Use Google applications more efficiently with Google Apps Script

With Google Apps Script, you can add menus, custom dialogs, write functions and macros, build add-ons for Google Docs, Sheets and Slides.

When you use the Google application, you're sure you have used up all of its features. With Google Apps Script, you can add menus, custom dialogs, write functions and macros, build add-ons for Google Docs, Sheets and Slides.

1. Some basic features of Google Docs
2. How to create Google Spredsheet automatically updates data
3. 10 tips to create beautiful Google Docs

What is Google Apps Script?

Google Apps Script is a cloud-based development platform for creating lightweight, custom web applications. You can build applications that extend directly inside the browser, integrating easily with Google products.

Apps Script uses JavaScript that brings familiarity when combining web development with Google products, making it a perfect tool for customizing enterprise applications, organizing or automating collaborations. service.

1. Task automation tools on Windows 10

You can create two types of scripts with Google Apps Script:

1. **Standalone** : These scripts are not limited to any service like Google Docs, Sheets or Slides. They can perform functions across the system, like macros. However, they cannot be shared with multiple objects because you need to copy and paste the code to use. For example, you can create scripts to search files on Drive with specific names or see who has the right to share files and folders on Drive.

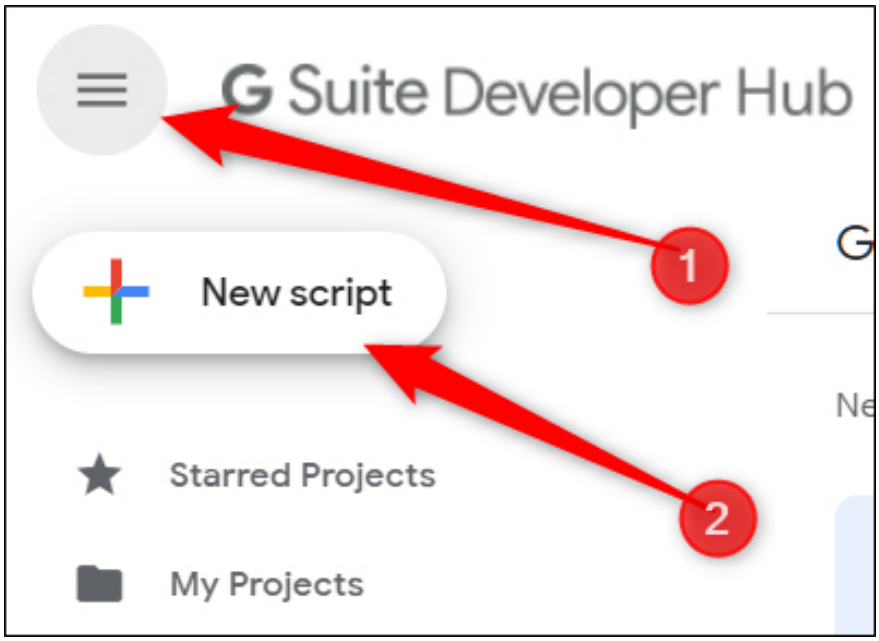
2. **Bound** (Limit): This script is linked to Google Docs, Sheets, Forms or Slides files. These script extensions extend the functionality of the file and only perform actions in that particular file. For example, you can script to add custom menus, dialogs, and sidebar to a service or script that sends an email to notify you whenever a specific box in the Sheet changes.

If you don't know much about JavaScript or you may have never heard of it before, don't worry because using Apps Script is easy, it provides many documents and examples that you can test yourself. Here are some examples for you to understand how they work.

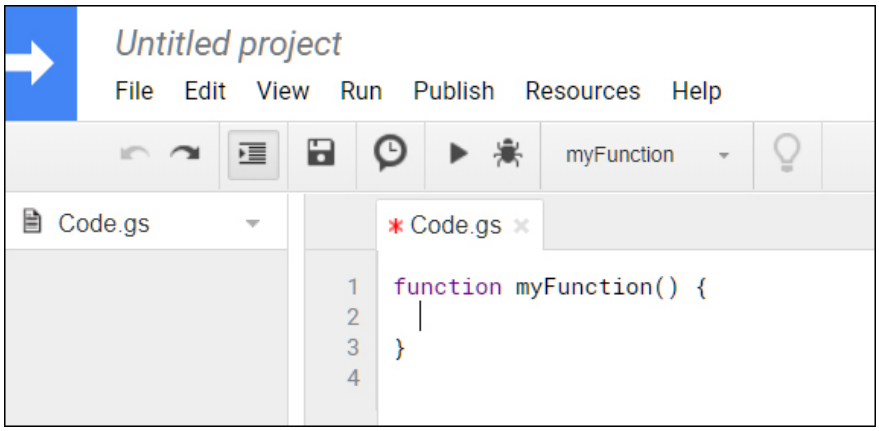
How to create Standalone scripts

Here will use a sample code from Google as an example and will explain these lines of code if you are not familiar with Google Script or JavaScript.

