



# From Scripts Newbie to PMax Pro

A Hands-On Journey to Better Performance

Mike Rhodes – [8020agent.com](https://8020agent.com)





channable

Next Level eCommerce | Webinar series

# Level up your PPC skills

Guided by the best in the industry!

- Hands-on tips
- Ready-to-use frameworks
- Step-by-step scripts

Bob Meijer



Andrew Lolk



Ruben Runneboom



Mike Rodes



Aaron Young



Nils Rooijmans





# Mike Rhodes

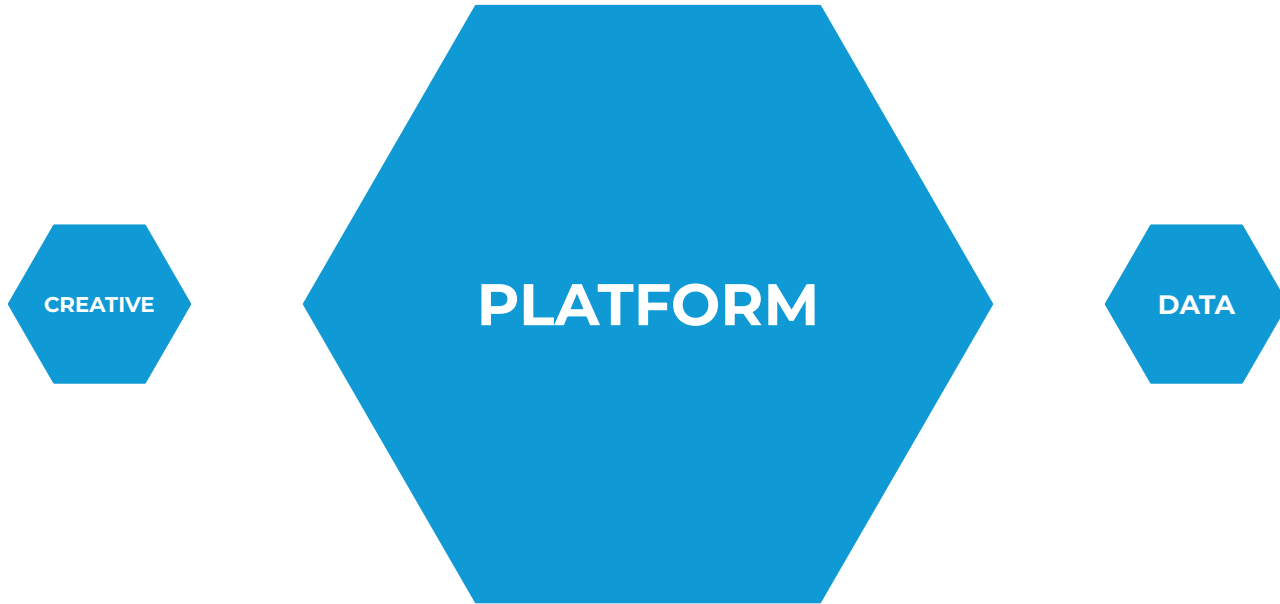
- **Serial entrepreneur** who has built and sold four companies, including **WebSavvy**.
- WebSavvy was named **by Google as one of their Top 18 global agencies**.
- Has spoken at **over 100 international events**. Co-authored the world's **best-selling book on Google Ads** with Perry Marshall.
- Currently, building **tools with AI & Agents** and **enjoys family travels** and **mountain biking**.

*Next Level eCommerce | Webinar series*

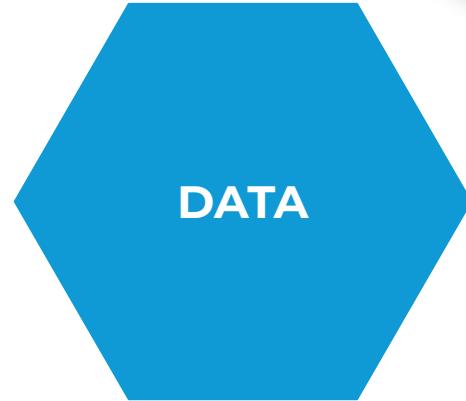
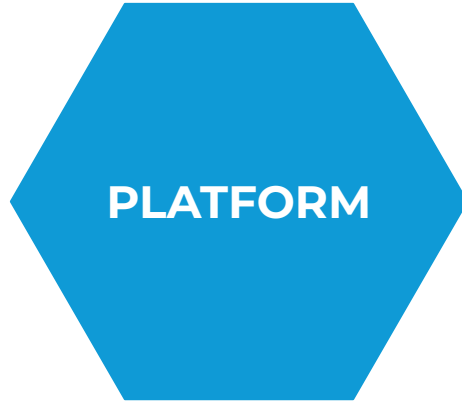
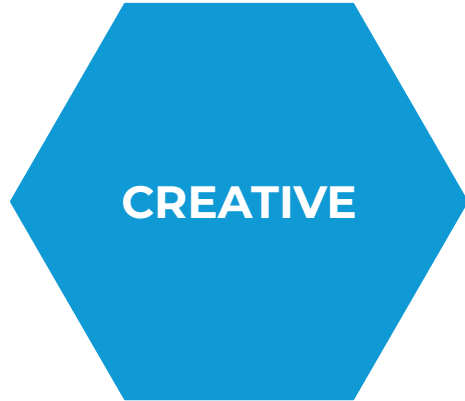
# **From scripts newbie to PMax Pro**

