



DAX User-Defined Functions (UDF)

Tristan Malherbe

Qu'est-ce qu'une User-Defined Function DAX ?

Fonction DAX **personnalisée** & **réutilisable** dans votre modèle de données Power BI

Syntaxe

```
DEFINE FUNCTION AddTwoNumbers = (x: int64, y: int64) => x + y
```

Principaux intérêts des UDF en DAX

1. **Factoriser** votre code: **Centraliser** à un seul endroit une logique répliquable
2. **Simplifier** votre code: **Masquer** certaines **complexités** inhérentes au langage DAX
3. **Réutiliser** votre code: **Appeler** ces fonctions **DAX UDF** dans votre modèle ou les réutiliser ailleurs.

Conclusion

- 1) Les **UDF** sont un excellent moyen de **factoriser, simplifier** et **réutiliser** votre code DAX.
- 2) Vous pouvez les **appeler partout** où le **DAX est présent: Mesures, Colonnes, Formats Dynamiques, Tables, Calculs visuels, Row-Level Security...**
- 3) Il ne faut pas les **confondre** aux **Calculation Groups** ni aux **Paramètres de Champs**. Les cas d'usage sont **différents** et **complémentaires**.
 - a. **DAX UDF**: **Centraliser** et réutiliser la même logique DAX 'partout'
 - b. **Calculation Groups**: Appliquer la même logique à toutes les **mesures visuellement disponibles**. Peut aussi servir de critères de filtres / axes d'analyses / colonnes calculées dynamiques.
 - c. **Paramètres de Champs**: Dynamiser **l'affichage** d'un ou plusieurs visuels (axes ou valeurs).
- 4) Ces DAX UDF sont **réexploitables** en Self-BI depuis Excel, Power BI (et autres).
- 5) Elles sont également **importables** et **réutilisables** via import de **daxlib.org** depuis Tabular Editor ou copier/coller TMDL dans Power BI Desktop

Ressources



Retrouvez les slides, fichiers .pbix et codes DAX de cette session sur mon [GitHub](#)



Retrouvez cette session sous format vidéo enregistrée sur ma chaîne **Youtube**