

☐

I'm not robot


reCAPTCHA

Next



Keyboard Shortcuts

Windows Shortcuts — Playback and navigation

F5 — View slide show from first slide
Shift + F5 — View slide show from current slide
Control + N — Opens new presentation file
Control + O — Open file
Control + S — Save
Control + W — Close presentation
Control + M — Insert new slide
Page Up — Move to previous slide
Page Down — Move to next slide
Control + P — Print
Control + F1 — Show/hide the Ribbon
Control + Q — Quit PowerPoint
Alt + F — Backstage View
Alt + M — Slide Master tab
Alt + H — Home tab
Alt + R — Review tab
Alt + N — Insert tab
Alt + W — View tab
Alt + G — Design tab
Alt + A — Animations tab
Alt + S — Slide Show tab
Alt + K — Transitions tab
Formatting and editing
Control + B — Bold text
Control + I — Italicize text
Control + U — Underline text
Control + A — Select all
Control + F — Find text
Control + H — Find and replace text, also hide pointer and navigation buttons (in Slide Show view)
Control + Shift + H — Show/hide Notes pane
Control + Z — Undo
Control + Y — Repeat or redo
Control + X — Cut selected text, formatting, or objects
Control + C — Copy selected text, formatting, or objects
Control + V — Paste copied text, formatting, or objects
Control + Shift + D — Duplicate active slide
Control + G — Group selected slide objects
Control + Shift + G — Ungroup selected slide objects
Tab — Indent text
Esc — Return to Normal view from Reading or Slide Show view
Control + E — Center-align text
Control + L — Left-align text
Control + R — Right-align text
Control + K — Insert hyperlink
Control + Home — Move cursor to first word in a text box, go to the first slide in the presentation
Control + End — Move cursor to the last word in a text box, go to the last slide in the presentation
Control + Backspace — Delete one word to the left
Control + Delete — Delete one word to the right
Control + Enter — Move to next text placeholder, add new slide if no text placeholders exist
Shift + F3 — Toggle case of selected text
F7 — Run spellcheck

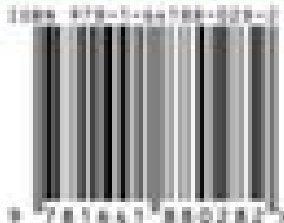
macOS Shortcuts — Playback and navigation

Command + Shift + Return — Play slide show from first slide
Command + Return — Play slide show from the current slide
Command + N — New document
Command + O — Open file
Command + S — Save file
Command + W — Close document
Command + P — Print presentation
Command + Q — Quit
Command + Shift + N — Insert new slide
Command + Shift + D — Make copy of the selected slide
Command + Minus Sign — Zoom out
Command + Plus Sign — Zoom in
Command + 1 — Switch to Normal view
Command + 2 — Switch to Slide Sorter view
Command + 3 — Switch to Notes Page view
Command + 4 — Switch to Outline view
Command + Shift + Return — Switch to slide show
Command + Control + F — Switch to full screen (hide menus)
Option + Return — Switch to Presenter view
Command + Option + Control + G — Show or hide guides
Command + Home or Command + Fn + Left Arrow — Beginning of document
Command + End or Command + Fn + Right Arrow — End of document
Esc, Command + Period, or Hyphen — End a slide show
Formatting and editing
Command + B — Bold text
Command + I — Italicize text
Command + U — Underline text
Command + A — Select all
Command + K — Insert link
Command + D — Duplicate selected objects
Command + F — Find text and formatting
Command + Z — Undo
Command + Y — Repeat or redo
Command + X — Cut selected text, formatting, or objects
Command + C — Copy selected text, formatting, or objects
Command + V — Paste copied text, formatting, or objects
Option + Left Arrow — Move cursor one word to the left
Option + Right Arrow — Move cursor one word to the right
Command + Delete — Delete one word to the left
Command + Fn + Delete — Delete one word to the right
Command + Option + G — Group the selected objects
Command + Option + Shift + G — Ungroup the selected objects
Page Up — Up one screen
Page Down — Down one screen
Shift + F3 or Fn + Shift + F3 — Toggle case of selected text
F7 or Fn + F7 — Check Spelling
Tab — Select the next object
Shift + Tab — Select the previous object

This reference is for informational purposes only. The author and the publisher cannot be held liable for any direct, indirect, incidental, consequential, or special damages of any kind, or any damages whatsoever, arising out of or in connection with the use of this reference. 30 Media Corporation and its publications are not associated with Microsoft PowerPoint® or any other Microsoft product. The PowerPoint 2019 Reference and Cheat Sheet is an independent publication and is not affiliated with, nor has it been authorized, sponsored, or otherwise approved by Microsoft Corporation.