A hands-on journey to better performance

# It used to be enough...



# ...but that's changed



# Ideally you'll know at least 2

## CREATIVE

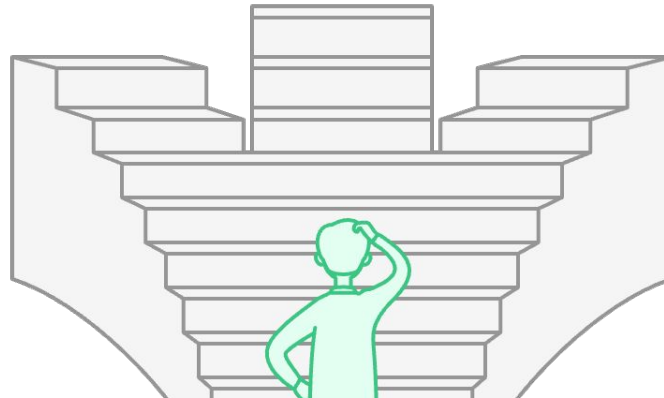
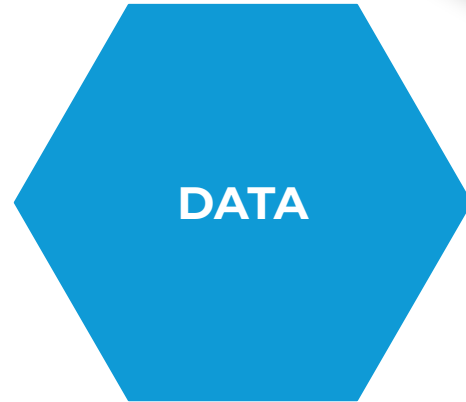
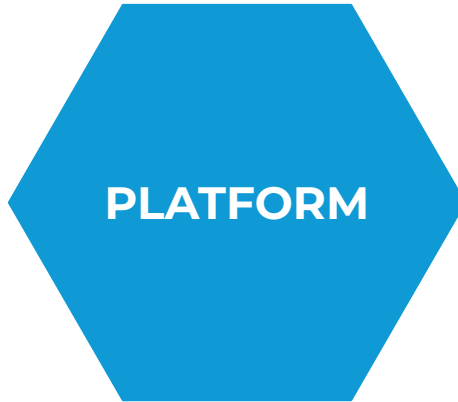
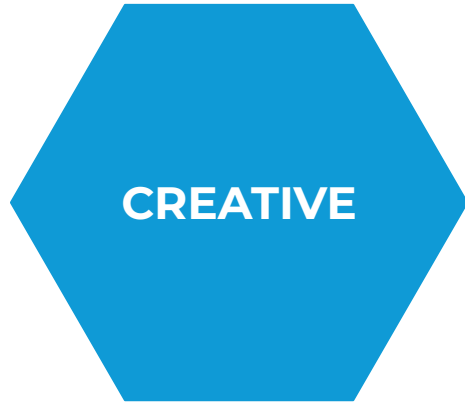
- Ad copy
- Image ads
- Video scripts
- Landing page
- Infographics
- Email copy
- Pitch decks
- Offers

## PLATFORM

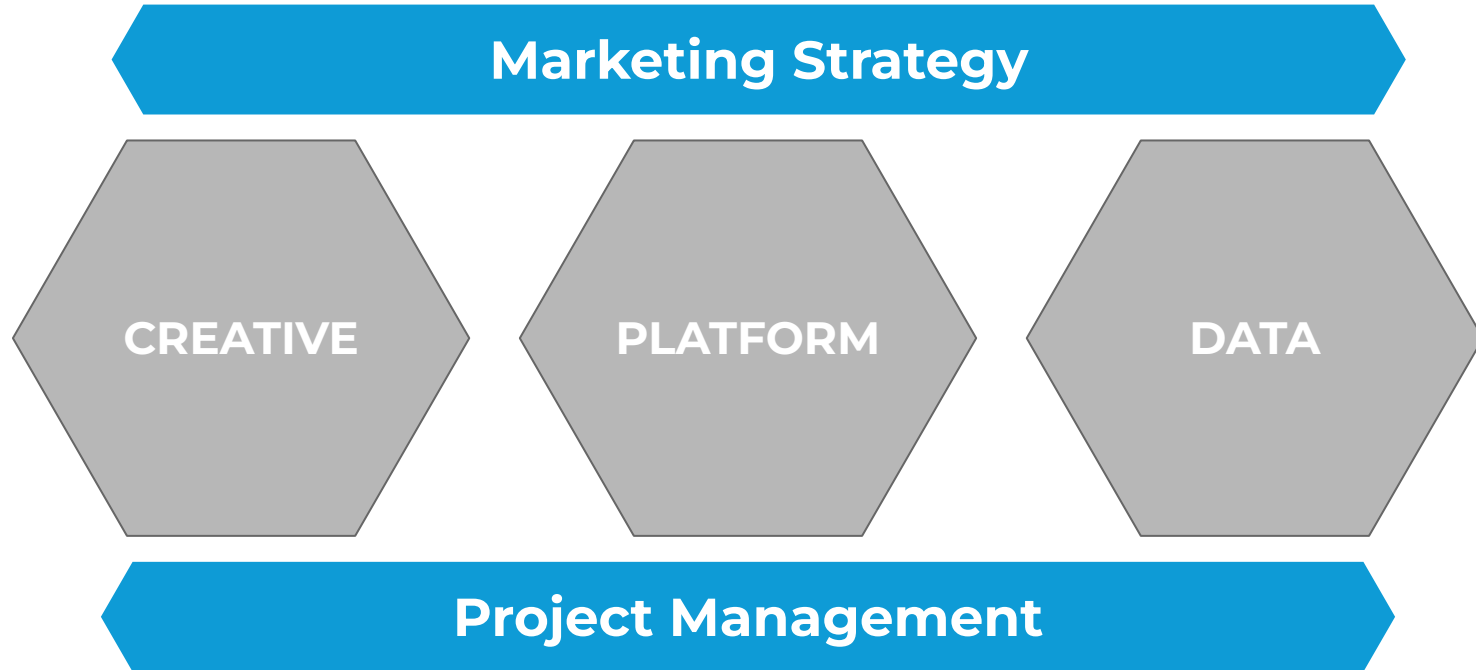
## DATA

- Scripts
- Analytics
- Attribution
- Automation
- Conv tracking
- Tag manager
- Lead scoring
- A/B testing

