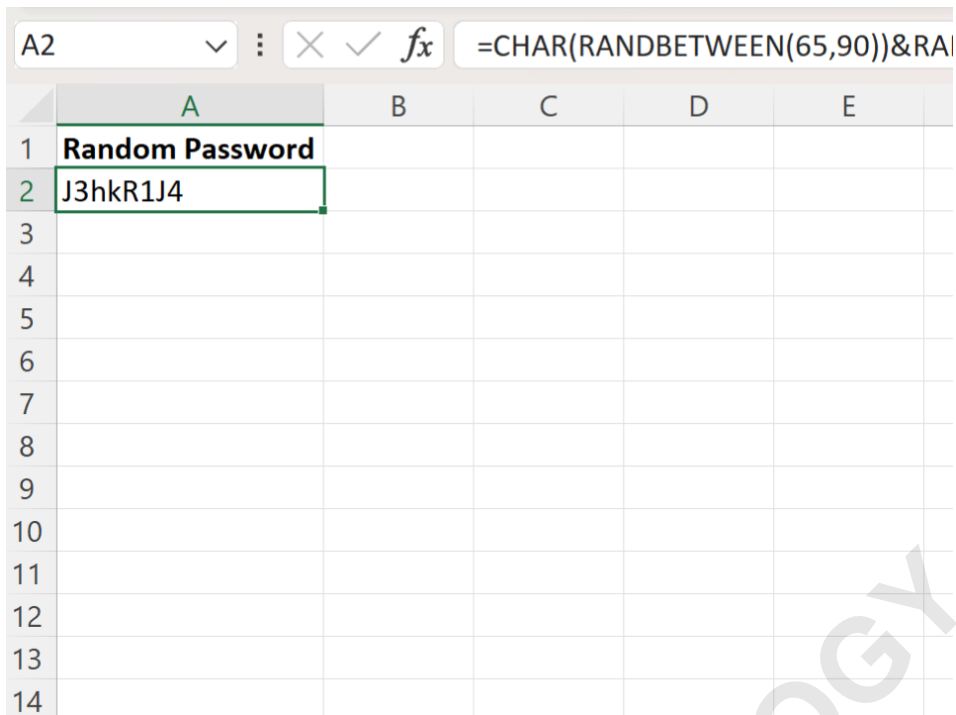
In this initial run, the formula successfully generates the random password **U0sjA1L2**. This output immediately confirms the functional success of the concatenation logic, integrating uppercase, lowercase, and numerical components into a single, strong string.

The Volatility of RANDBETWEEN and Recalculation

An important characteristic of the RANDBETWEEN function, and consequently this entire password generation formula, is that it is a **volatile function**. Volatile functions recalculate their results every time the spreadsheet changes, or every time the sheet is refreshed or manually recalculated. This means that if you perform any action--typing into another cell, deleting content, or even simply pressing **F9**--the password in cell A2 will immediately change.

This volatility is highly beneficial for generating multiple unique passwords quickly. To generate a brand new random password, one does not need to re-enter the formula; simply double-click on cell **A2** and press **Enter**, forcing a recalculation of that specific formula instance. This recalculation triggers a new set of random numbers, yielding a completely different output string.

Observing the effect of the recalculation, we can see the output change completely from the initial result:



| | A | B | C | D | E |
|----|-----------------|---|---|---|---|
| 1 | Random Password | | | | |
| 2 | J3hkR1J4 | | | | |
| 3 | | | | | |
| 4 | | | | | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |

Following the forced refresh, the formula now generates the random password **J3hkR1J4**. Due to the inherent randomness of the RANDBETWEEN function, every time the formula executes, it provides a unique and unpredictable combination of characters, ensuring high diversity among generated passwords. Once you have generated the desired password, it is crucial to copy the value (not the formula) and paste it as static text (using Paste Special: Values) to prevent it from changing unexpectedly.

Customizing Password Length and Complexity

The initial 8-character password formula is a template that can be easily modified to suit specific length and complexity requirements. Security policies often mandate lengths exceeding eight characters, and achieving this is simply a matter of adding more concatenated segments to the formula. The fundamental building blocks remain the same, allowing for infinite customization.

Remember that the following three core functions dictate the types of characters included:

CHAR(RANDBETWEEN(65,90)) to generate a random uppercase letter.

