

# Riferimento Rapido YAML

Scalar, sequenze, mapping, ancore, multi-riga

## Sintassi

### Regole di Base

<b>Indentation</b>	Solo spazi (nessun tab), profondità consistente
<b>#</b>	Commento (fino a fine riga)
<b>---</b>	Marcatore di inizio documento
<b>...</b>	Marcatore di fine documento
<b>Case sensitive</b>	Chiavi e valori distinguono le maiuscole

### Esempio Minimale

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

## Scalar

### Tipi Scalar

<b>hello</b>	Stringa (senza virgolette)
<b>"hello"</b>	Stringa (doppi apici, permette escape)
<b>'hello'</b>	Stringa (apice singolo, letterale)
<b>42 / 3.14</b>	Numero (intero / virgola mobile)
<b>true / false</b>	Booleano
<b>null / ~</b>	Valore null
<b>2026-03-26</b>	Data (ISO 8601)

### Regole di Virgolettatura

plain: nessuna virgoletta necessaria  
special: "il colon: e il cancelletto # richiedono virgolette"  
escape: "line\nnewline"  
literal: 'nessun escape \n qui'

## Sequenze

### Sequenza a Blocchi

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

### Sequenza Flow (Inline)

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

### Sequenza di Oggetti

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

## Mapping

### Mapping a Blocchi

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

### Mapping Flow (Inline)

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

### Chiavi Complesse

```
? [first, second]
: "valore chiave composta"
```

## Ancore e Alias

### Definizione e Riutilizzo

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

### Riferimento

<b>&amp;name</b>	Definisce un'ancora
<b>*name</b>	Riferimento (alias) a un'ancora
<b>&lt;&lt;</b>	Chiave merge — fonde il mapping nel corrente

## Stringhe Multi-riga

### Scalar a Blocchi

<b>  (letterale)</b>	Preserva i newline esattamente
<b>&gt; (ripiegato)</b>	Ripiega i newline in spazi
<b> + (conserva)</b>	Letterale, conserva newline finali
<b> - (elimina)</b>	Letterale, elimina newline finali
<b>&gt;- (piega+elimina)</b>	Ripiegato, elimina newline finali

### Blocco Letterale

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

### Blocco Ripiegato

```
description: >
  Questo lungo testo verrà
  ripiegato in una singola riga
  con spazi.
```

## Tag

### Tag di Tipo

<b>!!str 42</b>	Forza il valore a stringa "42"
<b>!!int "42"</b>	Forza il valore a intero 42
<b>!!float 1</b>	Forza il valore a float 1.0
<b>!!bool "true"</b>	Forza il valore a booleano
<b>!!null ""</b>	Forza il valore a null
<b>!!binary</b>	Dati binari codificati in base64

### Esempio di Tag

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

## Pattern Comuni

### 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
```

## Suggerimenti

<b>yamllint</b>	Linta YAML per sintassi e stile
<b>yq</b>	Processore YAML da riga di comando (come jq)
<b>Avoid Norway</b>	NO è booleano false — metti tra virgolette i codici paese
<b>Avoid !!python</b>	Non caricare mai YAML non attendibile con loader non sicuri