Copyright © 2019 30 Media Corporation.
IN 30 MINUTES® is a registered trademark
of 30 Media Corporation. All rights reserved.

Printed in the USA



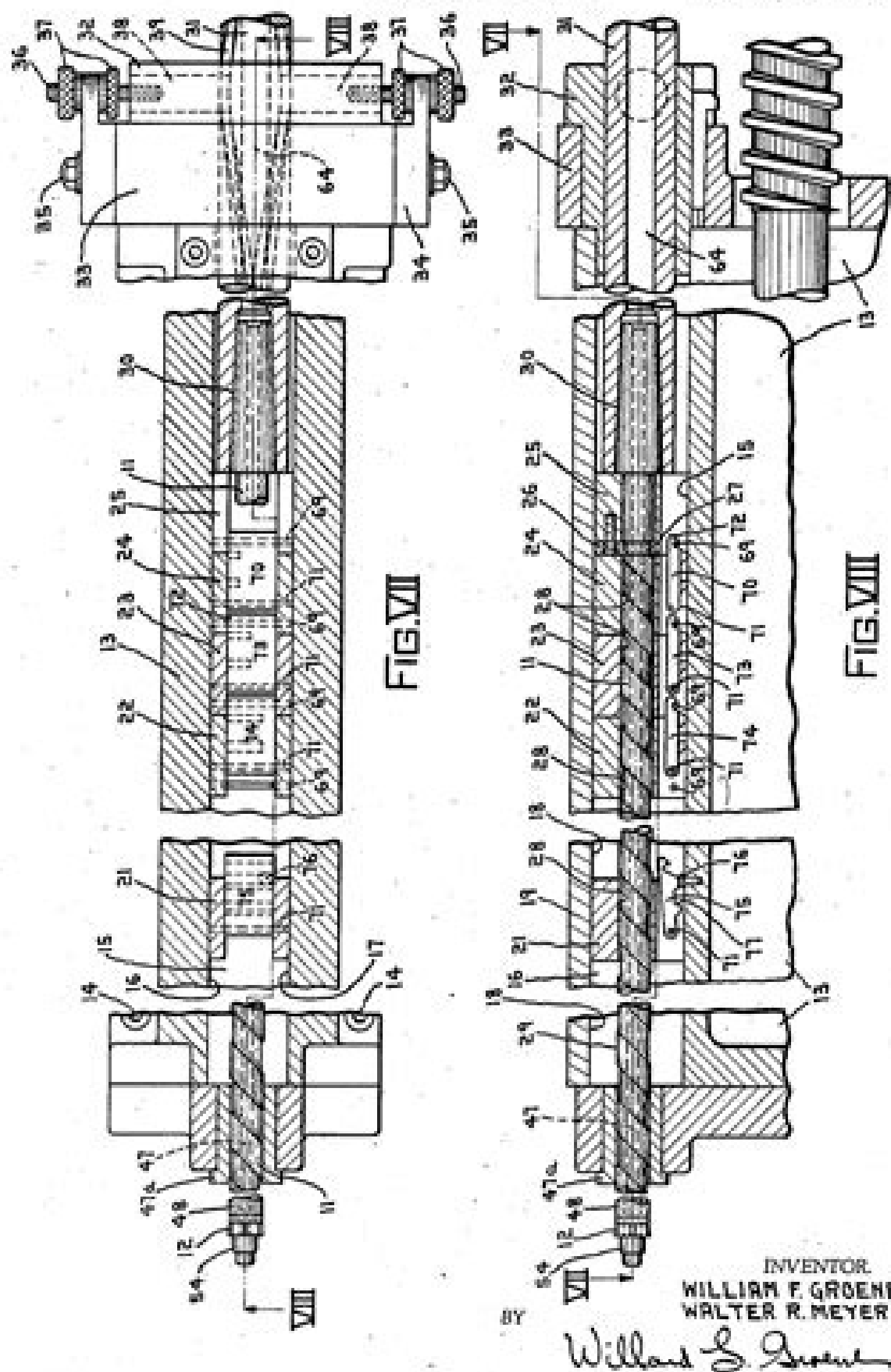
May 18, 1943

W. F. GROENE ET AL
GUN RIPLING MACHINE

