

Guide pratique Google Sheets

Bases et Analyse de données

(Modules 1 à 3)

Ali Houssene Silahi

`silahi.alihoussene@gmail.com`

Data Analyst

9 octobre 2025

Public visé : Apprenants des modules 1 à 3 — débutants à intermédiaires.

Objectif : Fournir un guide de référence clair, pratique et complet pour comprendre et appliquer les fonctions fondamentales et avancées de Google Sheets.

Jeu de données d'exemple : Transactions_Simplifiees (extraits inspirés de PaySim) — utilisé tout au long du document.

Table des matières

1	Bases essentielles de Google Sheets	4
1.1	Qu'est-ce qu'un tableur ? Pourquoi Google Sheets est un tableur ?	4
1.1.1	Exemple – Structure du jeu de données	4
1.2	Cellule, plage, feuille	4
1.3	Références : relative, absolue, mixte, plage nommée	5
1.3.1	Types	5
1.3.2	Astuce : touche F4	5
1.4	Types de données	5
1.5	Formules et fonctions — paramètre obligatoire vs facultatif	5
1.5.1	Comment reconnaître paramètres obligatoires / facultatifs ?	5
1.6	Expressions logiques — définitions et exemples	6
2	Fonctions arithmétiques et conditionnelles	7
2.1	Fonctions de base (SUM, AVERAGE, MIN, MAX, COUNT)	7
2.1.1	Variantes conditionnelles	7
2.2	Fonctions conditionnelles : définition et critères	7
2.2.1	Qu'est-ce qu'un critère ?	7
2.2.2	Exemples pratiques	8
2.3	Bonnes pratiques	8
3	Fonctions logiques et expressions conditionnelles	9
3.1	Définition et objectif	9
3.2	IF, IFS, AND, OR	9
3.2.1	Exemple pratique : statut automatisé	9
3.3	IFERROR et ISERROR	9
4	Fonctions de recherche et d'automatisation	10
4.1	Fonctions de recherche : concept	10
4.2	VLOOKUP et HLOOKUP	10
4.3	INDEX et MATCH : principe et lien	10
4.3.1	Pourquoi INDEX+MATCH ?	11

4.4	ARRAYFORMULA : appliquer une formule à une colonne entière	11
4.4.1	Bonnes pratiques ARRAYFORMULA	11
4.5	IMPORTDATA, IMPORTRANGE, IMPORTXML	11
5	Fonctions avancées — Query et transformation	12
5.1	Présentation de QUERY	12
5.2	Syntaxe générale	12
5.3	Clauses disponibles (exemples)	12
5.4	Exemples pratiques avec Transactions_Simplifiees	12
5.5	FILTER, SORT, VSTACK	13
5.6	Paramètres des fonctions et comportement par défaut	13
	Annexes	15

1. Bases essentielles de Google Sheets

1.1. Qu'est-ce qu'un tableur ? Pourquoi Google Sheets est un tableur ?

Un **tableur** est un logiciel permettant de saisir, organiser, calculer, analyser et visualiser des données structurées en **lignes** et **colonnes**. Google Sheets est un tableur en ligne offrant : accès depuis un navigateur, collaboration en temps réel, fonctions de calcul dynamiques, et intégration avec l'écosystème Google.

1.1.1. Exemple – Structure du jeu de données

Table : Transactions__Simplifiees

Date	Type	Montant	Statut	Client_ID	Région
01/10/2025	Paielement	1200	Succès	CL001	Dakar
01/10/2025	Retrait	800	Échec	CL002	Thiès
02/10/2025	Paielement	3200	Succès	CL003	Saint-Louis
02/10/2025	Transfert	450	Succès	CL001	Dakar
03/10/2025	Retrait	600	Succès	CL004	Kaolack
03/10/2025	Paielement	5200	Succès	CL002	Thiès

1.2. Cellule, plage, feuille

Cellule : intersection d'une ligne et d'une colonne (ex. B2).

Plage : zone de cellules contiguës (ex. A1:C10).

Feuille : onglet unique dans un fichier Google Sheets.

Astuce : Pour nommer facilement une plage (plage nommée) : Données → Plages nommées → Ajouter. Exemple : nommer C2:C100 en Montants.

1.3. Références : relative, absolue, mixte, plage nommée

1.3.1. Types

- **Relative** : A1 — change si copié.
- **Absolue** : \$A\$1 — fixe colonne et ligne.
- **Mixte** : \$A1 ou A\$1 — fige colonne ou ligne.
- **Plage nommée** : Montants pour C2:C100.

1.3.2. Astuce : touche F4

Après avoir saisi une référence (ex. A1) dans la barre de formule, appuie sur **F4** pour cycler :

1. \$A\$1 (absolue)
2. A\$1 (ligne figée)
3. \$A1 (colonne figée)
4. A1 (relative)

Utilise F4 pour verrouiller les plages dans les formules quand tu copies.

1.4. Types de données

- Texte : "Paiement", "OK"
- Nombre : 1200, 0.85
- Date / Heure : 01/10/2025, 12:30
- Booléen : TRUE, FALSE
- Formule : commence par =

1.5. Formules et fonctions — paramètre obligatoire vs facultatif

Formule : expression calculée, commence par =.

