

Fogli Di Calcolo 5

Prof. Lorian Storch

loriano@storch.org | <https://www.storch.org/>

Parte 1

Macro

Automatizzare una serie di operazioni

Cos'è una Macro



Funzione Utente

In pratica una funzione definita dall'utente che riceve come parametri i valori contenuti nelle celle.



Il Linguaggio

Google Sheets usa **JavaScript** come linguaggio per la definizione delle macro.



Automazione

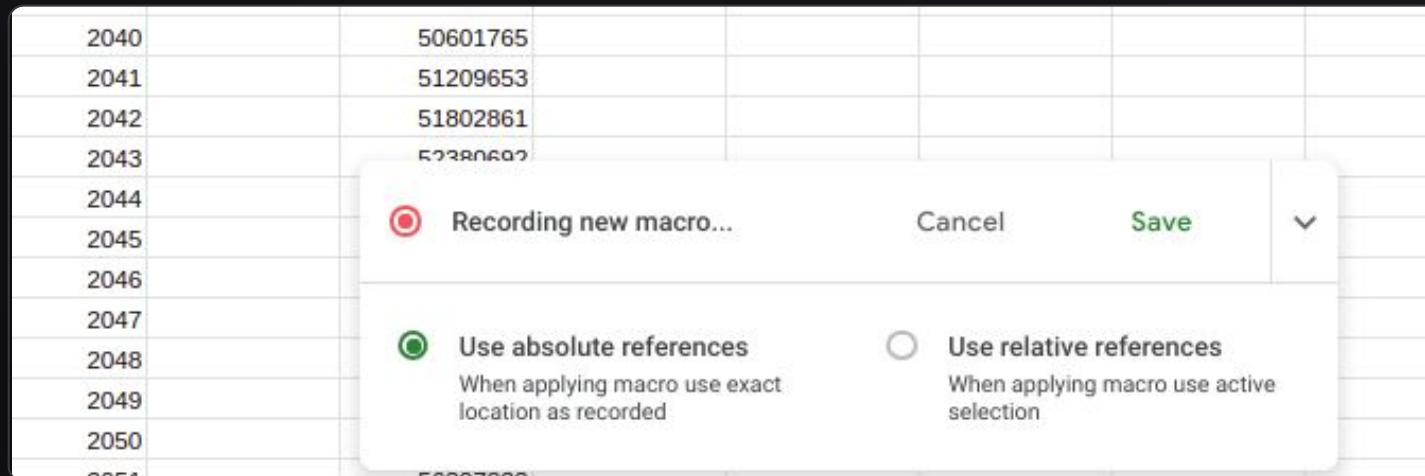
Possiamo usare una macro, ad esempio, per automatizzare una serie di operazioni ripetitive.

Registrazione una Macro

Avviare la Registrazione

Importiamo il file **world-population-1750-2015-and-un-projection-until-2100.csv** e poi avviamo la registrazione dal menu:

Tools → Macros → Record macro



Le Operazioni da Registrare

1. Testo a Capoverso

Selezionare la prima riga e applicare:

Format → Text wrapping → Wrap

2. Sfondovero Rosso

Selezionare le celle da A1 a E1 e colorare lo sfondo di rosso.

3. Colori Alternati

Selezionare le colonne dalla A alla E e poi:

Format → Alternating colors

4. Salvare la Macro

Premere **Save** e chiamare la macro **formatter**.

Un Nuovo Foglio di Dati

Nuovo Sheet

Aggiungiamo un nuovo sheet e importiamo
il file **world-population-subset.csv**.

A	B	C	D	E	F	G	H
Entity	Code	Year	World Population Medium Projection (UN Population Division (2015 revision)) (people)				
Italy	ITA	2015		59797685			
Italy	ITA	2016		59801004			
Italy	ITA	2017		59797978			
Italy	ITA	2018		59788104			
Italy	ITA	2019		59769595			
Italy	ITA	2020		59741327			
Italy	ITA	2021		59704031			
Italy	ITA	2022		59659281			
Italy	ITA	2023		59607682			
Italy	ITA	2024		59549862			
Italy	ITA	2025		59486401			
Italy	ITA	2026		59417644			
Italy	ITA	2027		59343986			
Italy	ITA	2028		59266082			
Italy	ITA	2029		59184651			
Italy	ITA	2030		59100219			
Italy	ITA	2031		59013045			
Italy	ITA	2032		58923116			
Italy	ITA	2033		58830307			
Italy	ITA	2034		58734335			
Italy	ITA	2035		58634915			
Italy	ITA	2036		58531995			
Italy	ITA	2037		58425413			
Italy	ITA	2038		58314606			
Italy	ITA	2039		58198854			
Italy	ITA	2040		58077567			

