

Référence rapide YAML

Scalaires, séquences, mappages, ancres, multi-ligne

Syntaxe

Règles de base

| | |
|----------------------------|---|
| Indentation | Espaces uniquement (pas de tabulations), profondeur cohérente |
| # | Commentaire (jusqu'à la fin de la ligne) |
| --- | Marqueur de début de document |
| ... | Marqueur de fin de document |
| Sensible à la casse | Les clés et valeurs distinguent majuscules et minuscules |

Exemple minimal

```
---
name: Alice
age: 30
active: true
```

Scalaires

Types de scalaires

| | |
|---------------------|--|
| hello | Chaîne (sans guillemets) |
| "hello" | Chaîne (guillemets doubles, permet les échappements) |
| 'hello' | Chaîne (guillemets simples, littérale) |
| 42 / 3.14 | Nombre (entier / décimal) |
| true / false | Booléen |
| null / ~ | Valeur nulle |
| 2026-03-26 | Date (ISO 8601) |

Règles de guillemets

```
plain: aucun guillemet nécessaire
special: "les deux-points : et le dièse # nécessitent des guillemets"
escape: "nouvelle\nligne"
literal: 'pas de \n d''échappement ici'
```

Séquences

Séquence en bloc

```
fruits:
- apple
- banana
- cherry
```

Séquence de flux (en ligne)

```
colors: [red, green, blue]
matrix: [[1, 2], [3, 4]]
```

Séquence d'objets

```
users:
- name: Alice
  role: admin
- name: Bob
  role: user
```

Mappages

Mappage en bloc

```
server:
  host: localhost
  port: 8080
  debug: false
```

Mappage de flux (en ligne)

```
point: {x: 10, y: 20}
```

Clés complexes

```
? [first, second]
: "valeur de clé composée"
```

Ancres et alias

Définir et réutiliser

```
defaults: &defaults
  timeout: 30
  retries: 3
production:
  <<: *defaults
  timeout: 60
```

Référence

| | |
|------------------|--|
| &name | Définir une ancre |
| *name | Référencer (alias) une ancre |
| << | Clé de fusion — fusionner un mappage dans l'actuel |

Chaînes multi-lignes

Scalaires en bloc

| | |
|----------------------------------|---|
| (littéral) | Préserve les sauts de ligne exactement |
| > (replié) | Replie les sauts de ligne en espaces |
| + (conserver) | Littéral, conserver les sauts de ligne finaux |
| - (supprimer) | Littéral, supprimer les sauts de ligne finaux |
| >- (replier+supprimer) | Replié, supprimer les sauts de ligne finaux |

Bloc littéral

```
script: |
echo "line one"
echo "line two"
```

Bloc replié

```
description: >
Ce long texte sera
replié en une seule ligne
avec des espaces.
```

Balises

Balises de type

| | |
|----------------------|-------------------------------------|
| !!str 42 | Forcer la valeur en chaîne "42" |
| !!int "42" | Forcer la valeur en entier 42 |
| !!float 1 | Forcer la valeur en décimal 1.0 |
| !!bool "true" | Forcer la valeur en booléen |
| !!null "" | Forcer la valeur en nul |
| !!binary | Données binaires encodées en Base64 |

Exemple de balise

```
port: !!str 8080
enabled: !!bool "yes"
version: !!float 3
```

Modèles courants

Docker Compose

```
services:
  web:
    image: nginx:alpine
    ports: ["80:80"]
    environment:
      NODE_ENV: production
```

GitHub Actions

```
on: [push, pull_request]
jobs:
  test:
    runs-on: ubuntu-latest
    steps:
      - uses: actions/checkout@v4
```

Conseils

| | |
|------------------------|---|
| yamllint | Valider YAML pour la syntaxe et le style |
| yq | Processeur YAML en ligne de commande (comme jq) |
| Éviter Norway | NO est false booléen — mettre les codes pays entre guillemets |
| Éviter !!python | Ne jamais charger du YAML non fiable avec des chargeurs non sécurisés |