

Capitolo 44: Script di Office per Excel

1. Introduzione agli Office Scripts per Excel

a. Introduzione agli Office Scripts

Gli Office Scripts sono script automatizzati scritti in TypeScript, pensati per automatizzare attività in Excel online. Funzionano solo in Excel per il web (Microsoft 365) e offrono un'alternativa moderna alle macro VBA, con il vantaggio dell'integrazione nel cloud.

b. Esplorazione della scheda Automatizza (Automate)

La scheda 'Automatizza' è disponibile in Excel Online con un abbonamento a Microsoft 365. Include strumenti per registrare script, modificarli con l'editor di codice e automatizzarli tramite Power Automate.

Contiene:

- Registra azioni (Registratore di script)
- Nuovo script
- Editor di script
- Avvia script
- Integra con Power Automate

Esempi pratici

Script per scrivere 'Ciao mondo' in A1

```
function main(workbook: ExcelScript.Workbook) {
  workbook.getActiveWorksheet().getRange("A1").setValue("Ciao mondo");
}
```

Script per sommare due celle

```
function main(workbook: ExcelScript.Workbook) {
  let ws = workbook.getActiveWorksheet();
  let somma = ws.getRange("A1").getValue() + ws.getRange("A2").getValue();
  ws.getRange("A3").setValue(somma);
}
```

Script per applicare colore rosso a un intervallo

```
function main(workbook: ExcelScript.Workbook) {
  workbook.getActiveWorksheet().getRange("B2:D4").getFormat().getFill().setColor("#FF0000");
}
```

Script per creare un nuovo foglio

```
function main(workbook: ExcelScript.Workbook) {
  workbook.addWorksheet("NuovoFoglio");
}
```

Script per eliminare righe vuote

```
function main(workbook: ExcelScript.Workbook) {
  let ws = workbook.getActiveWorksheet();
  let usedRange = ws.getUsedRange();
  usedRange.getRows().forEach(row => {
    if (row.getTexts().every(cell => cell === "")) row.delete(ExcelScript.DeleteShiftDirection.up);
  });
}
```

Esercizi

Crea uno script che scriva la data odierna nella cella B1

Utilizza `new Date().toLocaleDateString()` e `setValue()`

Crea uno script che formatti in grassetto l'intervallo A1:C1

Usa `getFormat().getFont().setBold(true)`

Crea uno script che ordini i valori in colonna A

Usa `getRange("A1:A10").getSort()`

Crea uno script che aggiunga 10 fogli con nomi diversi

Loop for con `workbook.addWorksheet("Foglio" + i)`

Crea uno script che cerchi e sostituisca il testo 'Vecchio' con 'Nuovo'

Usa `getRange().getText()` e `setValue()` con `replace()`

2. Differenze tra Office Scripts e VBA

a. Sicurezza

Office Scripts opera nel cloud Microsoft 365, garantendo ambienti isolati e autenticazione sicura. Non può accedere a risorse locali come file system o altri programmi, contrariamente al VBA.

b. Dove sono memorizzati gli script

Gli script sono archiviati nel cloud Microsoft 365, associati all'account dell'utente. Si accede e si gestiscono dalla scheda 'Automatizza' > 'Script personali'.

c. Cosa controllano gli script

Gli Office Scripts possono manipolare solo i contenuti e la struttura del file Excel in uso. Non possono interagire con altri file o con l'ambiente del sistema operativo.

d. Linguaggio e parole chiave

Office Scripts usa TypeScript (una variante di JavaScript) con l'API di Excel. VBA usa Visual Basic for Applications, con un set di comandi e sintassi diverso.