Access to Google Apps Scripts (https://script.google.com/home). In the top left corner, click on the three horizontal bar icon, then click **New Script** .



A new non-title project opens with an empty internal function, but because it uses code from Google, you can delete all the text in the file.



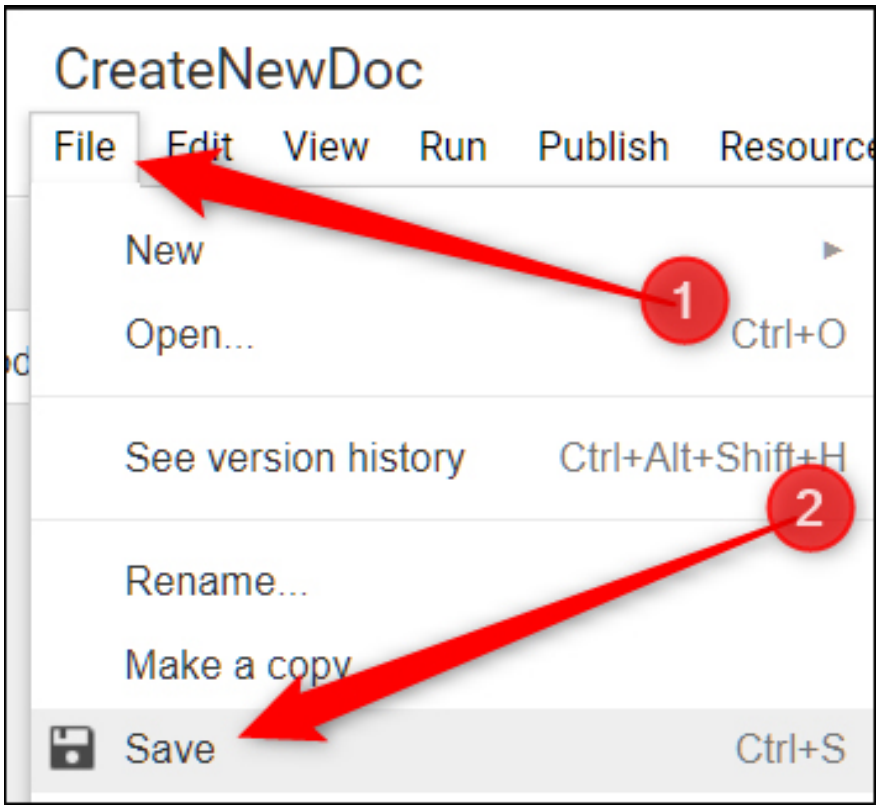
Note : You need to log in to your Google account in order for this script to work.

After deleting the preloaded code in the file, paste the following code:

```
// Initialize your function
function createADocument () {
```

```
// Create a new Google Doc named 'Hello, world!'
var doc = DocumentApp.create ('Hello, world!');

// Access the body of the document, then add a paragraph.
doc.getBody (). appendParagraph ('This document was created by Google Apps Scri
}
```

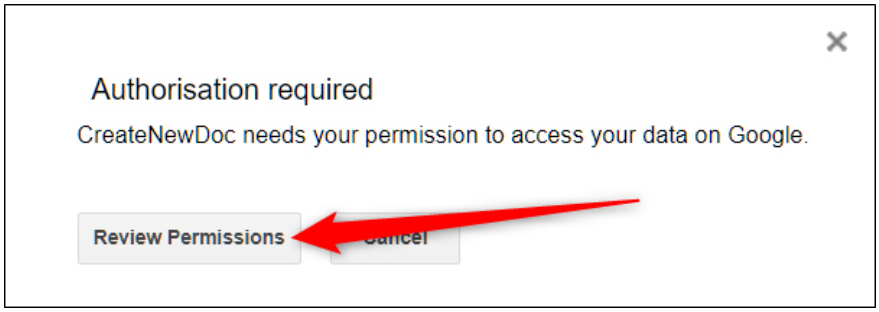
Before running the code, you must save this script by clicking **File**> **Save** .



