Daily Update - Apps Script

Dieses Skript ermöglicht dir, Folien automatisch basierend auf Wochentagen ein- oder auszublenden. Hier ist, wie du es verwenden kannst:

1. Öffnen Sie Ihre Google Slides Präsentation
2. Gehe zu Erweiterungen > Apps Script
3. Füge den Code ein und ersetzen Sie 'DEINE\_PRÄSENTATIONS\_ID' mit der ID deiner Präsentation
4. In den Notizen jeder Folie kannst du Steuerungsbefehle hinzufügen:  
   #show-on: Mon,Tue,Wed  
   #hide-on: Sat,Sun

Das Skript bietet drei Hauptfunktionen:

* managePresentationSlides(): Führt die eigentliche Foliensteuerung durch
* createDailyTrigger(): Erstellt einen täglichen Trigger, der das Skript automatisch ausführt
* manualUpdate(): Ermöglicht manuelles Aktualisieren der Folien

**WICHTIG:**

Unter "Erweiterungen > Apps Script", musst du kein Dienst oder eine Bereitstellung angeben. Aber du musst ein Trigger erstellen, damit das Skript automatisch zu bestimmten Zeiten ausgeführt wird.

Sie musst du folgendes tun:

1. Klicke auf "Ausführen" (▶️) um das Skript einmal manuell zu testen
2. Nach dem ersten Test klicke auf das Uhr-Symbol "⏰" in der linken Seitenleiste (Trigger)
3. Klicke auf den blauen Button "+ Trigger hinzufügen"
4. Stelle die Trigger-Einstellungen wie folgt ein:
   1. Funktion: manageTimeBasedSlides
   2. Ereignisquelle: "Zeit"
   3. Zeitintervall: "Tag-Timer"
   4. Intervall wähle am besten Tageszeit : z.B. "1:00 Uhr bis 2:00 Uhr" (damit die Änderungen früh morgens passieren)

So wird das Skript einmal täglich ausgeführt und prüft die Wochentage-Regeln in den Notizen Ihrer Folien.

function managePresentationSlides() {

  // ID deiner Präsentation

  const PRESENTATION\_ID = 'DEINE\_PRÄSENTATIONS\_ID';

  // Präsentation öffnen

  const presentation = SlidesApp.openById(PRESENTATION\_ID);

  // Aktuelles Datum

  const today = new Date();

  const dayOfWeek = today.getDay(); // 0 = Sonntag, 1 = Montag, ...

  // Alle Folien durchgehen

  const slides = presentation.getSlides();

  slides.forEach((slide, index) => {

    // Notizen der Folie abrufen (hier können wir Steuerungsbefehle hinterlegen)

    const notes = slide.getNotesPage().getSpeakerNotesShape().getText().asString();

    // Beispiel für Steuerungsbefehle in den Notizen:

    // #show-on: Mon,Tue,Wed

    // #hide-on: Sat,Sun

    const showOnMatch = notes.match(/#show-on:\s\*(.\*)/);

    const hideOnMatch = notes.match(/#hide-on:\s\*(.\*)/);

    // Tage in lesbare Format konvertieren

    const dayNames = {

      0: 'Sun',

      1: 'Mon',

      2: 'Tue',

      3: 'Wed',

      4: 'Thu',

      5: 'Fri',

      6: 'Sat'

    };

    const currentDay = dayNames[dayOfWeek];

    // Folie basierend auf den Regeln ein- oder ausblenden

    if (showOnMatch) {

      const allowedDays = showOnMatch[1].split(',').map(day => day.trim());

      if (!allowedDays.includes(currentDay)) {

        slide.setSkipped(true);

      } else {

        slide.setSkipped(false);

      }

    }

    if (hideOnMatch) {

      const hiddenDays = hideOnMatch[1].split(',').map(day => day.trim());

      if (hiddenDays.includes(currentDay)) {

        slide.setSkipped(true);

      } else {

        slide.setSkipped(false);

      }

    }

  });

}

// Funktion zum Erstellen eines Zeit-basierten Triggers

function createDailyTrigger() {

  // Alle bestehenden Trigger löschen

  const triggers = ScriptApp.getProjectTriggers();

  triggers.forEach(trigger => ScriptApp.deleteTrigger(trigger));

  // Neuen täglichen Trigger erstellen

  ScriptApp.newTrigger('managePresentationSlides')

    .timeBased()

    .everyDays(1)

    .atHour(1) // Wird um 1 Uhr nachts ausgeführt

    .create();

}

// Funktion zum manuellen Ausführen der Präsentationssteuerung

function manualUpdate() {

  managePresentationSlides();

}