Esempio Office Script:

```
function main(workbook: ExcelScript.Workbook) { ... }
```

Esempio VBA:

```
Sub NomeMacro()  
    ' codice  
End Sub
```

e. Tipi di procedure

Office Scripts ha una sola procedura chiamata 'main', mentre VBA consente Sub, Function e gestione eventi. Office Script è orientato a compiti batch, mentre VBA consente interazioni dinamiche con l'utente.

f. Come eseguire gli script

Office Scripts si eseguono da Excel Online tramite la scheda 'Automatizza'. Possono anche essere integrati in Power Automate per esecuzioni pianificate o basate su eventi.

Esempi pratici

Office Script per inserire testo in A1

```
function main(workbook: ExcelScript.Workbook) {  
    workbook.getActiveWorksheet().getRange("A1").setValue("Testo");  
}
```

VBA per inserire testo in A1

```
Sub InserisciTesto()  
    Range("A1").Value = "Testo"  
End Sub
```

Office Script per sommare due celle

```
let somma = ws.getRange("A1").getValue() + ws.getRange("A2").getValue();
```

VBA per creare un messaggio

```
MsgBox "Operazione completata"
```

Office Script per creare un nuovo foglio

```
workbook.addWorksheet("Nuovo")
```

Esercizi

Scrivi uno script e una macro che calcolano il quadrato di un numero

Confronta l'implementazione tra Office Script e VBA

Crea uno script che cambia il colore di sfondo in B2

Usa `.getFormat().getFill().setColor(...)`

Crea una macro VBA che ordina i dati in colonna A

Usa `Range.Sort`

Esegui uno script tramite Power Automate

Crea un flusso che richiama un Office Script da Excel

Scrivi una funzione VBA che restituisce il massimo di tre numeri

```
Function MaxTre(a,b,c): MaxTre = WorksheetFunction.Max(...)
```

3. Esplorare l'ambiente Office Scripts in Excel

a. Avviare un nuovo script

Dalla scheda 'Automatizza' in Excel per il web, clicca su 'Nuovo script'. Si aprirà l'editor con un modello base:

```
function main(workbook: ExcelScript.Workbook) {  
  // codice qui  
}
```

b. Aprire uno script esistente

Sempre dalla scheda 'Automatizza', seleziona 'Script personali'. Da qui puoi aprire, modificare, duplicare o eliminare i tuoi script salvati nel cloud Microsoft 365.

c. Esaminare il menu Altre opzioni (More Options)

Cliccando sui tre puntini accanto a uno script, puoi:

- Rinominare
- Eliminare
- Duplicare
- Condividere tramite link
- Eseguire direttamente lo script

d. Utilizzare la console

La console integrata nell'editor Office Scripts consente di visualizzare output e log. Utilizza:

```
console.log("Messaggio di debug");
```

per testare il comportamento del tuo script. È utile per eseguire il debug e visualizzare variabili.

Esempi pratici

Script per impostare il valore 'Ciao' in A1

```
function main(workbook: ExcelScript.Workbook) {  
  workbook.getActiveWorksheet().getRange("A1").setValue("Ciao");  
}
```

Stampare un messaggio nella console

```
console.log("Esecuzione completata");
```

Creare un nuovo foglio e scrivere il nome del foglio

```
let ws = workbook.addWorksheet("Nuovo");  
console.log(ws.getName());
```

Leggere il valore di una cella e mostrarlo in console

```
let valore = workbook.getActiveWorksheet().getRange("B1").getValue();  
console.log(valore);
```

Formattare un intervallo e verificarlo nella console

```
let rng = workbook.getActiveWorksheet().getRange("A1:B2");  
rng.getFormat().getFont().setBold(true);  
console.log("Grassetto applicato");
```

Esercizi

Crea uno script che scriva 'Esercizio' in A1 e lo stampi in console

Utilizza setValue e console.log

Scrivi uno script che crei un nuovo foglio e lo rinomini

Usa addWorksheet e setName

Modifica uno script esistente e testalo

Apri dalla sezione 'Script personali' e modifica il contenuto

Duplica uno script e applica una modifica visibile

Modifica i valori scritti o il range target

Crea uno script che evidenzi A1:C1 e stampi 'Fatto'