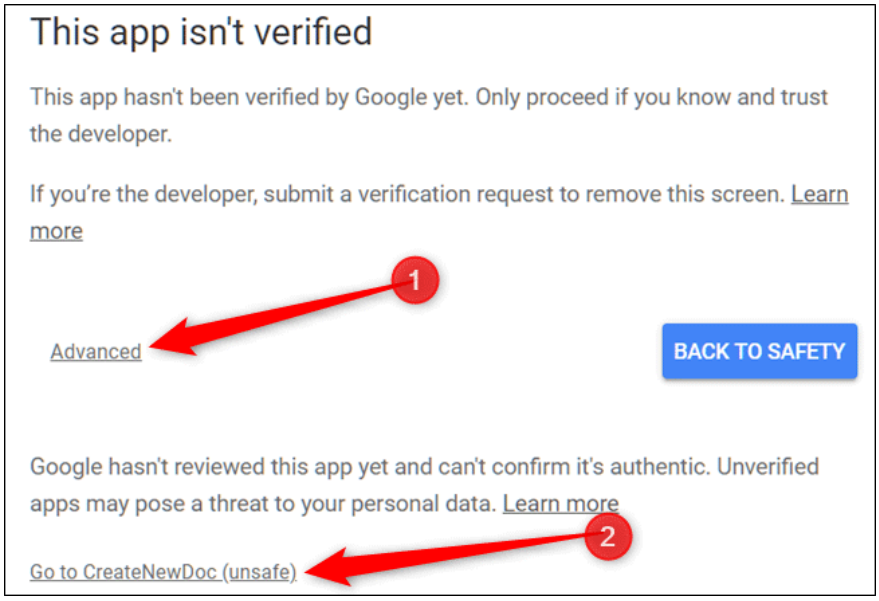
Rename the project, then click **OK** .

To run the code, click on the triangle icon on the toolbar.

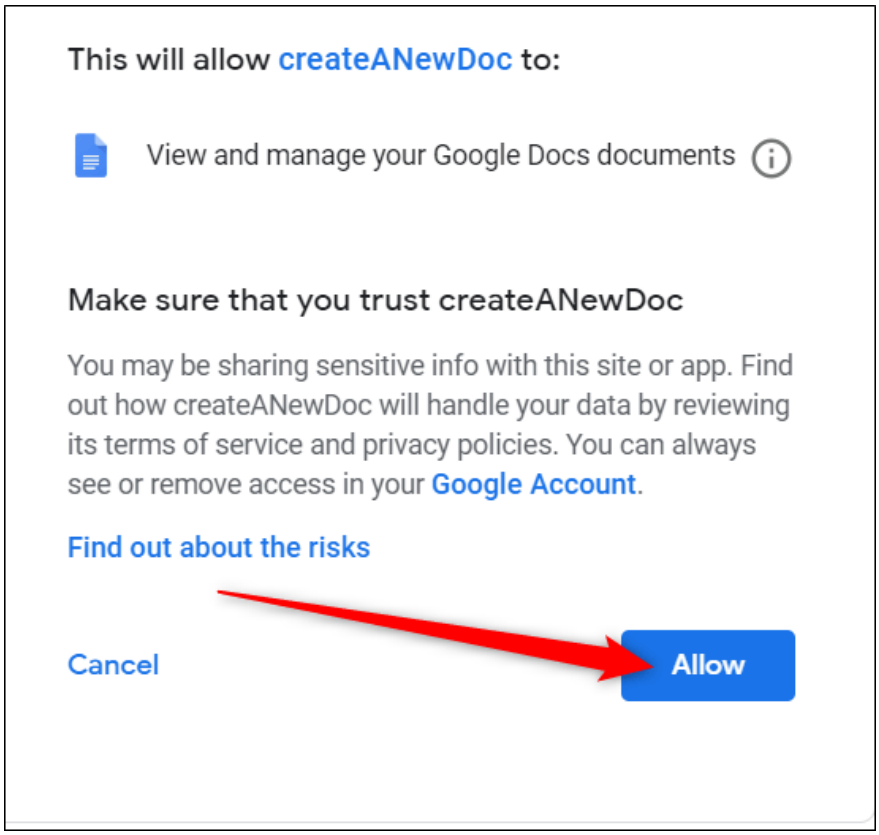
You will have to grant the script permission to access your Google account through the pop-up window after clicking **Run** for the first time. Click **Review Permissions** to see what permissions the script needs.



