

Google Apps Script: Doc to PDF Conversion

NOTE: Could not save link to chat script because automatically blocked since personal information may be involved.

Basic Conversion

The basic conversion algorithm is given here. A more sophisticated version that creates a finalized PDF subdirectory in the directory and routes all converted PDFs to there is given below.

Option 1: Single File Conversion

Use this for individual papers or specific manuscripts you are currently finalizing.

JavaScript

```
/**
 * Converts a specific Google Doc to a PDF in the same location.
 */
function convertSingleDocToPDF() {
  const fileId = 'REPLACE_WITH_YOUR_FILE_ID'; // Found in the document URL
  const originalFile = DriveApp.getFileById(fileId);
  const folder = originalFile.getParents().next();

  // Generate the PDF blob
  const pdfBlob = originalFile.getAs(MimeType.PDF);

  // Create the file in the same folder
  folder.createFile(pdfBlob).setName(originalFile.getName() + ".pdf");

  console.log("PDF Created: " + originalFile.getName());
}
```

Option 2: Batch Folder Conversion

Use this for organizing entire project directories (e.g., all drafts for a specific textbook chapter).

JavaScript

```
/**
 * Converts all Google Docs in a specified folder into PDFs.
```

```

*/
function convertFolderToPDF() {
  const folderId = 'REPLACE_WITH_YOUR_FOLDER_ID';
  const folder = DriveApp.getFolderById(folderId);
  const files = folder.GetFilesByType(MimeType.GOOGLE_DOCS);

  while (files.hasNext()) {
    const file = files.next();
    const pdfBlob = file.getAs(MimeType.PDF);

    // Check if PDF already exists to avoid duplicates (optional)
    const newName = file.getName() + ".pdf";
    folder.createFile(pdfBlob).setName(newName);

    console.log("Converted: " + newName);
  }
}

```

Quick Setup Guide

1. **Open Apps Script:** Go to script.google.com.
2. **Create Project:** Click **New Project** and paste the code.
3. **Identify IDs:** The ID is the long string of characters in your browser's address bar when you have the Doc or Folder open (e.g., 1abc123...).
4. **Permissions:** When you click **Run**, Google will ask for permission to access your Drive. Since this is your own script, you can safely grant it.

Would you like me to add a line of code to these scripts that automatically moves the newly created PDFs into a "Finalized" folder once they are converted?

Conversion with sorting PDFs into new folder

To keep your workspace organized, here are the updated scripts. I have added logic to automatically create a "Finalized PDFs" folder (if it doesn't already exist) and route all your new files there.

Batch Conversion with Auto-Sorting

This version is ideal for managing multiple drafts. It searches for a folder named **"Finalized PDFs"** within your project directory; if it's missing, the script creates it for you.

JavaScript

```
/**
 * Converts all Docs in a folder and moves them to a 'Finalized PDFs' subfolder.
 */
function convertAndOrganizePDFs() {
  const parentFolderId = 'REPLACE_WITH_YOUR_FOLDER_ID';
  const parentFolder = DriveApp.getFolderById(parentFolderId);

  // Find or create the 'Finalized PDFs' subfolder
  let targetFolder;
  const subFolders = parentFolder.getFoldersByName("Finalized PDFs");

  if (subFolders.hasNext()) {
    targetFolder = subFolders.next();
  } else {
    targetFolder = parentFolder.createFolder("Finalized PDFs");
  }

  const files = parentFolder.GetFilesByType(MimeType.GOOGLE_DOCS);

  while (files.hasNext()) {
    const file = files.next();
    const pdfBlob = file.getAs(MimeType.PDF);
    const newName = file.getName() + ".pdf";

    // Create the PDF directly in the target folder
    targetFolder.createFile(pdfBlob).setName(newName);

    console.log("Converted and filed: " + newName);
  }
}
```

How the File Structure Will Look

After running this, your Google Drive directory will look like this:

5. [Your Project Folder]

- *Manuscript_Draft_v1* (Google Doc)
- *Manuscript_Draft_v2* (Google Doc)
-  **Finalized PDFs**
 - *Manuscript_Draft_v1.pdf*

- *Manuscript_Draft_v2.pdf*

Pro Tip for Large Volumes

If you are running this on a folder with dozens of files, Google Apps Script has a **6-minute execution limit**. If the script times out, just run it again; it will pick up where it left off (though it may create duplicates unless you add a check for existing file names).