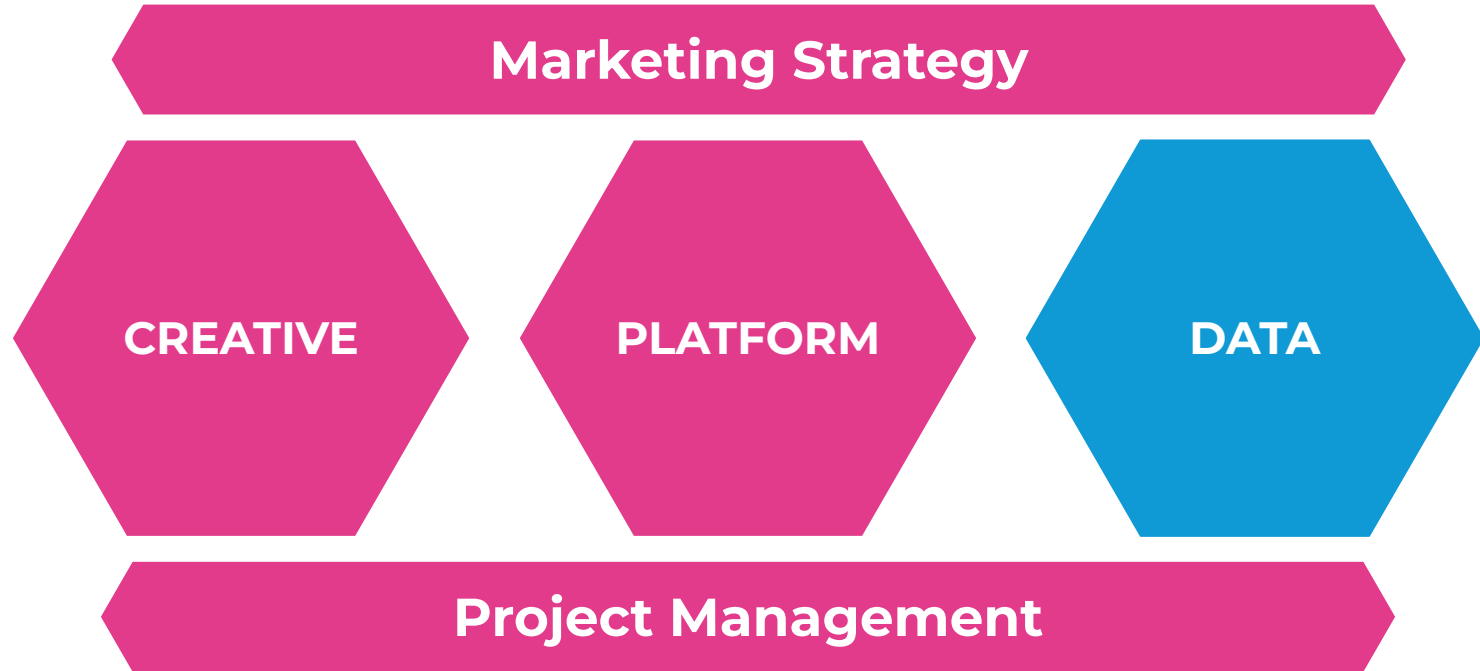
# What's Your Future?



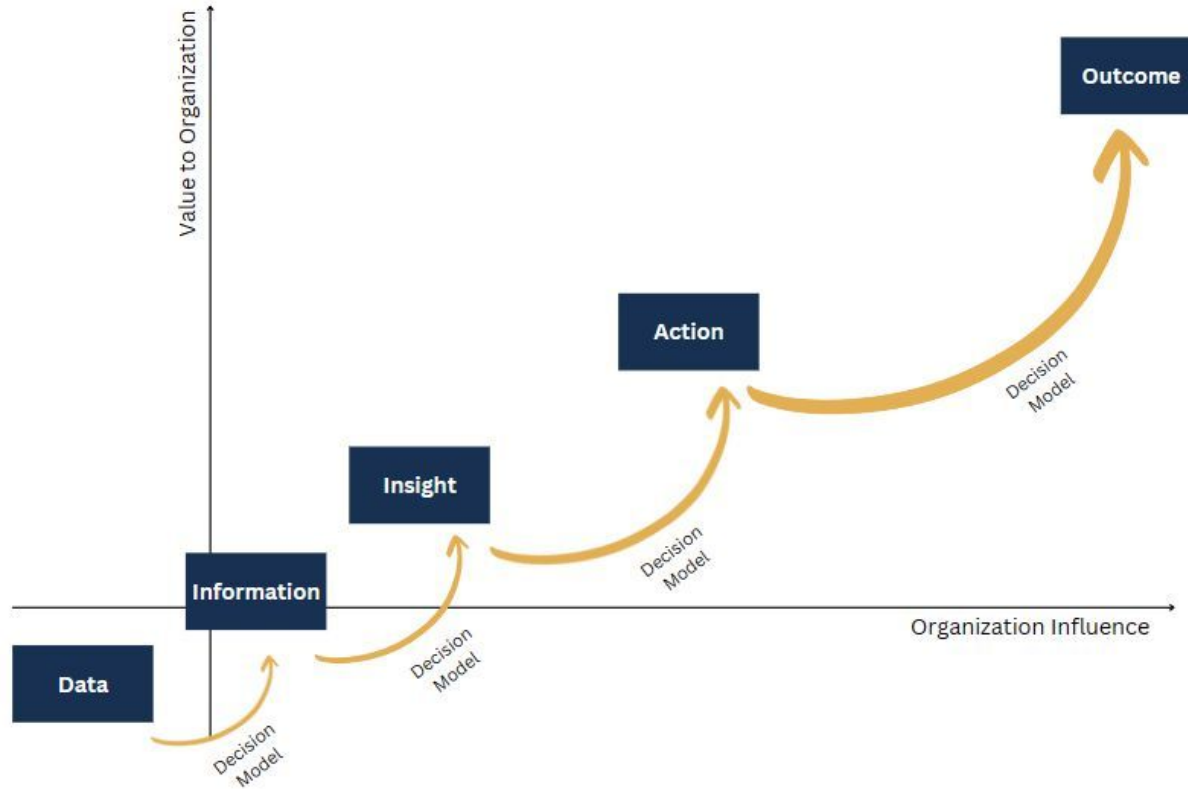
# There are 2 other options...



