

# Veille sur la data

---

## 1. Types de données

### a. Données structurées

Description : Données organisées de manière rigide avec des champs définis.

Exemples : Tables SQL, feuilles de calcul Excel.

### b. Données semi-structurées

Description : Données qui n'ont pas une structure fixe mais contiennent des balises ou des marqueurs.

Exemples : JSON, XML.

### c. Données non structurées

Description : Données sans organisation prédéfinie.

Exemples : Textes, images, vidéos.

## 2. Formats de données : exemples

### CSV (Comma-Separated Values)

Utilisation : Stockage de données tabulaires.

Avantages : Simple et largement supporté.

### JSON (JavaScript Object Notation)

Utilisation : Échange de données entre serveurs et applications web.

Avantages : Lisible par les humains et facile à manipuler en Python.

### XML (eXtensible Markup Language)

Utilisation : Stockage et transport de données.

Avantages : Flexible et extensible.

### Parquet

Utilisation : Stockage de grandes quantités de données analytiques.

Avantages : Efficace en termes de stockage et de performance.

### 3. Où les trouver ?

#### Bases de données internes

Description : Données générées et stockées par l'entreprise.

#### APIs publiques

Description : Interfaces fournissant des données accessibles publiquement.

Exemples : APIs météo, APIs financières.

#### Sources ouvertes (Open Data)

Description : Données disponibles gratuitement pour le public.

Exemples : Données gouvernementales, datasets de recherche.

#### Fichiers locaux

Description : Données stockées sur des ordinateurs ou des serveurs internes.

Exemples : Fichiers CSV, JSON.

### 4. Bonnes pratiques : Comment les stocker ?

#### Choisir le bon type de base de données

SQL : Pour les données structurées avec relations complexes.

NoSQL : Pour les données semi-structurées ou non structurées.

#### Assurer la qualité des données

Validation : Vérifier l'exactitude et la cohérence des données avant le stockage.

Nettoyage : Éliminer les doublons et corriger les erreurs.

#### Sécuriser les données

Accès restreint : Limiter l'accès aux données sensibles.

Sauvegardes régulières : Prévenir la perte de données en cas de panne.

#### Optimiser le stockage

Indexation : Améliorer la vitesse des requêtes dans les bases de données.

Compression : Réduire l'espace de stockage nécessaire.

#### Documentation

Description des schémas : Documenter la structure des bases de données.

Processus de transformation : Enregistrer les étapes de transformation des données.