Eseguire la Macro

Applicare formatter

Proviamo ora a eseguire la nostra macro sul nuovo foglio, semplicemente con:

Tools → Macros → formatter

Il Risultato

La prima riga va a capo, l'intestazione diventa rossa e le colonne da A a E ricevono i colori alternati.

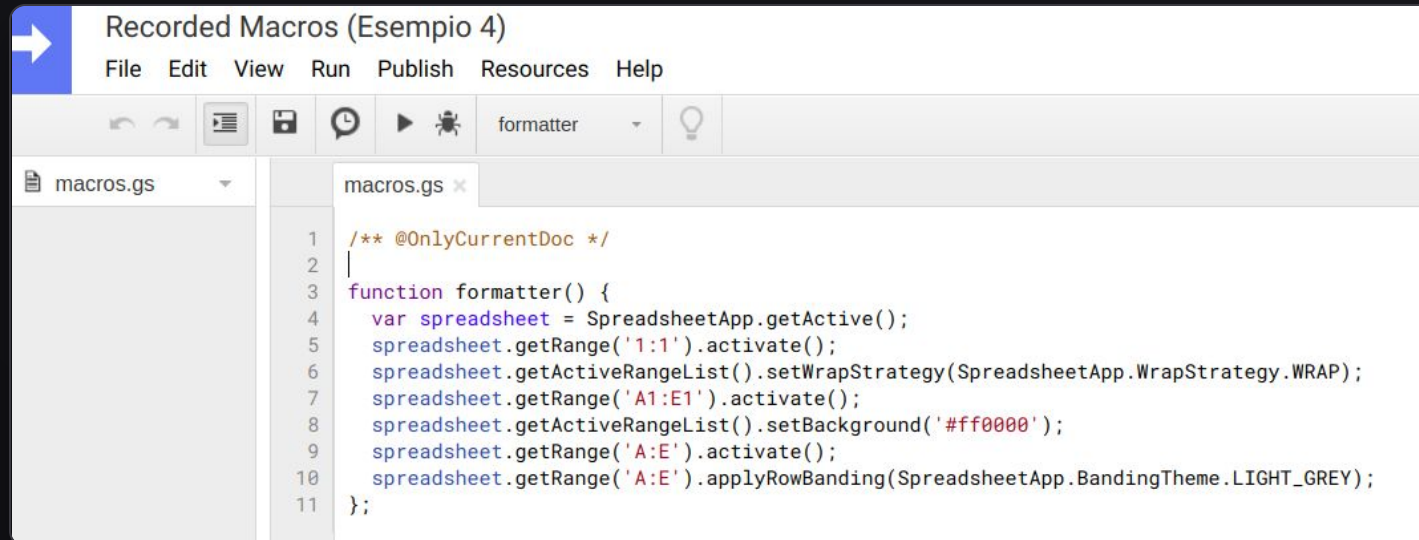
	A	B	C	D	E	F
	Entity	Code	Year	World Population over 12000 years (various sources (2016)) (people)	Medium Projection (UN Population Division (2015 revision)) (people)	
	Italy	ITA	2015		59797685	
	Italy	ITA	2016		59801004	
	Italy	ITA	2017		59797978	
	Italy	ITA	2018		59788104	
	Italy	ITA	2019		59769595	
	Italy	ITA	2020		59741327	
	Italy	ITA	2021		59704031	
	Italy	ITA	2022		59659281	
0	Italy	ITA	2023		59607682	
1	Italy	ITA	2024		59549862	
2	Italy	ITA	2025		59486401	
3	Italy	ITA	2026		59417644	
4	Italy	ITA	2027		59343986	
5	Italy	ITA	2028		59266082	
6	Italy	ITA	2029		59184651	
7	Italy	ITA	2030		59100219	
8	Italy	ITA	2031		59013045	
9	Italy	ITA	2032		58923116	
0	Italy	ITA	2033		58830307	