Since this is not an application verified by Google, you will receive another warning. Basically it says, you should only proceed if you trust the developer. Click **Advanced** , then click **Go to CreateNewDoc** (or whatever name you have set for this script).

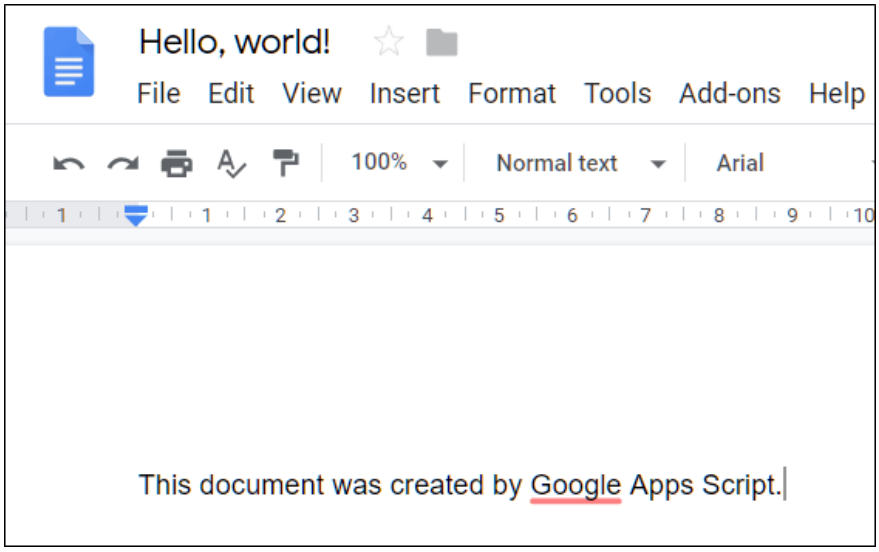


Review the permissions that the script requires, then click **Allow** .



Now, go to your Drive and if everything is right, you will see the file **Hello, World!** Here, double click on it to open.

When opening the file, you will see a line of text from the code added to your document.



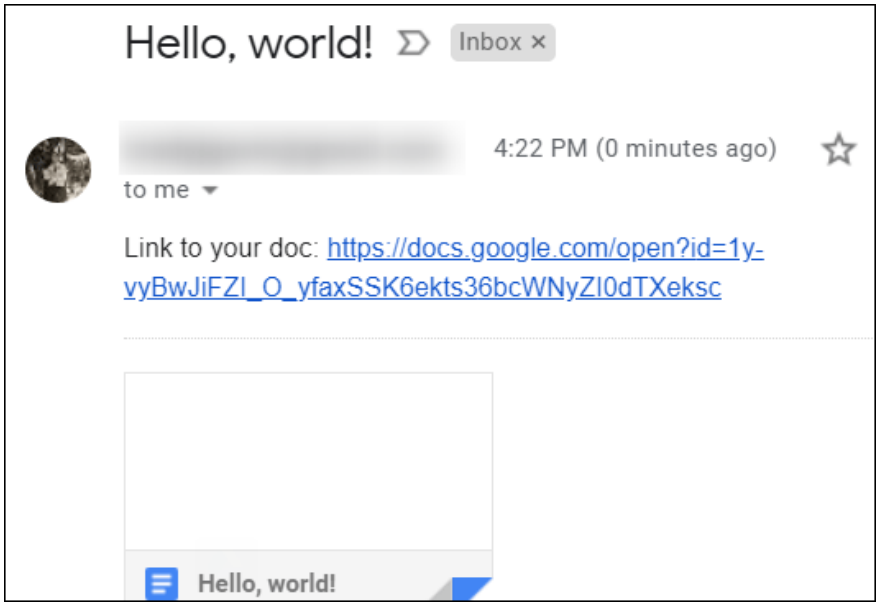
Now, if you want to receive email notifications when the document is created, you can add a few lines of code to automatically send to your Google account. Add the following lines of code after doc.getBody (). AppendParagraph ('This document was created by Google Apps Script.'); but before the last curly brace:

```
// Get the URL of the document.
var url = doc.getUrl ();
// Get the address email of the active user - that's you.
var email = Session.getActiveUser (). getEmail ();

// Get the name of the document ?? s? d?ng nh? là dòng ??u th?.
var subject = doc.getName ();

// Append a new string to the variable "url" to use as an email body.
var body = 'Link to your doc:' + url;

// Send yourself an email with a link to tài li?u.
GmailApp.sendEmail (email, subject, body);
```