# Data is a Critical Foundation



# Real Business Value



# The 4 Levels of Scripts



The 4 Cs

# Check - Monitor & Alert

- Daily alerts
- Anomaly detection
- Budget pacing checks
- Check for disapproved ads
- Monitor landing page status. 404 checks
- Alert for conversion tracking issues
- Change history alerts

# Chart - Turn data into information

- **Automate chart creation for insights**
- **Data Export** (Sheets/Slides/Data Studio etc)
- **Generate custom reports/dashboards**
- **Get data out of Platform = more control**

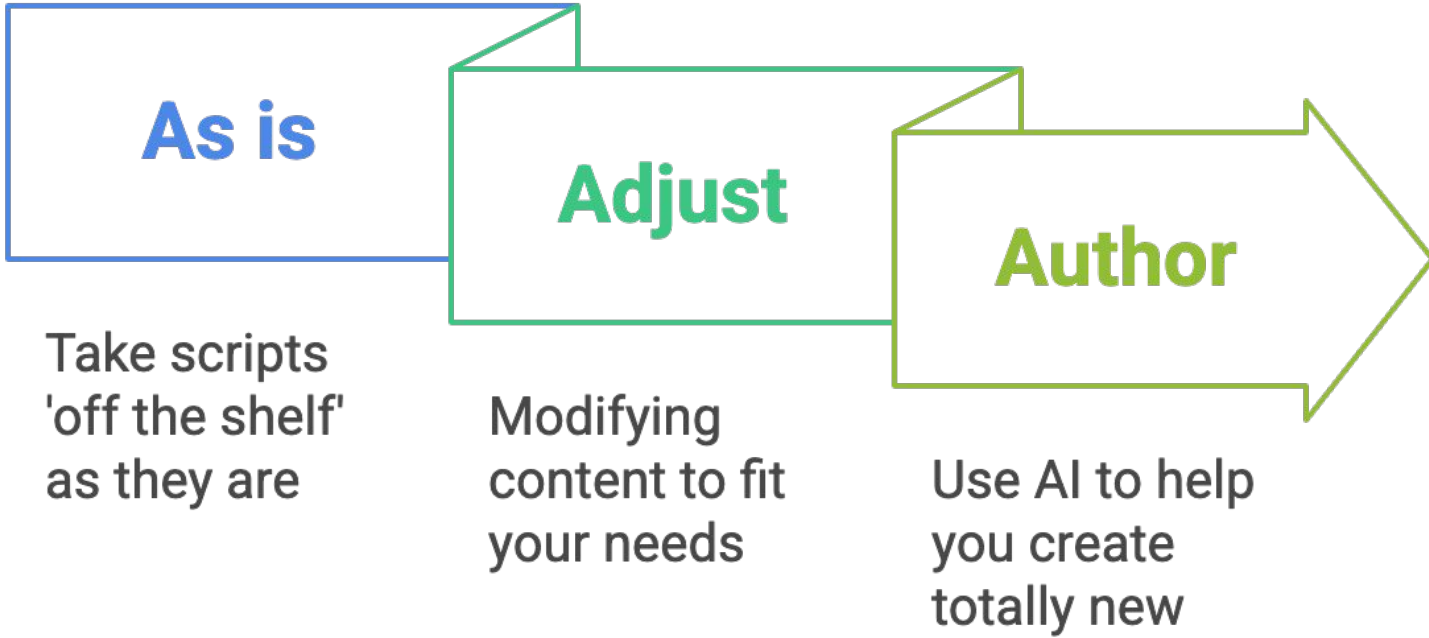
# Change - Rule based changes

- Make changes based on rules
- Bid adjustments based on performance
- Budget allocation optimization
- Weather-based bid adjustments
- Keyword expansion from search terms
- Build campaign from template sheet

# Cognition - **Scripts that 'think'!**

- **Classify Search term intent**
- **Score Landing page relevance**
- **Ad copy effectiveness & compliance**
- **Audience segment discovery**
- **Competitor analysis**
- **Advanced Budget forecasts**
- **Recommendations based on your guidelines**

# There are 3 Ways to Create Scripts



# Today we'll dig into these 3

	As is	Adjust	Author
Check (alerts)			
Chart (info)			
Change (action)			
Cognition (think)			

# A (very) quick tour

Less than 5% of Google Ads users  
have ever used a script



Create

Planning

Shared Library

Content suitability

Campaigns

Data manager



Goals

Troubleshooting



Tools

Bulk actions

All bulk actions

Rules

Scripts

Solutions

Uploads



Billing

Budgets and bidding



Admin

Business data

# Scripts

Custom 13 Dec 2024 - 9 Jan 2025

Scripts Script history

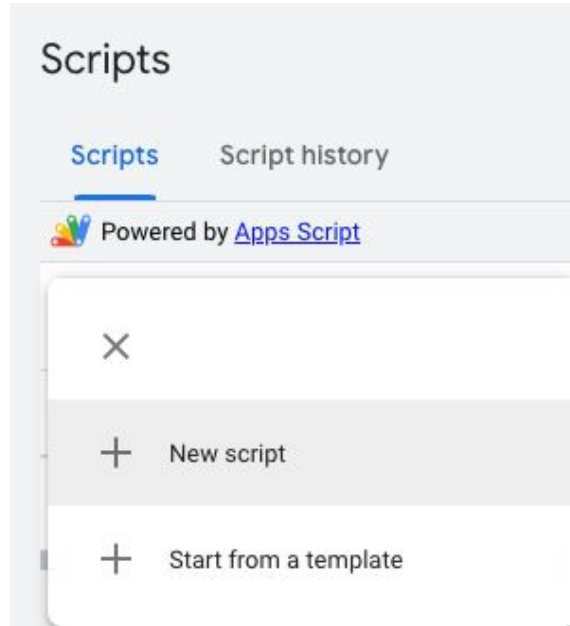
Powered by Apps Script

+ Add filter

Script	Created by	Frequency	Last edit date/time	Last run/previ ↓	Results	Status
--------	------------	-----------	---------------------	------------------	---------	--------