Fonction : primitive prédéfinie (ex. SUM, IF) avec paramètres.

1.5.1. Comment reconnaître paramètres obligatoires / facultatifs ?

- La documentation de la fonction liste les paramètres ; ceux sans crochets sont obligatoires, ceux entre crochets [...] sont facultatifs.

- Exemple : `ROUND(value, [places])` → `value` est obligatoire ; `places` facultatif (défaut : 0).

Exemple (copiable) :

```
=ROUND(12.345, 2)    -> 12.35  
=ROUND(12.345)       -> 12
```

1.6. Expressions logiques — définitions et exemples

Une expression logique renvoie `TRUE` ou `FALSE`.

- `=C2>1000` renvoie `TRUE` si Montant en C2 > 1000.
- `=B2="Paielement"` teste l'égalité texte.
- `=AND(C2>1000, D2="Succès")` combine deux conditions.
- `=OR(E2="Dakar", E2="Thiès")`

2. Fonctions arithmétiques et conditionnelles

2.1. Fonctions de base (SUM, AVERAGE, MIN, MAX, COUNT)

Ces fonctions permettent de résumer des plages numériques.

Exemples (copiables) :

<code>=SUM(C2:C7)</code>	# somme des montants
<code>=AVERAGE(C2:C7)</code>	# moyenne des montants
<code>=MIN(C2:C7)</code>	# minimum
<code>=MAX(C2:C7)</code>	# maximum
<code>=COUNT(C2:C7)</code>	# compte les cellules numériques

2.1.1. Variantes conditionnelles

- `SUMIF(range, critère, [sum_range])`
- `COUNTIF(range, critère)`
- `AVERAGEIF(range, critère, [average_range])`

Somme des montants pour les transactions Succès :

```
=SUMIF(D2:D7, "Succès", C2:C7)
```

2.2. Fonctions conditionnelles : définition et critères

Une **fonction conditionnelle** applique un calcul seulement si un ou plusieurs critères sont satisfaits.

2.2.1. Qu'est-ce qu'un critère ?

Un **critère** est une règle logique qui sélectionne des éléments. Format fréquent :

- Comparaisons numériques : ">1000", "<5000"
- Égalités texte : "Succès", "=Dakar"
- Expressions combinées via fonctions logiques (AND/OR)

2.2.2. Exemples pratiques

```
=COUNTIF(D2:D7, "Succ s")           # nombre de transactions  
r ussies  
=SUMIF(E2:E7, "=Dakar", C2:C7)       # somme des montants pour la  
r gion Dakar  
=AVERAGEIF(B2:B7, "Paie ment", C2:C7) # moyenne des paiements
```

2.3. Bonnes pratiques

- Toujours vérifier le format des données (nombre vs texte).
- Utiliser plages nommées pour lisibilité : =SUM(Montants).
- Protéger votre feuille si des plages sont sensibles.

3. Fonctions logiques et expressions conditionnelles

3.1. Définition et objectif

Fonctions logiques : permettent de prendre des décisions dans les formules en renvoyant différentes valeurs selon que des conditions sont vraies ou fausses.

3.2. IF, IFS, AND, OR

Exemples (copiables) :

```
=IF(C2>1000, "Gros montant", "Petit montant")  
=IFS(C2>5000, "Tr s lev ", C2>1000, "Moyen", TRUE, "Bas")  
=AND(C2>1000, D2=" Succ s")  
=OR(E2="Dakar", E2=" Thi s")
```

3.2.1. Exemple pratique : statut automatisé

On souhaite remplir une colonne StatutLabel :

```
=IF(AND(D2=" Succ s", C2>1000), "Valide", " v rifier")
```

3.3. IFERROR et ISERROR

- IFERROR(expression, valeur_si_erreur) : cachera les erreurs et affichera valeur_si_erreur.
- ISERROR(expression) : renvoie TRUE si expression génère une erreur.

```
=IFERROR(VLOOKUP("CL999", A2:E7, 3, FALSE), "Client inconnu")
```

4. Fonctions de recherche et d'automatisation

4.1. Fonctions de recherche : concept

Une **fonction de recherche** permet de retrouver une information basée sur une clé (ex : ID client) dans une table ou une plage.

4.2. VLOOKUP et HLOOKUP

VLOOKUP (vertical lookup) :

`=VLOOKUP(valeur_recherchée, plage, numéro_colonne, [exact_ou_approx])`

Récupérer le montant associé à CL001 (recherche exacte) :

`=VLOOKUP("CL001", E2:C7, 3, FALSE)`

Remarque : VLOOKUP cherche la valeur dans la première colonne de la plage et renvoie la colonne indiquée par l'index. Si la colonne clé n'est pas la première, utilisez INDEX + MATCH.

4.3. INDEX et MATCH : principe et lien

MATCH renvoie la position d'une valeur dans une plage ; INDEX renvoie la valeur à une position donnée.

