

Ansible:

- **YAML**
 - **Langage descriptif**

clair
facile
humain
exécution lente (par rapport à JSON) / parsing

commence toujours par --- mais ok sans

les types de données:

Scalars (string et chiffres)

```
---
name: "Name"
premon: Leo
age: 18
```

Sequences (tableaux)

Ordre important (ordonné)

```
---
tableau:
  - "premier"
  - "deuxieme"
  - "troisieme"
```

format JSON dans YAML

```
tableau: ["premier", "deuxieme", "troisieme"]
```

Dictionary (Dictionnaire)

```
---
mysqlatabase:
  hostname: localhost
  username: root
  password: 1234
```

format JSON dans YAML

```
mysqlatabase: { hostname: localhost, username: root, password: 1234 }
```

Mapping (objets) / mélange des deux

C'est l'indentation qui définit les groupements des objets.

```
---
objet:
  prenom: Leo
  age: 18
  tableau:
    - "premier"
    - "deuxieme"
    - "troisieme"
    - quatrieme:
      uneCle: test
```

Commentaires

=====

```
---  
# Mon objet  
objet:  
  prenom: Leo  
  age: 32 # autre commentaire
```

Multi lignes

=====

Ecrire plusieurs lignes dans une meme commande

```
action:  
  script: |  
    echo "Espace disque" > /tmp/txt.log  
    df -Th >> /tmp/txt.log
```

Ecrire une ligne de commande longue

```
action:  
  script: >  
    ssh -p 22 root@192.168.0.22  
    "df -Th >> /tmp/txt.log"
```

```
# Et équivalent à  
  ssh -p 22 root@192.168.0.22 "df -Th >> /tmp/txt.log"
```

Ancres

=====

```
# C'est le '&' qui définit l'encre
# Et pour lier une autre variable à l'encre on utilise '*'
passwd1: 1234 &motdepasse
passwd2: *motdepasse

# Et équivalent à
passwd1: 1234
passwd2: 1234
```

Ancrages multilignes

```
dev-template: &env
  var:
    dir: /usr
  script: |
    echo "Espace disque" > /tmp/txt.log
    df -Th >> /tmp/txt.log

prod:
  <<: *env

dev:
  <<: *env
  dir: /srv
```

Est équivalent à

```
prod:
  var:
    dir: /usr
  script: |
    echo "Espace disque" > /tmp/txt.log
    df -Th >> /tmp/txt.log

dev:
  var:
    dir: /srv
  script: |
    echo "Espace disque" > /tmp/txt.log
    df -Th >> /tmp/txt.log
```

```
# yaml linter pour tester le code
https://yamlchecker.com
https://yamlvalidator.com # + json
```