You don't have any scripts yet

Click 



## New script

Script name:



Powered by [Apps Script](#)

[Advanced APIs](#)

[Documentation](#)



Expand



Scripts make changes on a user's behalf. You must authorize scripts before they can make any changes. [Learn more about authorisation](#)

[Authorise](#)

Info.gs

Config.gs

**Code.gs**

```
1 /**
2  * Performs custom validation on a URL, with access to details such as the URL,
3  * the response from the server, configuration options and entity Details.
4  *
5  * To use, the "Use Custom Validation" option in the configuration spreadsheet
6  * must be set to "Yes", and your custom validation code implemented within the
7  * below function.
8  *
9  * See the documentation for this solution for further details.
10 *
11 * @param {string} url The URL being checked.
12 * @param {!Object} response The response object for the request.
13 * @param {!Object} options Configuration options.
14 * @param {!Object} entityDetails Details of the associated Ad / Keywords etc.
15 * @return {boolean} Return true if the URL and response are deemed valid.
16 */
17 function isValidResponse(url, response, options, entityDetails) {
18   /*
19    * Some examples of data that can be used in determining the validity of this
20    * URL. This is not exhaustive and there are further properties available.
21    */
22
23   // The HTTP status code, e.g. 200, 404
24   // const responseCode = response.getResponseCode();
25
26   // The HTTP response body, e.g. HTML for web pages:
27   // const responseText = response.getContentText();
28
29   // The failure strings from the configuration spreadsheet, as an array:
```



## New script

Script name:



Powered by [Apps Script](#)

[Advanced APIs](#)

[Documentation](#)



Expand



Scripts make changes on a user's behalf. You must authorize scripts before they can make any changes. [Learn more about authorisation](#)

[Authorise](#)

Code.gs

```
1 function main() {  
2  
3 }
```



# Let's start with checks

# 8020agent.com/ nle

This will **much** more useful  
If you follow along at home

## Scripts Made Simple - Exclusively for NLE Viewers



[Watch More Webinars](#)

**Note:** This challenge is just for viewers of the Next Level eCommerce webinar.  
If you'd like to play with the app either with sample data, or with your own Just click the 'Return to App' link at the top (after the webinar.)

Scared of scripts? There's no need to be.  
These quick challenges will show you just how easy they can be.  
We'll start super simple (literally 3 lines of code) and build up to creating automated reports and alerts.

### Before You Start:

- You need a single Google Ads account (not an MCC)
- You must have standard or edit access (read-only won't work)
- ✓ **Safe to Use:** None of these scripts will make any changes to your account

# Scripts Made Simple - Exclusively for NLE Viewers



[Watch More Webinars](#) 

**Note:** This challenge is just for viewers of the Next Level eCommerce webinar.


If you'd like to play with the app either with sample data, or with your own Just click the 'Return to App' link at the top

Scared of scripts? There's no need to be.

These quick challenges will show you just how easy they can be.

We'll start super simple (literally 3 lines of code) and build up to creating automated reports and alerts.

## Before You Start:

- You need a single Google Ads account (not an MCC)
- You must have standard or edit access (read-only won't work)
-  **Safe to Use:** None of these scripts will make any changes to your account



### No Experience Required

Start from absolute zero - I'll guide you step by step



### Real Working Code

Copy, paste & run actual scripts that work in your account



### Quick to Complete

10 super short levels, that take just 1-3 minutes each

[Follow Along Now](#) 

# Level 1: Let's Start Simple – Copy & Paste

Level 1 of 10

Begin by running a basic script that logs a simple message. This will help you see how Google Ads Scripts work.

## Step 1: Copy the Code

- Open Google Ads & click on Tools in the left menu.
- Click on Bulk Actions, then Scripts.
- Click the big blue '+' button & then 'New Script'.
- It should look like the image below with a 3-line script already in it.
- Click the button below the image to copy a new script to the clipboard (you can see the code we'll be using below).

Scripts >

### New script

Script name:  Powered by [Apps Script](#) [Advanced APIs](#) [Documentation](#) Expand

Scripts make changes on a user's behalf. You must authorize scripts before they can make any changes. [Learn more about authorisation](#) [Authorise](#)

Code.gs

```
1 function main() {
2     Logger.log('Hello, Google Ads Scripts!');
3 }
```

Copy Code to Clipboard

```
1 function main() {
2     Logger.log("Hello, Google Ads Scripts!");
3 }
```

# Level 1: Let's Start Simple – Copy & Paste

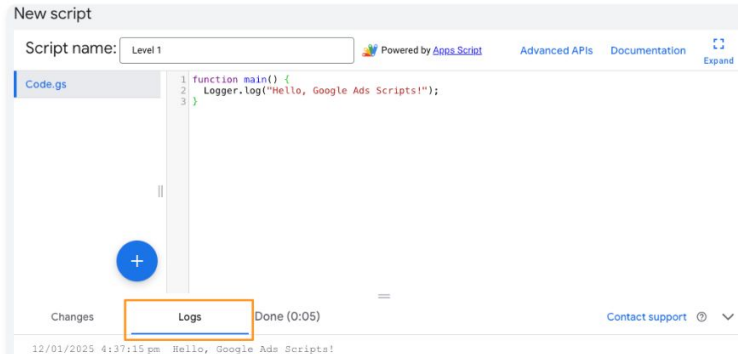
Level 1 of 10

Begin by running a basic script that logs a simple message. This will help you see how Google Ads Scripts work.

## Step 2: Preview the Code

- Name your script 'Level 1'.
- Delete the 3 lines of placeholder code from Google
- Paste in the code we just saved to your clipboard.
- Click 'Authorise' to give the script permission to run.
- Click 'Preview' (at the bottom) to get the script running in preview mode.
- (You might need to 'Authorize' again - you may need to scroll down to see it).
- If you see an 'App Blocked' message, talk to your Google admin.
- Once script is 'Done', check the output of your script by clicking 'Logs'.

