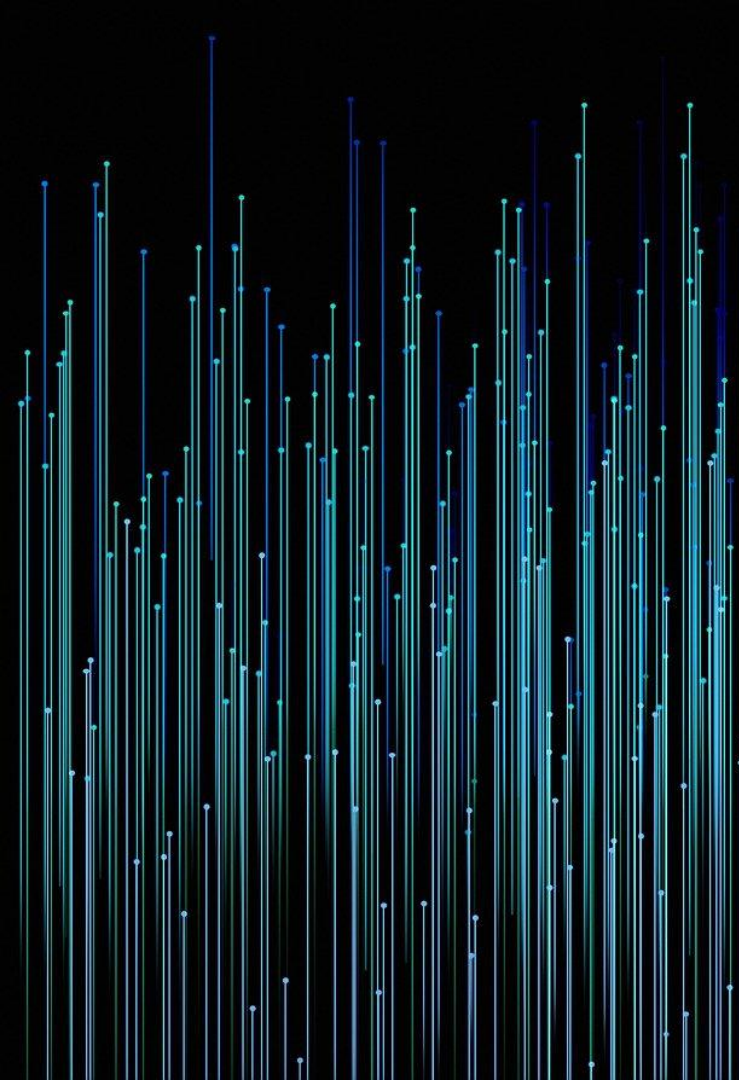




PROMPT LIBRARY

FOR TECHNICAL SEO



A guide to harnessing the power of AI to help you in your daily SEO tasks

As SEO professionals, we daily need to perform checks and monitor pages. Technical checks and optimisations play a very important role in ensuring search engine visibility and overall website performance.

With our prompt library, we aim to empower you with a set of actionable prompts and recommendations to automate some of those SEO checks and processes.

Each prompt has been created after several trial and error tests, and we would be more than happy to hear your ideas and feedback!

Use case: I want to create a robots.txt file that fully excludes some crawlers and excludes some parts for the rest. I also want to include the sitemap(s) in the file.

The prompt

- Please create a robots.txt file for the domain: **DOMAIN NAME**
- The following user agents should be completely disallowed from the entire website: **LIST OF USER AGENTS**
- The rest of user agents should have complete access to the website, except for the following website section(s)/path(s): **DIRECTORIES (EG. /admin/)**
- Please also state the sitemaps in the robots.txt. Their locations are the following: **LIST OF SITEMAP(S)**

Apps Script use cases

Use ChatGPT to supercharge your Apps Scripts creation.

AppsScripts are scripts that can help you automatise processes on your G-Suite products. Once your script has been created by ChatGPT, what you need is open your Google Sheet > Go to extensions > Apps Script > Paste code in a new script file, save and run.



Use case: I want to extract URLs from a sitemap

The prompt

- Please write me a Google Sheets app script. The script name should be **SCRIPT NAME**
- It should:
 - check the data on the **TAB NR.** tab of the sheet called: **NAME OF TAB TO CHECK**
 - Open the sitemap URL on cell A2 and return all URLs found under the sitemap in column B, starting from cell B2

Notes

In contrast to the previous scenarios, in this one we have titles in our column (so line 1 is used as a title, which is why data is starting from line 2 onwards)

Use case: I have a set of URLs that I want to ensure remain live and set to noindex, follow. I want to create a script that will check these URLs in a Google sheet and let me know whenever this is not the case.

The prompt

- Please write me a Google Sheets Apps script. The script name should be: **SCRIPT NAME**
- It should:
 - Check the data on the **TAB NR.** tab of the sheet called **NAME OF TAB TO CHECK**
 - Check the URLs on column A (from cell A1 onwards) for their status code and meta robots
- It should create:
 - one column with the status code discovered (and create column B with the status code)
 - one column with the meta robots discovered (and create column C with the content of the meta robots)
- It should ensure that the following criteria is being met:
 - The page has a 200 status code
 - The page's meta robots are noindex, follow
- If the above criteria is met, then there should be no notifications. If any of the above criteria is not met, an email should come to the address: **EMAIL ADDRESS** with the following text:

TEXT

Use case:

I have a set of URLs that I want to ensure are redirected (either permanently or temporarily). I want to create a script that will check these URLs in a Google sheet and let me know whenever this is not the case.