Il Codice della Macro

Script Editor

Aprendo l'editor vediamo il codice JavaScript generato dalla registrazione della macro:

Tools → Script editor



```
Recorded Macros (Esempio 4)
File Edit View Run Publish Resources Help

macros.gs
macros.gs x

1  /** @OnlyCurrentDoc */
2  |
3  function formatter() {
4    var spreadsheet = SpreadsheetApp.getActive();
5    spreadsheet.getRange('1:1').activate();
6    spreadsheet.getActiveRangeList().setWrapStrategy(SpreadsheetApp.WrapStrategy.WRAP);
7    spreadsheet.getRange('A1:E1').activate();
8    spreadsheet.getActiveRangeList().setBackground('#ff0000');
9    spreadsheet.getRange('A:E').activate();
10   spreadsheet.getRange('A:E').applyRowBanding(SpreadsheetApp.BandingTheme.LIGHT_GREY);
11  };
```

Parte 2

Scrivere una Funzione Custom

Definire nuove funzioni in JavaScript

Preparare i Dati

Nuovo Sheet

Creiamo un nuovo sheet e inseriamo alcuni valori numerici.

	A	B
1	1	-3
2	4	2
3	4	-6

La Funzione ADOUBLE

Nuovo Script

Apriamo l'editor e creiamo un nuovo file di script:

Tools → Script editor

File → New → Script file

Come Funziona

Se l'input è un intervallo (array), la funzione si richiama su ogni elemento con **map**. Altrimenti restituisce il valore raddoppiato se positivo, oppure moltiplicato per -1 .

```
macros.gs x * adouble.gs x
1 function ADOUBLE(input)
2 {
3   if (input.map) // se array
4   {
5     return input.map(ADOUBLE); // richiama la funzione su ogni elemento
6   }
7   else
8   {
9     if (input > 0)
10    {
11      return input * 2;
12    }
13    else
14    {
15      return input * -1;
16    }
17  }
18 }
```

Usare la Funzione sui Dati

Una Cella o un Intervallo

La funzione si usa come qualsiasi funzione predefinita, su una singola cella o su un intervallo:

D2: =ADOUBLE(A1)

D3: =ADOUBLE(A2:B3)

1	-3	2	
4	2	=ADOUBLE(A2:B3)	4
4	-6	8	6

Parte 3

Oltre i Fogli di Calcolo

Un paio di esempi pratici in Python

Un Paio di Esempi in Python



Media e Deviazione Standard

Calcoliamo media e deviazione standard leggendo i dati da un file di testo oppure con la libreria pandas.



Repository Git

Il codice si trova nella cartella **basictests** del repository:

<https://github.com/lstorchi/teaching>

Da un File di Testo

Lettura Riga per Riga

Lo script legge il file passato come argomento, estrae il valore di ogni riga e lo aggiunge a una lista.

Le Statistiche

Alla fine stampa media e deviazione standard con **numpy.mean** e **numpy.std**.

```
import numpy
import sys
import re

filename = ""
if len(sys.argv) == 2:
    filename = sys.argv[1]
else:
    print "usage: ", sys.argv[0], " filename.txt"
    exit(1)

values = []
fp = open(filename, "r")
for line in fp:
    p = re.compile(r'\s+')
    line = p.sub(' ', line)
    line = line.lstrip()
    line = line.rstrip()

    sline = line.split(' ')

    if len(sline) == 1:
        values.append(float(sline[0]))

print numpy.mean(values), " ", numpy.std(values)

fp.close()
```

Con la Libreria pandas

Lettura con pandas

I dati vengono letti con **pandas.read_excel** e la prima colonna viene convertita in un array numpy.

Le Statistiche

Come prima, media e deviazione standard si ottengono con **numpy.mean** e **numpy.std**.

```
import pandas
import numpy
import sys
import re

filename = ""
if len(sys.argv) == 2:
    filename = sys.argv[1]
else:
    print "usage: ", sys.argv[0], " filename.txt"
    exit(1)

df = pandas.read_excel(filename)
cn = df.columns

print "CN: ", cn

values = numpy.asarray(df[cn[0]].values)
print numpy.mean(values), " ", numpy.std(values)
```