2,319,451

Filed June 18, 1941

7 Sheets-Sheet 5



Dec. 3, 1968

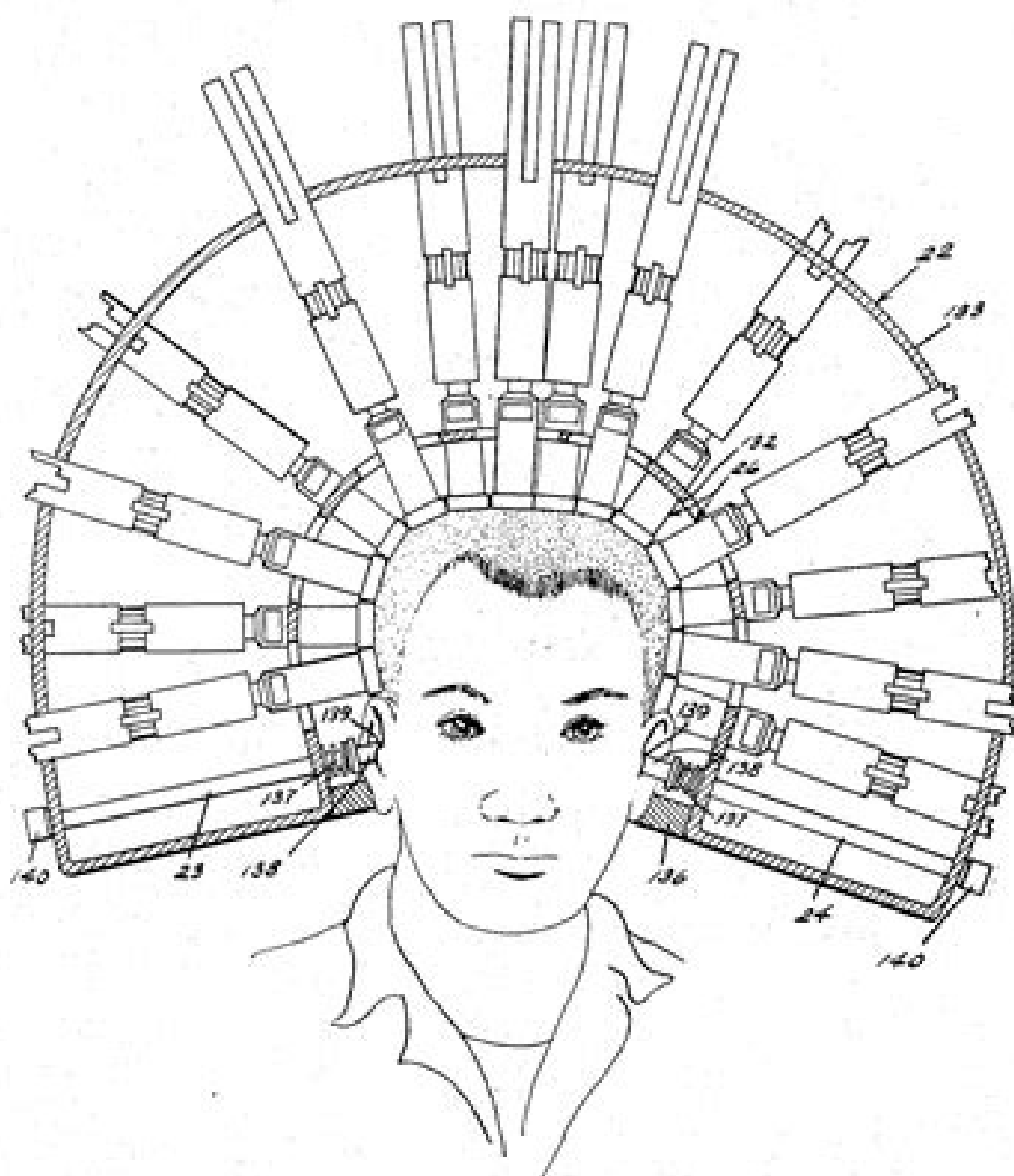
J. A. DLOUHY ETAL
HAIR CUTTING APPARATUS HAVING MEANS FOR CUTTING HAIR IN
ACCORDANCE WITH PREDETERMINED HAIR STYLES AS
A FUNCTION OF HEAD SHAPE