The prompt

- Please write me a Google Sheets Apps script. The script name should be: **SCRIPT NAME**
- It should:
 - Check the data on the **TAB NR.** tab of the sheet called **NAME OF TAB TO CHECK**
 - Check the URLs on column A (from cell A1 onwards) for their status code
- It should create:
 - one column with the status code discovered (and create column B with the status code)
- It should ensure that the following criteria is being met:
 - The page has a 301,302 or 308 status code
- If the above criteria is met, then there should be no notifications. If any of the above criteria is not met, an email should come to the address: **EMAIL ADDRESS** with the following text: **TEXT**

Use case: I have a set of URLs that I want to ensure are live, set to index and self-ref. canonical. I want to create a script that will check these URLs in a Google sheet and let me know whenever this is not the case.

The prompt

- Please create a Google Sheets app script. The script name should be **SCRIPT NAME**
- It should check the data on the **TAB NR.** tab of the sheet called: **NAME OF TAB TO CHECK**
- It should:
 - Fetch the URLs on column A from cell A2 onwards and check their status code, meta robots & canonical URL using the UrlFetchApp service. If the status code is a redirect, it should not follow this redirection.
- It should create:
 - headers "URL," "Status code," "Meta robots," and "Canonical" in the sheet from A1 to D1
 - one column with the status code discovered (and create column B with the status code)
 - one column with the meta robots discovered (and create column C with the content of the meta robots). If no meta robots are found, please write "no robots found" in column C.
 - one column with the canonical URL discovered (and create column D with the canonical URL). If no canonical is found, please write "no canonical found" in column D.
- if the URL has a status 200 code, then paint the cell B green
- if the URL's meta robots are set to "index, follow" or "all", then paint the cell C green
- if the content of column A is equal to the content of column D, then paint the cell D green
- If all cells in columns B, C and D are with value and green, then do not send notification.
- If any of the cells in columns B,C or D are with value and not green, then send notification. Notification should be an email to **EMAIL ADDRESS** with the text: **TEXT**

Use case: I want to extract URLs from a sitemap, and get their status code, meta robots and canonical URL.

The prompt

Please create a Google Apps script called **SCRIPT NAME** that:

- Checks the data on the **TAB NR.** tab of the sheet called **NAME OF TAB TO CHECK**
- Opens the sitemap URL found on cell A2 and stores all URLs found under the sitemap in column B, starting from cell B2 in an array called `urls`.
- Creates an empty array called `results`.
- Iterates over each URL in `urls` using a loop:
 - Inside the loop, fetch the response for the current URL
 - Get the response status code. In case it is a redirect, do not follow it.
 - Set the initial values of `metaRobots` and `canonicalURL` to 'no robots found' and 'no canonical found' respectively.
 - If the `statusCode` is 200:
 - Fetch the HTML content using `response.getContentText()` and assign it to the variable `html`.
 - Use regular expressions to extract the meta robots and canonical URL values from the HTML.
 - If the meta robots tag is found, assign its content to `metaRobots`.
 - If the canonical link tag is found, assign its href value to `canonicalURL`.
 - Append an array containing `url`, `statusCode`, `metaRobots`, and `canonicalURL` to the `results` array.

How did we do it? & Disclaimer

What we did, and what we propose to do is a mix of two things.

We **firstly** created the prompt as well as possible, but sometimes it led to some errors in the script. We would then **tweak** the script slightly to give us the result we wanted, and **ask ChatGPT what prompt it would need** to give us the corrected code version.

Please note, whenever we refer to a number of a tab or cell, make sure you **write these in letters** and not numbers (eg. fifth instead of 5th or 5)

Another thing to note is that even if the prompt is exactly the same, the code delivered might not be, so please use that as inspiration and run more tests to see what works for you!



```
what prompt would you need to create me the following code? function SitemapExtractor() {  
  var sheetName = 'Sitemap check';  
  var urlColumn = 'A';  
  var urlCell = 'A2';  
  var resultColumnCodes = 'C';  
  var resultColumnRobots = 'D';  
  var resultColumnCanonical = 'E';  
  var startingRow = 2;
```



Sure! Here's a prompt to generate the code you provided:



```
python Copy code  
  
prompt = """  
Create a function called 'SitemapExtractor()' that performs the following ta  
  
1. Define the following variables:  
- 'sheetName' with the value 'Sitemap check'  
- 'urlColumn' with the value 'A'  
- 'urlCell' with the value 'A2'  
- 'resultColumnCodes' with the value 'C'  
- 'resultColumnRobots' with the value 'D'  
- 'resultColumnCanonical' with the value 'E'  
- 'startingRow' with the value 2  
  
2. Get the active spreadsheet using 'SpreadsheetApp.getActiveSpreadsheet()'
```

Questions/suggestions? We are all ears!

If you have any ideas on checks you would like to automate, we would love to hear them! Stay tuned, as we will keep on updating our script library, one prompt at a time. You can follow us on [LinkedIn](#) or contact us directly at: seo@the-boutique-agency.de