Formatta con colore e stampa un log

4. Creazione e scrittura di Office Scripts in Excel Online

a. Utilizzare il registratore di azioni (Action Recorder)

Il registratore consente di automatizzare operazioni in Excel registrando le azioni effettuate manualmente. Si accede dalla scheda 'Automatizza' > 'Registra azioni'. Dopo l'interruzione, viene generato automaticamente uno script modificabile.

b. Registrare le azioni per creare Office Scripts

Avvia la registrazione, esegui operazioni (es. inserimento dati, formattazione), quindi arresta la registrazione. Lo script viene generato automaticamente con codice leggibile e modificabile.

c. Registrare uno script: esempio semplice

Esempio: Inserisci 'Report' in A1 e formatta in grassetto.

Lo script risultante sarà simile a:

```
function main(workbook: ExcelScript.Workbook) {  
  let ws = workbook.getActiveWorksheet();  
  ws.getRange("A1").setValue("Report");  
  ws.getRange("A1").getFormat().getFont().setBold(true);  
}
```

d. Esaminare e testare lo script

Dopo la registrazione, clicca su 'Modifica' per visualizzare e comprendere il codice generato. Premi 'Esegui' per testare lo script sul foglio di lavoro attivo.

e. Modificare e salvare lo script

Modifica direttamente il codice TypeScript. Premi 'Salva' e assegna un nome descrittivo. Lo script sarà disponibile tra gli script personali.

f. Scrivere Office Scripts manualmente

Gli script possono essere scritti manualmente usando il pacchetto ExcelScript, che fornisce oggetti e metodi per manipolare il foglio.

g. Il pacchetto ExcelScript

Offre classi come Workbook, Worksheet, Range, Table, ecc. Accessibili tramite workbook.getActiveWorksheet(), ecc.

h. Oggetti e metodi

Esempi:

- getRange()
- setValue()
- getFormat()
- addWorksheet()

Ogni metodo interagisce con un oggetto Excel. La documentazione ufficiale Microsoft fornisce dettagli su tutti i metodi disponibili.

i. Variabili

Si usano let o const:

```
let valore = 10;  
const nomeFoglio = workbook.addWorksheet("Foglio1");
```

j. Oggetti più utilizzati

- Workbook
- Worksheet
- Range
- Table
- Chart

Questi permettono di navigare, leggere e modificare dati nei fogli Excel.

k. Controllare il flusso di esecuzione

Si usano costrutti classici:

- if/else
- for
- while

Esempio:

```
for (let i = 1; i <= 10; i++) {  
  ws.getRange("A" + i).setValue(i);  
}
```

Esempi pratici

Scrivere un testo in una cella

```
workbook.getActiveWorksheet().getRange("B2").setValue("Testo inserito");
```

Applicare grassetto a un intervallo

```
workbook.getActiveWorksheet().getRange("A1:C1").getFormat().getFont().setBold(true);
```

Creare un nuovo foglio

```
workbook.addWorksheet("NuovoFoglio");
```

Riempire un intervallo con numeri da 1 a 10

```
for (let i = 0; i < 10; i++) {  
  sheet.getRange("A" + (i+1)).setValue(i+1);  
}
```

Usare console.log per mostrare un valore

```
console.log(workbook.getActiveWorksheet().getRange("A1").getValue());
```

Esercizi

Crea uno script che scriva 'Fattura' in A1 e lo formatti in grassetto

Combina setValue e setBold

Scrivi uno script che sommi A1 e A2 e metta il risultato in A3

Usa getValue e setValue

Registra un'azione in cui cambi il colore di sfondo di un intervallo

Utilizza il registratore e modifica il codice

Crea un ciclo che popola B1:B5 con i numeri da 5 a 1

Usa un ciclo for decrescente

Scrivi uno script che inserisca la data odierna in C1

Usa new Date().toLocaleDateString()