Click the triangle icon. Since you have added a few additional lines that require additional permissions, you should follow the same steps, click **Review Permissions** > **Advanced** , then click **Go to CreateNewDoc** .

Note : Because Google warns you about launching an unverified application, you'll receive a security warning email. Google does this in case you suspect that you are not an authority for an unverified application.

Preview the permissions granted to the script, then click **Allow** . Once the document is created, you will receive an email with the link file in Google Drive.

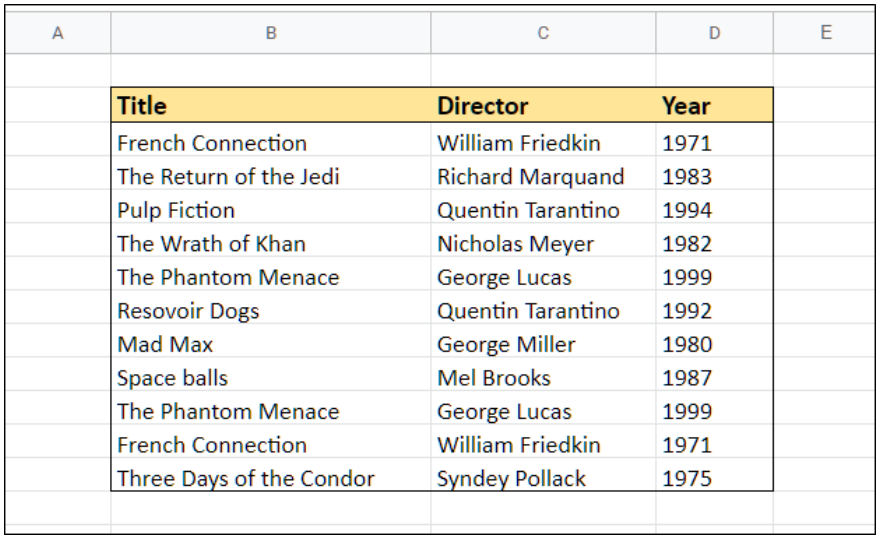


Click on this link to directly open the file inside Google Drive.

How to create a Bound script

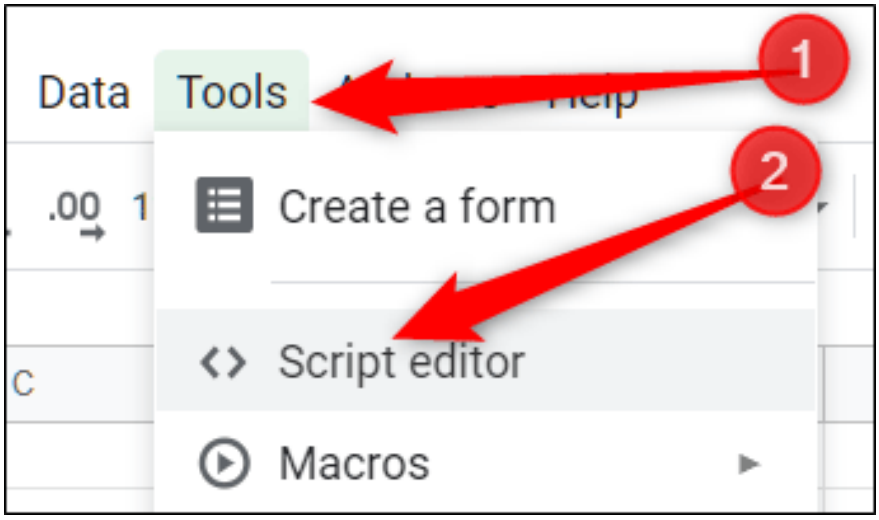
Now we will create a limited script for Google Sheets to analyze the current worksheet that has the same item in a row and then delete it.

