CHEAT SHEET · PART 1

Apps Script for GTM

Five scripts a product marketer can run this week. JavaScript that runs inside Google Workspace: Sheets, Slides, Docs, Gmail, Drive, Calendar, Forms.

WHERE IT LIVES

Bound vs standalone

- **Bound:** attached to one file. Extensions › Apps Script. Can add menus and react to edits. Start here.
- **Standalone:** script.google.com. For jobs across many files.

RUN IT

Save, run, allow, reload

1. Open a sheet. **Extensions › Apps Script**.
2. Paste the code. **Save**.
3. Pick a function in the toolbar dropdown. **Run**.
4. First run: read the permission screen, then allow.
5. Reload the sheet. Your menu appears next to Help.

FIRST SCRIPT

A GTM menu in your sheet

- `onOpen` runs every time the sheet opens.
- Each item calls a function by name.
- Reload the sheet to see the menu.

```
function onOpen() {
  SpreadsheetApp.getUi()
    .createMenu('GTM')
    .addItem('Build account decks', 'buildDecks')
    .addItem('Draft follow-ups', 'draftFollowUps')
    .addToUi();
}
```

SUPERPOWER 1

A formula that checks copy length

- Use in a cell: `=FITS(D2, E2)`. Catch overflow before it hits a slide.

```
/**
 * Says whether a line fits its character budget.
 * @customfunction
 */
function FITS(text, budget) {
  const n = String(text).length;
  return n <= budget ? 'OK' : 'Over by ' + (n - budget);
}
```

DEBUG

Test small, read the log

- Run a tiny test function from the editor.
- `console.log` output shows in the execution log.
- View › Executions lists every run.

BEFORE YOU SHARE IT

Safety checklist

- Read the permission screen. It lists exactly what the script can touch.
- Scripts run as whoever runs them, with that person's access.
- Draft before you send. Copy before you overwrite.
- Test on copies of real files.
- Quotas apply: one run stops at 6 minutes, and email has daily limits.

RECAP

Five superpowers

`FITS()`	Checks copy against a character budget, in a cell.
`buildDecks()`	One personalised deck per row, from a template.
`draftFollowUps()`	Gmail drafts from a sheet. A person sends.
`weeklyDigest()`	A Monday email, on a time trigger.
`onEdit()`	A timestamp on every messaging change.

SUPERPOWER 2

One deck per account, from a template

- Template slides hold tokens like `{{Company}}`.
- Sheet headers match the tokens.
- One run copies the template per row and fills it in.

```
const TEMPLATE_ID = 'paste-your-template-deck-id';

function buildDecks() {
  const [head, ...rows] = SpreadsheetApp.getActive()
    .getSheetByName('Accounts').getDataRange().getValues();
  rows.forEach(row => {
    const copy = DriveApp.getFileById(TEMPLATE_ID)
      .makeCopy('Pitch - ' + row[0]);
    const deck = SlidesApp.openById(copy.getId());
    head.forEach((key, i) =>
      deck.replaceAllText('{{' + key + '}}', String(row[i])));
  });
}
```

SUPERPOWER 4

Run it every Monday

- Run `installWeekly` once. It schedules the digest.
- The digest emails you proof points close to expiry.
- Triggers run as you, sheet closed or not.

```
function installWeekly() {
  ScriptApp.newTrigger('weeklyDigest').timeBased()
    .onWeekDay(ScriptApp.WeekDay.MONDAY).atHour(9).create();
}

// Proof tab: column G holds days left.
function weeklyDigest() {
  const rows = SpreadsheetApp.getActive()
    .getSheetByName('Proof').getDataRange().getValues();
  const soon = rows.filter(r => r[6] >= 0 && r[6] < 30);
  if (!soon.length) return;
  MailApp.sendEmail(Session.getEffectiveUser().getEmail(),
    soon.length + ' proof points expire within 30 days',
    soon.map(r => r[1] + ': ' + r[0]).join('\n'));
}
```

SUPERPOWER 3

Follow-ups as drafts, not sends

- Rows marked Ready become Gmail drafts.
- Nothing is sent. A person reads each one.

```
// Columns: Name, Email, Company, Status
function draftFollowUps() {
  const rows = SpreadsheetApp.getActive()
    .getSheetByName('Follow-ups').getDataRange().getValues();
  rows.slice(1).filter(r => r[3] === 'Ready').forEach(r => {
    const body = 'Hi ' + r[0] + ',\n\nThanks for the time today.';
    GmailApp.createDraft(r[1], 'Next steps for ' + r[2], body);
  });
}
```

SUPERPOWER 5

Stamp every messaging change

- `onEdit` is a simple trigger. No setup.
- Sort by column I to see what changed.

```
// Stamps column I when a row on Messages changes.
function onEdit(e) {
  const sheet = e.range.getSheet();
  if (sheet.getName() !== 'Messages') return;
  if (e.range.getRow() === 1) return;
  sheet.getRange(e.range.getRow(), 9).setValue(new Date());
}
```

COMING IN PART 2

Talk to everything else

- UrlFetchApp: pull CRM and product data into Sheets.
- Post launch updates to a Google Chat space.
- Call Gemini from a script to draft copy to a budget.
- Ship a tiny internal web app with doGet().

Reference: developers.google.com/apps-script · Quotas: developers.google.com/apps-script/guides/services/quotas · 2 / 2