

Le format YAML

YAML, qui signifie "YAML Ain't Markup Language", est un format de données lisible par l'homme, largement utilisé pour la configuration des applications, la définition des déploiements d'infrastructure en tant que code, et d'autres tâches où les données sont à la fois lues par les machines et les humains.

Ce langage est extrêmement utilisé en DevOps : Docker, CI / CD, Ansible, Kubernetes etc utilisent tous le YAML.

Il est nécessaire de bien maîtriser le YAML avant de démarrer la formation.

La syntaxe YAML

La syntaxe YAML est conçue pour être simple et lisible par l'humain. Elle est moins verbeuse que le XML et plus descriptive que le CSV.

Elle est plus facile à lire que le JSON.

La syntaxe emprunte des éléments de langages de haut niveau, tels que l'utilisation du dièse (#) pour les commentaires et le tilde (~) pour représenter une valeur nulle.

Les concepts clés sont :

- **Indentation** : l'indentation (habituellement avec des espaces, non des tabulations) est utilisée pour indiquer les niveaux de hiérarchie des données.
- **Listes** : collections ordonnées d'éléments, et sont représentées en YAML en commençant chaque élément de la liste par un tiret (-).
- **Dictionnaires** : collection de paires de clés et de valeurs, et peut être représenté en YAML en utilisant des deux-points (:) pour séparer chaque clé de sa valeur.
- Les documents YAML peuvent être séparés dans un même fichier par trois traits d'union (---).

Commentaires

Les commentaires commencent par le signe dièse (#) et s'étendent jusqu'à la fin de la ligne.

```
# Ceci est un commentaire
```

Valeurs Null

Vous pouvez représenter une valeur null avec le tilde (~).

```
clé: ~ # Ceci signifie que la valeur est nulle
```

Tableaux (listes)

Les tableaux sont définis avec des tirets (-) suivi d'une espace ou dans des crochets [].

```
fruits:  
  - Pomme  
  - Orange  
  - Banane
```

```
légumes: [Courgette, Carrotte, 'Pomme de terre']
```

Dictionnaires (clé-valeur)

Les paires clé-valeur sont séparées par des deux-points (:), suivi d'une espace.

```
personne:  
  nom: Jean # string  
  âge: 30 # number
```

Texte Multiligne

YAML permet d'écrire du texte sur plusieurs lignes en utilisant le pipe (|) pour conserver les retours à la ligne, ou le chevron (>) pour les convertir en espaces.

```
description: |  
  Ceci est un texte  
  qui conserve les retours à la ligne.
```

```
description: >  
  Ceci est un texte  
  qui remplace les retours à la ligne par des espaces.
```

Références et alias

Vous pouvez réutiliser des nœuds en les marquant avec une esperluette (&) et en y faisant référence avec un astérisque (*).

```
étudiants:
  - &jean
    nom: Jean
    âge: 20
  - nom: Marie
    âge: 22
  - *jean # Ceci est une référence à l'étudiant "Jean"
```

Types de Données

YAML identifie automatiquement les types de données (comme les entiers, les flottants et les booléens) mais vous pouvez également les marquer explicitement.

```
un_entier: !!int 42
un_booléen: !!bool true
```

Indentation

L'indentation est utilisée pour représenter la structure imbriquée, et seuls les espaces sont autorisés pour l'indentation (les tabulations ne sont pas autorisées).

Documents multiples

Vous pouvez avoir plusieurs documents YAML dans un seul fichier, séparés par ----.

```
----
document1: valeur1
----
document2: valeur2
```

Exemple CI / CD

Voici un exemple de fichier de configuration YAML pour Github Actions :

```
name: Node.js CI

on:
  push:
    branches:
      - main

jobs:
  build:
    runs-on: ubuntu-latest

    strategy:
      matrix:
        node-version: [12.x, 14.x, 16.x]

    steps:
      - uses: actions/checkout@v4
      - name: Use Node.js ${{ matrix.node-version }}
        uses: actions/setup-node@v3
        with:
          node-version: ${{ matrix.node-version }}
      - run: npm install
      - run: npm test
```