I've run the code



The screenshot shows the Google Ads Scripts interface. At the top, it says "New script" and "Script name: Level 1". Below this is a code editor with the following code:

```
1 function main() {  
2   Logger.log("Hello, Google Ads Scripts!");  
3 }
```

At the bottom of the interface, there is a "Logs" panel which is highlighted with an orange box. The logs panel shows the following output:

```
12/01/2025 4:37:15 pm Hello, Google Ads Scripts!
```

# Level 1: Let's Start Simple – Copy & Paste

Level 1 of 10

Begin by running a basic script that logs a simple message. This will help you see how Google Ads Scripts work.

## Step 3: Confirm Output

Did the output look like this? (Note: if metrics are shown, numbers will be different)

Yes

No - Please Help

Changes Logs Done (0:07)

```
14/01/2025, 1:08:39 pm Hello, Google Ads Scripts!
```

## Don't Worry - We'll Figure This Out!

If you're not seeing the expected output, that's completely normal! There are several common reasons this might happen, and they're all easy to fix.



### Common Solutions

- Make sure you've clicked "Preview" before running the script
- Check that you've copied the entire code block
- Verify you're in the correct Google Ads account
- Try clearing your browser cache and refreshing



### Still Having Issues?

Sometimes scripts can take a few moments to show up in the logs. Try waiting 30 seconds and checking the logs again.



### Need More Help?

If you're still stuck, feel free to reach out to our support team. We're here to help you succeed!

[← Back to Main Tutorial](#)

[Try Level 1 Again](#)

## Level 2: Find Your Account Info

Level 2 of 10

Learn how to pull basic details about your account, like its name and ID. A quick way to see what's available.

### Step 1: Copy the Code

Click the button below to copy this new script to the clipboard.

 Copy Code to Clipboard

```
1 function main() {  
2     const account = AdsApp.currentAccount();  
3     Logger.log("Account Name: " + account.getName());  
4     Logger.log("Account ID: " + account.getCustomerId());  
5 }
```

## Level 2: Find Your Account Info

Level 2 of 10


Learn how to pull basic details about your account, like its name and ID. A quick way to see what's available.

### Step 2: Preview the Code

- In the script editing window, delete the code from Level 1.
- Paste this new code into the script window.
- You should now see the 5 lines of code below.
- Click 'Preview' to run the script, then check the logs.

I've run the code

New script

Script name:   Powered by [Apps Script](#) [Advanced APIs](#)

**Code.gs\***

```
1 function main() {
2   const account = AdsApp.currentAccount();
3   Logger.log("Account Name: " + account.getName());
4   Logger.log("Account ID: " + account.getCustomerId());
5 }
```

# Level 2: Find Your Account Info

Level 2 of 10

Learn how to pull basic details about your account, like its name and ID. A quick way to see what's available.

## Step 3: Confirm Output

Did the output look like this? (Note: if metrics are shown, numbers will be different)

Yes

No - Please Help

Changes Logs Done (0:07)

```
14/01/2025, 1:12:04 pm  Account Name: [Your Account Name]
14/01/2025, 1:12:05 pm  Account ID: [Your Account ID]
```

## Level 3: Check Your Key Stats

Level 3 of 10

Discover how to pull numbers like impressions, clicks, and CTR. These are the building blocks for analyzing performance.

### Step 2: Preview the Code

- This time let's click 'Run' instead of 'Preview'.
- A lightbox will appear, click 'Run without Preview'.
- You'll end up on the 'Script History' page. Click '3 log statements' to see the logs of your script run.
- To get back to your script, click 'Scripts' at the top.
- Scripts is where you can see all your scripts & schedule a script to run on a regular basis (look for 'Frequency').
- To get back to editing your script for the next level, just click on the script name.

I've run the code


### Scripts

Custom 13 Dec 2024 – 9 Jan 2025

Scripts Script history

The history of bulk actions (edits, rules and uploads) are now only visible based on the account that you're signed in to. Scripts and account that you're drilled in to. [Learn more about bulk actions with a manager account](#)

To see today's activity, add today's date to your date range.

 Owner: !! Sheet Society Add filter

User / Date & time	Owner	Status	Script name	Changes	Actions
		Finished successfully Completed in 4 seconds	<a href="#">Level 1</a>	✓ 3 log statements	

## Level 4: Choose Your Date Range

Change the date range in your script to see data for different time periods, like yesterday or the last 7 days. While we're at it, let's add some more metrics!

### Step 1: Copy the Code

- Copy the code.
- Paste it into Google Ads.
- If still on the 'Details' page you'll need to click the script name to go back to editing.

 Copy Code to Clipboard

```
1 function main() {  
2     const stats = AdsApp.currentAccount().getStatsFor("LAST_7_DAYS");  
3     Logger.log("Impressions: " + stats.getImpressions());  
4     Logger.log("Clicks: " + stats.getClicks());  
5     Logger.log("CTR: " + stats.getCtr());  
6     Logger.log("Cost: " + stats.getCost());  
7     Logger.log("Conv: " + stats.getConversions());  
8 }
```

**Checking Stats is ok  
Reports/Charts are  
even more useful**

# Level 5: Save Data to Google Sheets

Level 5 of 10

Logging metrics on the screen can be useful, but it's even better to save them to a Google Sheet.

So let's do that! This time a Google Sheet will be created for you - remember to check the logs to get the link.

## Step 1: Copy the Code

- Copy the code.
- Paste it into Google Ads.