Combinaison (copiable) :

<code>=MATCH("CL001", E2:E7, 0)</code>	# position de CL001
<code>=INDEX(C2:C7, MATCH("CL001", E2:E7, 0))</code>	# montant correspondant

4.3.1. Pourquoi INDEX+MATCH ?

- Indépendant de l'ordre des colonnes (VLOOKUP exige la clé à gauche).
- Plus robuste lors d'insertion/suppression de colonnes.
- Permet de faire des recherches bidirectionnelles.

4.4. ARRAYFORMULA : appliquer une formule à une colonne entière

Exemples (copiables) :

```
=ARRAYFORMULA(IF(C2:C>1000, "Gros", "Petit"))  
=ARRAYFORMULA(C2:C * 1.18) # appliquer TVA
```

4.4.1. Bonnes pratiques ARRAYFORMULA

- Place toujours l'ARRAYFORMULA en tête de colonne (ex : cellule d'entête).
- Utilise des plages entières (ex : C2 :C) plutôt que plages finies quand possible.

4.5. IMPORTDATA, IMPORTRANGE, IMPORTXML

- IMPORTDATA("url_csv") : importe un CSV.
- IMPORTRANGE("url", "Feuille!A1:C100") : importe depuis un autre Google Sheets.
- IMPORTXML("url", "xpath") : extrait du contenu structuré.

Importer un extrait PaySim (CSV) :

```
=IMPORTDATA("https://exemple.com/paysim_sample.csv")
```

5. Fonctions avancées — Query et transformation

5.1. Présentation de QUERY

QUERY permet d'interroger une plage avec un langage proche de SQL : SELECT, WHERE, GROUP BY, ORDER BY, LIMIT, OFFSET, LABEL, etc.

5.2. Syntaxe générale

```
=QUERY(plage, "requête", [nombre_en_tetes])
```

5.3. Clauses disponibles (exemples)

- SELECT : colonnes et expressions (ex : SELECT B, SUM(C))
- WHERE : filtres (ex : WHERE D='Succès' AND C>1000)
- GROUP BY : regrouper par colonne (ex : GROUP BY B)
- ORDER BY : tri (ex : ORDER BY SUM(C) DESC)
- LIMIT : nombre de lignes retournées (ex : LIMIT 10)
- OFFSET : ignorer N premières lignes (ex : OFFSET 2)
- LABEL : renommer colonnes (ex : LABEL SUM(C) 'Montant total')
- Agrégations : SUM(), AVG(), MIN(), MAX(), COUNT()

5.4. Exemples pratiques avec Transactions_Simplifiees

1) Somme des montants par type pour les transactions réussies :

```
=QUERY(A1:F7 ,  
"SELECT B, SUM(C)  
WHERE D = 'Succès'
```

```
GROUP BY B
ORDER BY SUM(C) DESC",
1)
```

2) Top 3 des clients par montant total (transactions réussies) :

```
=QUERY(A1:F7,
"SELECT E, SUM(C)
WHERE D = 'Succ s'
GROUP BY E
ORDER BY SUM(C) DESC
LIMIT 3",
1)
```

3) Transactions CASH_OUT supérieures à 1000 :

```
=QUERY(A1:F7,
"SELECT *
WHERE B = 'CASH_OUT' AND C > 1000",
1)
```

5.5. FILTER, SORT, VSTACK

- `FILTER(tableau, critere1, [critere2,...])` : extrait lignes satisfaisant tous les critères (critères séparés signifiant AND).
 - `SORT(tableau, colonne_tri, asc_ou_desc)` : trie un tableau.
 - `VSTACK(plage1, plage2, ...)` : empile verticalement plusieurs plages.
- Transactions > 1000 et succès :

```
=FILTER(A2:F7, C2:C7>1000, D2:D7="Succ s")
```

5.6. Paramètres des fonctions et comportement par défaut

Pour chaque fonction, vérifie :

- l'ordre des paramètres (obligatoires d'abord),
- le type accepté (texte / nombre / plage),

- le comportement par défaut (ex : **VLOOKUP** par défaut peut faire une correspondance approximative si le dernier argument est omis).

Toujours explicitement mettre **FALSE** pour des recherches exactes dans **VLOOKUP** : . . . , **FALSE**.

Annexes

Exercices de consolidation

1. Importer l'extrait et nommer la plage des montants **Montants**. Calculer la somme totale.
2. Créer une colonne **StatutLabel** via **ARRAYFORMULA** :

```
=ARRAYFORMULA(IF(ROW(C2:C)-ROW(C2)+1>COUNTA(C2:C), "", IF((D2:D="Succès")*(C2:C>1000), "Valide", "à vérifier")))
```
3. Utiliser **INDEX+MATCH** pour retrouver le montant de **CL003**.
4. Construire avec **QUERY** la somme des montants par région pour les transactions réussies.

Corrigés rapides (extraits)

```
=SUM(Montants)
=INDEX(C2:C7, MATCH("CL003", E2:E7, 0))
=QUERY(A1:F7, "SELECT F, SUM(C) WHERE D='Succès' GROUP BY F", 1)
```

Remarque finale : Ce document sert de guide de référence. Pour des exercices plus avancés (scripts Apps Script, Add-ons, BigQuery) un module complémentaire pourra être ajouté.