

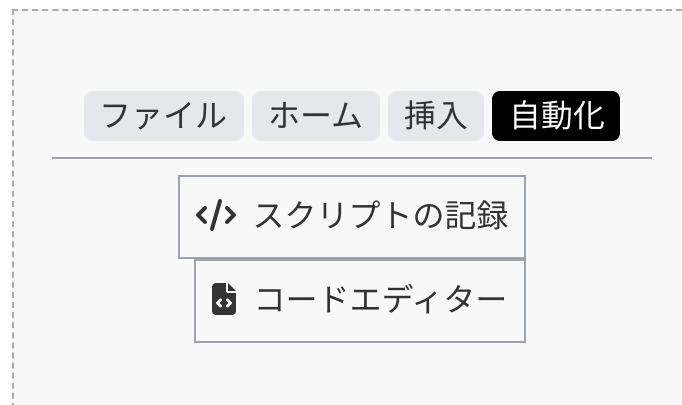
Officeスクリプトとは？ ～Excel作業を自動化する新しい力～

Officeスクリプトとは？

- ✔ Microsoft Excel のオンライン版で使える自動化ツール
- ✔ VBAに似た仕組みだが、よりシンプルで安全に使える
- ✔ JavaScriptベースで構成されているが、コーディング経験がなくても大丈夫

どこで使えるの？

- 🌐 Excel for the Web（Microsoft 365）上で利用可能
- ▶ [自動化] タブ → 「スクリプトの記録」または「コードエディター」をクリック



学ぶとどうなる？

- ✍ 「転記」「集計」「並べ替え」などの反復作業をボタン1つで実行
- 📈 Excel業務の効率アップ → ミスが減り、時間と心にゆとりが生まれる



作業時間短縮！

💡 最初はむずかしそうに見えても大丈夫。今日は"やさしい自動化"と一緒に体験していきましょう。

Officeスクリプトって、どんなことができるの？

活用シーンの例

- 📅 曜日ごとに同じ処理を繰り返す日報作成
- 📊 データ収集ファイルから集計表をつくる
- ⬇️ 顧客一覧の並べ替え・重複削除

💡 「あ、これ私の仕事にもある！」と思った方はOfficeスクリプトの恩恵を受けられます

サンプルコード

```
function main(workbook: ExcelScript.Workbook) {  
  let sheet = workbook.getWorksheet("売上");  
  let range = sheet.getRange("B2:B10");  
  range.setValues([[ "合計" ]]);  
}
```

これは何をするコード？

'売上'シートのB2～B10に'合計'と入力するだけのスクリプトです

workbook = ファイル

sheet = シート

range = 範囲

コードの基本構造

```
function main(workbook: ExcelScript.Workbook) {
```

すべてのスクリプトの入り口

※ 型の名前（ExcelScript.Workbook）は最初は気にしなくてOK

```
  let sheet = workbook.getWorksheet("売上");
```

データの取得や操作対象の指定

「let」 = 変数（データの入れ物）を用意する

```
  range.setValues([[ "合計" ]]);
```

値をセットする命令

「setValues」 = 値を設定する

💡 このように、Officeスクリプトは"やることの順番"を簡単な言葉で書いていくイメージ。
ひとつひとつ見ていけば、あなたにもきっとできます！

よくある悩み①：毎日Excelで数値を手集計...

手作業での集計の流れ



集計担当者

「また間違えた...」

「これ昨日もやったよね...」

❗ 手作業での集計は時間がかかるうえ、ミスが起きやすい...

✂ この作業、スクリプトなら1クリックで完了します！

💡 次のスライドでは、ボタン1つで集計できる方法をご紹介します。

Officeスクリプトで「集計ボタン」を作ってみよう

スクリプト実行ボタン



Excel画面

リボンメニュー：

ホーム

挿入

数式

自動化

「自動化」タブを選択

▶ スクリプトを実行

コードエディタ

ワンクリックで集計完了 スクリプト編集

▶ ボタン1つで複数シートのデータを自動集計

自動化の流れ

支店A

日付・担当者・売上
...
...

支店B

日付・担当者・売上
...
...