 Copy Code to Clipboard

```
1 function main() {
2   const newSheet = SpreadsheetApp.create("Google Ads Data " + new Date().toLocaleDateString());
3   const sheet = newSheet.getActiveSheet();
4   const stats = AdsApp.currentAccount().getStatsFor("LAST_30_DAYS");
5   const headers = ["Metric", "Value"];
6   const data = [
7     ["Impressions", stats.getImpressions()],
8     ["Clicks", stats.getClicks()],
9     ["CTR", stats.getCtr()],
10    ["Cost", stats.getCost()],
11    ["Conversions", stats.getConversions()],
12    ["Cost per Conversion", stats.getCost() / stats.getConversions()]
13  ];
14  sheet.getRange(1, 1, 1, headers.length).setValues([headers]);
15  sheet.getRange(2, 1, data.length, data[0].length).setValues(data);
16  Logger.log("Sheet URL: " + newSheet.getUrl());
17 }
```

# You might need to scroll down

Changes

Logs

Done (0:01)

Contact support  

Campaign


Ad group

Change description

Status

No changes



Scripts make changes on a user's behalf. You must authorize scripts before they can make any changes. [Learn more about authorisation](#) 

Authorise

Close

Run

Save

Preview

# Level 5: Save Data to Google Sheets

Logging metrics on the screen can be useful, but it's even better to save them to a Google Sheet.

So let's do that! This time a Google Sheet will be created for you - remember to check the logs to get the link.

## Step 3: Confirm Output

Did the output look like this? (Note: if metrics are shown, numbers will be different)

Yes

No - Please Help

Changes Logs Done (0:07)

```
14/01/2025, 1:18:35 pm Sheet URL: [Google Sheet Link]
```



100% ▾



.0 ↵

.00 ↵

123

Defaul... ▾



10



A1



Metric

	A	B	C	D	E	F
1	Metric	Value				
2	Impressions	2499765				
3	Clicks	111662				
4	CTR	0.04467				
5	Cost	51990.19804				
6	Conversions	8641.863091				
7	Cost per Conver	6.01608675				
8						
9						
10						
11						

## Level 6: Organize Data into Tabs

Level 6 of 10

Now that you're getting the hang of this, we're going to try a new way of getting data. It's faster but looks a bit scarier.

Let's create 3 tabs with data for campaigns, ad groups, and keywords. This will make it easier to read and report on.

After all, the reason to pull this data out is to create reports & charts and gain new insights.

It might take about a minute or two to run.

Want to learn more? Try pasting this code into chatGPT or Claude.ai & ask it to explain what each part does.

NOTE: This code only looks for SEARCH campaigns. But try adjusting it (with AI) to get data for other campaign types.

### Step 1: Copy the Code

- Copy the code.
- Paste it into Google Ads.

 Copy Code to Clipboard

```
1 function main() {
2   const newSheet = SpreadsheetApp.create("Google Ads Search Data " + new Date().toLocaleDateString
3   ());
4   // Get the first sheet (default sheet) - this works regardless of language settings
5   const defaultSheet = newSheet.getSheets()[0];
6
7   const sheets = {
8     'Campaigns': { headers: ['Campaign', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cos
9     t/Conv'] },
10    'Ad Groups': { headers: ['Campaign', 'Ad Group', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conver
11    sions', 'Cost/Conv'] },
12    'Keywords': { headers: ['Campaign', 'Ad Group', 'Keyword', 'Match Type', 'Impressions', 'Click
13    s', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] }
14  };
15 }
```



**This script might take a  
minute or two**

	A	B	C	D	E	F	G	H
1	Campaign	Impressions	Clicks	CTR	Cost	Conversions	Cost/Conv	
2		0	0	0	0	0	0	
3		97032	37837	0.3899435238	18149.69734	4628.944207	3.920915122	
4		0	0	0	0	0	0	
5		0	0	0	0	0	0	
6		0	0	0	0	0	0	
7		0	0	0	0	0	0	
8		0	0	0	0	0	0	
9		0	0	0	0	0	0	
10		0	0	0	0	0	0	
11		0	0	0	0	0	0	
12		0	0	0	0	0	0	
13		0	0	0	0	0	0	
14		0	0	0	0	0	0	
15		0	0	0	0	0	0	
16		0	0	0	0	0	0	
17		0	0	0	0	0	0	
18		0	0	0	0	0	0	
19		0	0	0	0	0	0	
20		69741	6997	0.1003283578	4639.758347	372.117954	12.46851515	
21		28824	3151	0.1093186234	1757.828482	142.239959	12.3581903	
22		0	0	0	0	0	0	
23		0	0	0	0	0	0	
24		0	0	0	0	0	0	
25		0	0	0	0	0	0	
26		0	0	0	0	0	0	



**Now we'll add some  
filters to our code**

## Level 7: Making Your Sheet More Useful

Now that you've created your first report, you might notice it contains a lot of rows with zeros - campaigns or keywords with no activity.

(Assuming you had some Search campaigns & keywords in your account).

Let's clean up our data by filtering out these inactive items. This will make your report:

- Easier to read
- Faster to generate
- More focused on what matters

We'll also format the numbers to make them more readable.

### Step 1: Copy the Code

- Copy the code below.
- Create a new script in Google Ads (or edit your existing one).
- With a bit of help from AI, you could try adjusting the filters in the code to match your needs.

 Copy Code to Clipboard

```
1 function main() {
2   const newSheet = SpreadsheetApp.create("Google Ads Data " + new Date().toLocaleDateString());
3   const sheets = {
4     'Campaigns': { headers: ['Campaign', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] },
5     'Ad Groups': { headers: ['Campaign', 'Ad Group', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] },
6     'Keywords': { headers: ['Campaign', 'Ad Group', 'Keyword', 'Match Type', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] }
7   };
```

# Level 7: Making Your Sheet More Useful

Level 7 of 10

Now that you've created your first report, you might notice it contains a lot of rows with zeros - campaigns or keywords with no activity.

(Assuming you had some Search campaigns & keywords in your account).

Let's clean up our data by filtering out these inactive items. This will make your report:

- Easier to read
- Faster to generate
- More focused on what matters

We'll also format the numbers to make them more readable.

## Step 3: Confirm Output

Did the output look like this? (Note: if metrics are shown, numbers will be different)

Yes

No - Please Help

Changes Logs Done (0:07)

14/01/2025, 1:29:56 pm Filtered report created! URL: [Google Sheet Link]



# **Same Sheet each time? We need a variable**

## Level 8: Using the same sheet every time

Level 8 of 10

So far we've been creating a new sheet every time the script runs. While this is useful for testing, in practice you'll want to update the same sheet with new data, not create new sheets each time.

