

YAML

06:46 AM

YAML Syntax

This page provides a basic overview of correct YAML syntax, which is how Ansible playbooks (our configuration management language) are expressed.

We use YAML because it is easier for humans to read and write than other common data formats like XML or JSON. Further, there are libraries available in most programming languages for working with YAML.

You may also wish to read Working with playbooks at the same time to see how this is used in practice.

1) Every YAML starts with `---` which denotes the start of a YAML file.

YAML Basics

For Ansible, nearly every YAML file starts with a list. Each item in the list is a list of key/value pairs, commonly called a "hash" or a "dictionary". So, we need to know how to write lists and dictionaries in YAML:

There's another small quirk to YAML. All YAML files (regardless of their association with Ansible or not) can optionally begin with `---` and end with `...`. This is part of the YAML format and indicates the start and end of a document.

All members of a list are lines beginning at the same indentation level starting with a "-" (a dash and a space):

```
---
# A list of tasty fruits
- Apple
- Orange
- Strawberry
- Mango
...
```

→ comment
Simple

A dictionary is represented in a simple key: value form (the colon must be followed by a space):

```
# An employee record
martin:
  name: Martin D'vloper
  job: Developer
  skill: Elite
```

More complicated data structures are possible, such as lists of dictionaries, dictionaries whose values are lists or a mix of both:

```
# Employee records
- martin:
  name: Martin D'vloper
  job: Developer
  skills:
    - python
    - perl
    - pascal
- tabitha:
  name: Tabitha Bitumen
  job: Developer
  skills:
    - lisp
    - fortran
    - erlang
```

Dictionaries and lists can also be represented in an abbreviated form if you really want to:

```
---
martin: {name: Martin D'vloper, job: Developer, skill: Elite}
['Apple', 'Orange', 'Strawberry', 'Mango']
```

Structure
SQ

1) Single

2) Low

3) db se

4) No E

5) No Su

6) return

```
{ pin: 1234,
  location: Berlin,
  city: f }
```

```
{
  . . .
}
```

```
key: value
key: value
```

These are called "Flow collections".

Ansible doesn't really use these too much, but you can also specify a boolean value (true/false) in several forms:

```
create_key: yes
needs_agent: no
knows_oop: True
likes_emacs: TRUE
uses_cvs: false
```

Use lowercase 'true' or 'false' for boolean values in dictionaries if you want to be compatible with default yamllint options.

Values can span multiple lines using | or >.

Spanning multiple lines using a "Literal Block Scalar" | will include the newlines and any trailing spaces.

Using a "Folded Block Scalar" > will fold newlines to spaces;

it's used to make what would otherwise be a very long line easier to read and edit. In either case the indentation will be ignored. Examples are:

```
include_newlines: |
    exactly as you see
    will appear these three
    lines of poetry
```

```
fold_newlines: >
    this is really a
    single line of text
    despite appearances
```

While in the above > example all newlines are folded into spaces, there are two ways to enforce a newline to be kept:

```
fold_some_newlines: >
a
b

c
d
e
f
same_as: "a b\nc d\n e\nf\n"
```


Let's combine what we learned so far in an arbitrary YAML example. This really has nothing to do with Ansible, but will give you a feel for the format:

```
---
# An employee record
name: Martin D'vloper
job: Developer
skill: Elite
employed: True
foods:
- Apple
- Orange
- Strawberry
- Mango
languages:
perl: Elite
python: Elite
pascal: Lame
education: |
4 GCSEs
3 A-Levels
BSc in the Internet of Things
```

That's all you really need to know about YAML to start writing Ansible playbooks.

list -
- ~~key~~
- ~~key~~
- ~~key~~
dict -

list -
- ~~key~~
- ~~key~~
- ~~key~~
dict -



Basic rules.

- Data elements at the same level in the hierarchy (such as items in the same list) must have the same indentation.
- Items that are children of another item must be indented more than their parents. You can also add blank lines for readability.

Important:

Only the space character can be used for indentation; tab characters are not

<https://yaml-online-parser.appspot.com/>