As mentioned, Bound script acts as an add-on for a specific file. To create a limited script, open the available Google Sheets sheet containing at least one duplicate data.



Title	Director	Year
French Connection	William Friedkin	1971
The Return of the Jedi	Richard Marquand	1983
Pulp Fiction	Quentin Tarantino	1994
The Wrath of Khan	Nicholas Meyer	1982
The Phantom Menace	George Lucas	1999
Resovoir Dogs	Quentin Tarantino	1992
Mad Max	George Miller	1980
Space balls	Mel Brooks	1987
The Phantom Menace	George Lucas	1999
French Connection	William Friedkin	1971
Three Days of the Condor	Syndey Pollack	1975

Click on **Tool** and then **Script Editor** .



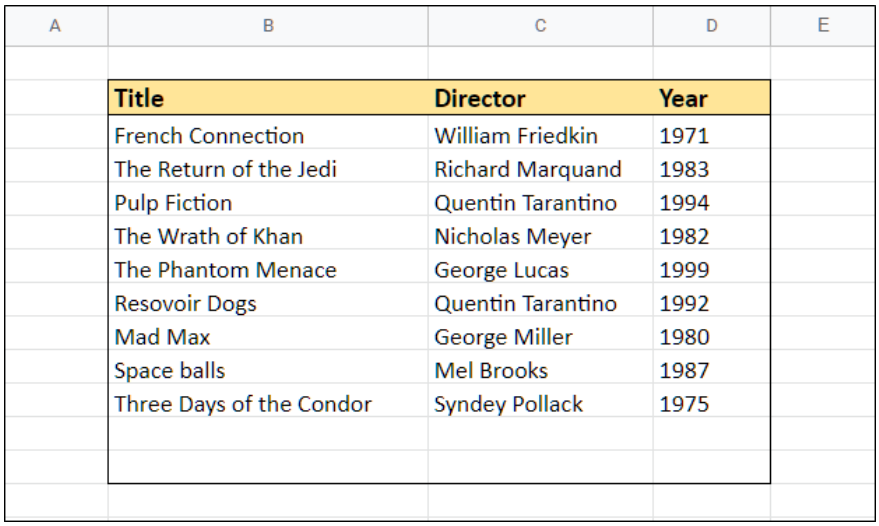
Google Apps Script opens in a new tab with a blank script. However this time the script is limited in the form of Sheet. Just like before, delete the blank row and paste the following code:

```
  // Removes duplicate rows from the current sheet.

  function removeDuplicates () {
  // Get current active Spreadsheet
  var sheet = SpreadsheetApp.getActiveSheet ();
  // Get all values ??from spreadsheet's hàng
  var data = sheet.getDataRange (). getValues ??();
  // Create an array for non-duplicates
  var newData = [];
  // Iterate qua m?t ???ng d?n c?a hàng
  cho (var i trong d? li?u) {
  var row = data [i];
  var duplicate = false;
  cho (var j trong newData) {
  if (row.join () == newData [j] .join ()) {
  duplicate = true;
  }
  }
  // If not a duplicate, put in newData array
  if (! duplicate) {
  newData.push (row);
  }
  }
  // Delete the Sheet old and insert array này
  sheet.clearContents ();
  sheet.getRange (1, 1, newData.length, newData [0] .length) .setValues ??
(newData);
  }
```

Note : For scripts that delete duplicate data, all cells in the row must match.

Save and rename the script, then press the Run icon. Same as above, you must preview the required script permissions and grant it access to the sheet. Click **Review Permissions** . Then accept the message and click **Allow** to grant the script permission.

After running, go back to the Sheet and you will see the duplicate items disappear.



Title	Director	Year
French Connection	William Friedkin	1971
The Return of the Jedi	Richard Marquand	1983
Pulp Fiction	Quentin Tarantino	1994
The Wrath of Khan	Nicholas Meyer	1982
The Phantom Menace	George Lucas	1999
Resovoir Dogs	Quentin Tarantino	1992
Mad Max	George Miller	1980
Space balls	Mel Brooks	1987
Three Days of the Condor	Syndey Pollack	1975

However, if the data inside a table is like the above example, this script will not change the size of the table to fit the number in it.

The above are two simple examples of how to use Apps Script, but it has many other options depending on your intended use.

I wish you all success!