RANDBETWEEN(0,9) to generate a random number between 0 and 9.

CHAR(RANDBETWEEN(97,122)) to generate a random lowercase letter.

For example, if a short, 3-character alphanumeric code is needed--perhaps for an internal access key--you can reduce the formula significantly by selecting only three components and concatenating them. This flexibility allows the user to precisely control the structure and character

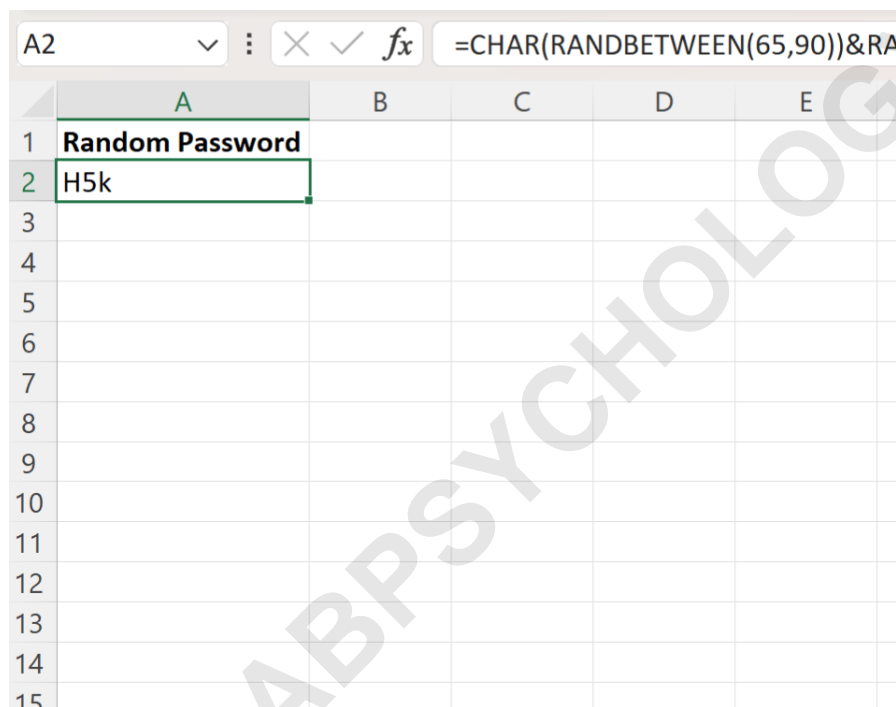
mixture.

A simplified, 3-character example utilizing one of each type might look like this:

```
=CHAR(RANDBETWEEN(65,90))&RANDBETWEEN(0,9)&CHAR(RANDBETWEEN(97,122))
```

This formula is significantly shorter but still combines the necessary elements for a mixed-case, alphanumeric output.

The following screenshot illustrates the result of this truncated formula, demonstrating how easily the complexity can be scaled down:



In this case, the formula generates the compact password **H5k**. Users are encouraged to experiment with different combinations of these functions to generate random passwords with any specific length, distribution, or complexity level required by their security protocols.

Limitations and Security Considerations

While Excel provides a highly accessible method for generating random passwords, it is important to acknowledge the limitations associated with using spreadsheet functions for cryptographic purposes. The primary limitation stems from the inherent nature of the underlying algorithms used by functions like `RANDBETWEEN`. These are not true random number generators; rather, they are pseudorandom number generators (PRNGs).

A pseudorandom number generator relies on a deterministic algorithm and a starting value (known as a seed) to produce a sequence of numbers that appear random but are ultimately predictable if the seed and algorithm are known. For high-security applications, such as generating encryption keys or master passwords for enterprise systems, relying on Excel's PRNG is strongly discouraged in favor of cryptographically secure random number generators (CSPRNGs) used in dedicated security tools.

However, for internal testing, sample data generation, or generating randomized, non-critical access codes, this Excel technique is efficient and effective. Users should always prioritize dedicated, audited password management tools for any sensitive accounts. This method serves as an excellent practical demonstration of combining string manipulation and randomization in Excel, but its security context must be carefully considered based on the criticality of the system the password is protecting.

ARABPSYCHOLOGY.COM