3,413,985

Filed Nov. 28, 1962

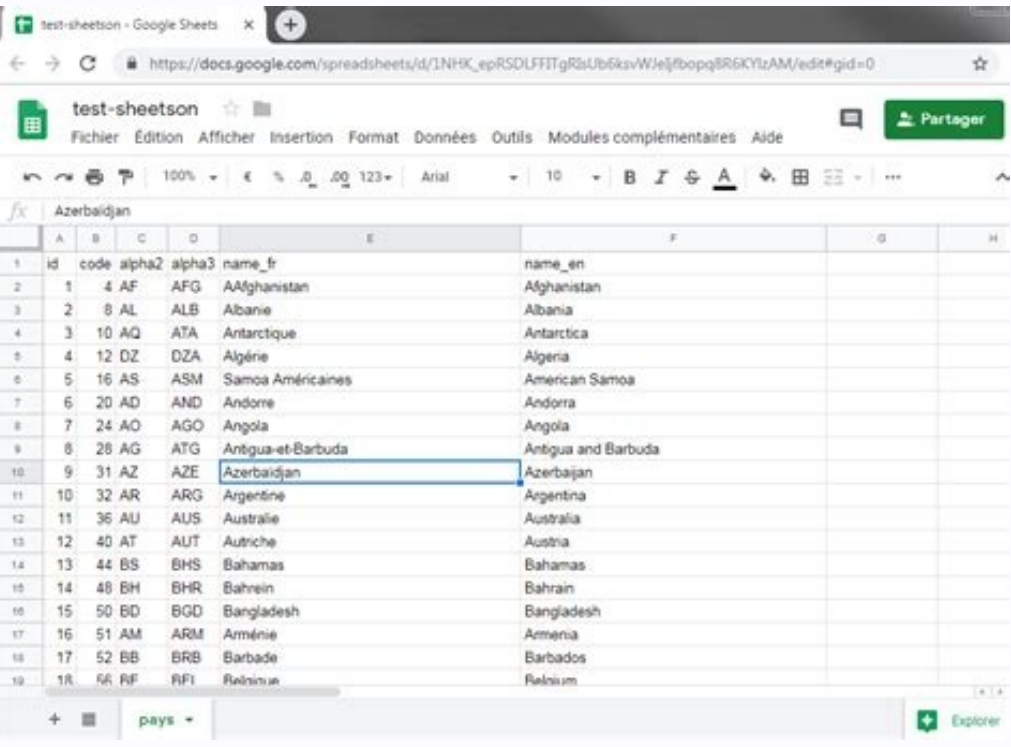
8 Sheets-Sheet 3

FIG. 4



INVENTORS
JOHN A. DLOUHY
WILLIAM E. REYNOLDS
BY JOHN BALLA
Hill, Sherman, Merriam, Jones & Thompson
ATTORNEYS

```
<people>
  <jason>
    <person type = 'fictional'>
      <first_name>
        <married>
          Jason
        </married>
      </first_name>
      <last_name>
        Bourne
      </last_name>
      <occupation>
        Spy
      </occupation>
    </person>
  </jason>
  <carol>
    <person type = 'real'>
      <first_name>
        <married>
          Carol
        </married>
      </first_name>
      <last_name>
        Kalp
      </last_name>
      <occupation>
        Scientist
      </occupation>
    </person>
  </carol>
</people>
```