Let's modify our script to use the same sheet everytime. This way you can: add your own formatting and charts, track changes over time, share the sheet with your team & schedule the script to auto-update your sheet

### Step 1: Copy the Code


- Copy the complete code
- Paste it into Google Ads (or edit your existing script)

 Copy Code to Clipboard

```
1 function main() {  
2   // Replace this URL with your Google Sheet URL  
3   const SHEET_URL = '';  
4  
5   const sheet = SpreadsheetApp.openByUrl(SHEET_URL);  
6  
7   const sheets = {  
8     'Campaigns': { headers: ['Campaign', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] },  
9     'Ad Groups': { headers: ['Campaign', 'Ad Group', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] },  
10    'Keywords': { headers: ['Campaign', 'Ad Group', 'Keyword', 'Match Type', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] }  
11  };  
12
```

## Edit script

Script name:

 Powered by [Apps Script](#)

Code.gs\*

```
1 function main() {
2   // Replace this URL with your Google Sheet URL
3   const SHEET_URL = '';
4
5   const sheet = SpreadsheetApp.openByUrl(SHEET_URL);
6
7   const sheets = {
8     'Campaigns': { headers: ['Campaign', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Co
9     'Ad Groups': { headers: ['Campaign', 'Ad Group', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversion
10    'Keywords': { headers: ['Campaign', 'Ad Group', 'Keyword', 'Match Type', 'Impressions', 'Clicks', 'C
11  };
12
13  // Setup sheets and delete default
14  Object.keys(sheets).forEach(name => {
15    sheets[name].sheet = sheet.getSheetByName(name) || sheet.insertSheet(name);
16    sheets[name].sheet.clear();
17    sheets[name].sheet.getRange(1, 1, 1, sheets[name].headers.length).setValues([sheets[name].headers]);
18    sheets[name].sheet.getRange("1:1").setFontWeight("bold");
19    sheets[name].sheet.setFrozenRows(1);
20  });
21
22  // GAQL queries
```



## Level 8: Using the same sheet every time

Level 8 of 10

So far we've been creating a new sheet every time the script runs. While this is useful for testing, in practice you'll want to update the same sheet with new data, not create new sheets each time.

Let's modify our script to use the same sheet everytime. This way you can: add your own formatting and charts, track changes over time, share the sheet with your team & schedule the script to auto-update your sheet

### Step 2: Preview the Code

- IMPORTANT - Use the same Google account for the sheet that you use for Google Ads
- Open a new browser tab
- Type 'sheet.new' in address bar (and press enter) to create a new, blank Google Sheet
- ⚠️ Copy the entire URL from your browser's address bar - don't try to 'tidy it up' :)
- Find row 3 of the script, the one that says `const SHEET_URL = ''`;
- Enter the url of your new sheet between the single quotes
- Click 'Preview' to run the script
- Check that data appears in your sheet
- Try running the script again - now it updates the same sheet instead of creating a new one

I've run the code

```
Script name: Level 1 Powered by Apps Script Advanced APIs
```

```
Code.gs*  
1 function main() {  
2   // Replace this URL with your Google Sheet URL  
3   const SHEET_URL = 'https://docs.google.com/spreadsheets/d/17ptnPw0EjJCewwKH19gh85ey1mrUPrhV3rANmnIwTVk/edit?gid=0#gid=0';  
4  
5   const sheet = SpreadsheetApp.openByUrl(SHEET_URL);  
6  
7   const sheets = {  
8     'Campaigns': { headers: ['Campaign', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] },  
9     'Ad Groups': { headers: ['Campaign', 'Ad Group', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversions', 'Cost/Conv'] },  
10    'Keywords': { headers: ['Campaign', 'Ad Group', 'Keyword', 'Match Type', 'Impressions', 'Clicks', 'CTR', 'Cost', 'Conversio
```

## Level 9: Let's automate some email alerts

Set up automated email alerts when enabled campaigns get zero impressions.

Perfect for catching issues early, like disapproved ads or incorrect targeting.

### Step 1: Copy the Code

Copy the code below & paste it into Google Ads.

 Copy Code to Clipboard

```
1 const YOUR_EMAIL = ''; // enter your email address here between the single quotes
```

```
2
3 function main() {
4   const campaigns = AdsApp.campaigns()
5     .withCondition('Status = ENABLED')
6     .withCondition('Impressions = 0')
7     .forDateRange('YESTERDAY')
8     .get();
9
10  if (campaigns.totalNumEntities() === 0) {
11    Logger.log('All campaigns received impressions yesterday - no email sent');
12    return;
13  }
14
15  const problemCampaigns = [];
16  while (campaigns.hasNext()) {
17    const campaign = campaigns.next();
18    problemCampaigns.push(campaign.getName());
19  }
20
```

# Level 10: Optimise your profits with 8020agent

Level 10 of 10

Congratulations! Thanks to NLE, you've proven that Scripts aren't scary at all. In fact, they're incredibly powerful tools when used correctly.

You've just learnt to :

- preview, run & schedule scripts
- create & use Google Sheets with your scripts
- send yourself email alerts

Now you can put these building blocks together to create your own automations.

**Ready to take your automation to the next level? I've got a special gift & two other options for you:**

## PMax Asset Chart Script

Normally \$49

**Free gift for NLE viewers**

- ✓ Paste code into Google Ads
- ✓ Create a copy of the template sheet
- ✓ Add your URL to the script

[Get Script \\$0](#)

## More Scripts & Courses

### Advanced Google Ads Tools

- ✓ Google Ads scripts
- ✓ Video training courses
- ✓ Automation tools

[View All Tools →](#)

## Setup 8020agent

### Automate Everything

- ✓ Full automation setup
- ✓ Daily monitoring
- ✓ Automated optimization

[Learn More →](#)

**Congrats!**  
**Now you know**  
**what's possible**

What will you build?

# Next Level eCommerce is powered by...