✎ スクリプトが自動処理

↓

集計結果

すべての支店データが統合されました！

日付	担当者	売上
2025/7/1	佐藤	15,000円
2025/7/1	田中	22,000円

✔ データが自動的に集計されました

💡 これはほんの一例。あなたの仕事にも活かれます。
毎日の定型作業をスクリプト化すれば、時間の節約とミスの削減ができます

自分でも書ける？シンプルなスクリプトの中身

コード例と解説

```
function main(workbook: ExcelScript.Workbook) {
```

スクリプトの入口

すべてのOfficeスクリプトはここから始まります
※ 最初は「お決まりの書き方」として覚えるだけでOK

```
let summarySheet = workbook.getWorksheet("集計結果");
```

出力先シートの指定

集計結果 という名前のシートを取得
※ シート名を変えるところを修正

```
let sourceSheet = workbook.getWorksheet("支店A");
```

データ元シートの指定

支店A という名前のシートを取得
ここを「支店B」に変えると集計対象が変わる！

```
let data = sourceSheet.getRange("A2:C10").getValues();
```

データの読み取り

A2からC10までの範囲のデータを取得

Range = セル範囲

getValues() = 値を取得

```
summarySheet.getRange("A2:C10").setValues(data);
```

データの書き込み

取得したデータを集計結果シートに貼り付け

setValues() = 値を設定

💡 これらの行を1つずつ理解していけば、スクリプトの仕組みがわかります！

まずは「どのシートから」「どの範囲を」「どこに出力するか」という基本の流れをつかみましょう。

今日のポイントと「あなたの仕事での使いどころ」

本日学んだこと

- 1 Officeスクリプトは自動化の味方**
Excel上で作業を自動化でき、繰り返し作業から解放されます
- 2 ボタン1つで日々の集計を簡略化**
複数シートのデータ収集・加工が瞬時に完了します
- 3 カスタマイズが可能**
自分の業務に合わせてスクリプトを調整できます

こんな場面で使えるかも

-  **複数ファイルの統合**
各部署から届くデータを一つのレポートにまとめる作業を自動化
-  **定期レポート作成**
週次・月次の集計処理を一括で実行し、時間を大幅削減
-  **複写と並べ替え作業**
顧客データの整理・抽出・条件付き書式設定を自動で適用

😊 最初はマネからでOK。少しずつ自分用に直していけば大丈夫です😊
日々の小さな自動化が、大きな時間の節約につながります

やってみよう！スクリプト付きファイルで自動化体験

配布ファイルの説明



OfficeScript_Practice_Task.xlsx
スクリプト付きExcelファイル

ファイル構成



支店A / 支店B：日付・担当者・売上のサンプルデータ



集計結果：スクリプトで自動集計して出力するシート

ポイント

ボタン1つでデータ集計を行えるようスクリプトが組み込まれています。初期状態では「集計結果」シートは空です。

演習課題

1

集計ボタンを押して結果を確認してみよう

「自動化」タブ → 「コードエディター」を開いて「実行」ボタンで動作確認

2

集計対象のシート名を変更してみよう

コード内の `sourceSheets = ["支店A", "支店B"]` の部分を編集

3

範囲を少し変えて自分仕様にしてみよう

15,000円以上のデータだけを抽出するif文を追加してみましょう

応用課題：

売上が15,000円以上の行だけを抽出するようにスクリプトを修正してみましょう。

```
if (Number(row[2]) >= 15000) {  
  // ここに条件に合う場合の処理を書く  
}
```

自信を引き出す問いかけ



できたこと、気づいたことをメモしてみましょう



あなたの仕事でも「ボタン1つ」でラクになる場面はありませんか？

スクリプト例

```
function main(workbook: ExcelScript.Workbook) {  
  // 出力先のシートを取得  
  const summarySheet = workbook.getWorksheet("集計結果");  
  summarySheet.getRange("A2:C1000").clear();  
  
  // 見出しのセット (A1~C1)  
  summarySheet.getRange("A1").setValue("日付");  
  summarySheet.getRange("B1").setValue("担当者");  
  summarySheet.getRange("C1").setValue("売上");  
  
  // データを取り出す対象のシート名一覧  
  const sourceSheets = ["支店A", "支店B"];
```

💡 「自動化」タブ → 「コードエディター」を開いて、上記のスクリプトを貼り付けます。
「スクリプトを保存」 → 「実行」で、集計結果シートにデータがまとめられることを確認しましょう。