In this API tutorial for beginners, you will learn how to connect to the API through Google Apps Script, to retrieve data from third parties and show them on your Google Telculo. Example 1 Show how to use Google Apps Script to connect to a simple API to retrieve some data and display them in Google Sheets: In Example 2, we will use Google Apps Script to create a music discovery application using the iTunes API: Finally, In Example 3, I will leave you try to create a Star Wars data scanning application, with some tips: API tutorial for beginners: What is an API? You probably heard the API TA © rhine before. Maybe you have heard how technology companies use them when they channel data between their applications. Or how companies build complex systems from many smaller micro-services linked by API, instead of as unique monolithic programs today. API means «Interface of application programs», and the term commonly refers to Web URLs that can be used to access raw data. Basically, the API is an interface that provides raw data for the public to use (although many require some kind of authentication). As third-party software developers, we can access the API of an organization and use your data in our own applications. The good news is that there is a simple API lot by there, with which we can cut our teeth. We will see three of them in this API tutorial for beginners. We can connect a Google sheet to an API and bring data from that API (for example, iTunes) to our Google sheet using Google Apps Script. It is fun and very satisfying if you are new in this world. API tutorial for beginners: What is Apps Script? In this API tutorial for beginners, we will use Google Apps Script to connect to external APIs. Google Script is a scripting language based on JavaScript hosted and executed on Google servers, which extends Google Apps functionality. If you have never used it before, check out my post: Google Apps Script: Guide for beginners Example 1: Connect Connect Leaves of the numbers API. We are going to start something very simple in this API tutorial for beginners, so you can focus on the data and not miss on the lines and lines of code. We write a short program that calls the API numbers and requests a fact of basic mathematics. Step 1: Open a new sheet Open a new blank sheet of Google and change the name: API numbers Example Step 2: Go to the Application Script Editor Navigate to Tools> Script Editor ... Step 3: Name your project Open a new eye and this is where we write our code. Project name: API numbers Example Step 4: Add an API sample code Remove all the code that is currently in the file code.gs, and replace it with this: Callnumbers () function { // Call the API numbers for the fact of random mathematics Var Answer = urlfetchapp.fetch (""); Recorder.log (response.getcontenttext ()); } We are using the URLFetChapp class to communicate with other applications on the Internet to access resources, to obtain a URL. Now, the window of its code should be seen as: Step 5: Run your function Run the function by clicking on the Play button on the toolbar: Step 6: Authorize your script. This will ask you to authorize your script to connect to an external service. Click on "ventilation permissions" and then ", take care of". Step 7: See congratulations, your program has now been executed. A request was sent to a third party for some data (in this case, a fact of random mathematics) and that service has responded with that data. But wait, where is it? How do we see what the data? Well, you will notice the line 5 of our previous code was logger.log (...), which means we have recorded the response text in our log files. So let's check it. Go to see> Records: You will see your answer (of course, have a different fact): [17.02.03 08: 52: 41: 236 PST] 1158 is the maximum number of pieces that can be a bull. Cut with 18 cuts. It looks like this in the pop-up: Great! Try to run it several times, check the logs and see will see facts. Next, try changing the URL to these examples to see some different data in the answer: 1729 You can also place them directly in your browser if you want to play with them. More information on the Numbers API page. So, what if we want to print the result to our spreadsheet? Well, that's pretty easy. Step 8: Add Data to Sheet Add these few code lines (lines 7, 8 and 9) under your existing code: Run Callnumbers () { // Call API numbers for the FACT response FAB VAR = urlfetchapp.fetch ("http: //numbers) api.com/random/math "); Logger.log (response.getcontenttext ()); var dap = response.getcontenttext (); sheet var = SpreadsheetApp.getactivesheet (); sheet.getRange (1,1).setValue ([done]); } Line 7 Simply assigns the response text (our data) to a variable called fact, so we can consult it using that name. Line 8 is expected from our current active sheet (number sheet of the example spreadsheet API) and assigns it to a variable called sheet, so that we can access it with that name. Finally, on line 9, we get cell A1 (range to 1.1) and set the value in that cell to equalize the fact of the variable, which supports the answer text. Step 9: Run and re-authorize your program again. You will be asked to allow your script to view and manage your spreadsheets on Google Drive, so click Allow: Step 10: Check the external data on your spreadsheet You must now get the random fact that appears on your Google Sheet: How cool is that! To summarize our progress so far on this beginner API tutorial: We have requested data from a third party service on the Internet. That service has responded with the data we wanted and now we have generated it on our Google sheet! Step 11: new cell data The script, as it is written in this API tutorial for beginners, will always overwrite the A1 cell with its new fact every time it executes the program. If you want to create a list and continue adding new facts.The existing ones, then make this change less than line 9 of your code (which is shown below), to write the answer to the first blank row: Function Callnumbers () { // Call the API numbers for the response of FACT of random maths = urlfetchapp.fetch (""); Recorder.log (responsive) It's nice and easy! Step 12: Add the code for a custom menu Add the following code into your script editor: Function ONOPEN () {var ui = SpreadSheetApp.getui (); ui.createmenu ('Menu') Custom Numbers APIs' .Additem ('Show the fact of the random number', 'callnumbers match' .Adtoui Step 13: Add the custom menu Run the ONOPEN function, which will add the menu to the spreadsheet. We just need to take this step once. Step 14: Run your script in the custom menu Use the new menu to run your script from the Google sheet and see the random facts popup window on your Google sheet! All right, ready to try something a little harder? We're gonna build a music discovery app on Google Sheets. Example 2: Music discovery application Using iTunes API This

[illegible]