

# ANSIBLE

Google™ Custom Search

Introduction

Quickstart Video

Playbooks

Playbooks: Special Topics

About Modules

Module Index

Detailed Guides

Developer Information

Ansible Tower

Community Information & Contributing

Ansible Galaxy

Testing Strategies

Frequently Asked Questions

Glossary

YAML Syntax

YAML Basics

Gotchas

A

TOWER

FREE 30-day Trial



# ANSIBLE TOWER

The Official UI of ANSIBLE

A

#ANSIBLEFEST

N

Y

C

2015

[Docs](#) » [YAML Syntax](#)

[Edit on GitHub](#)

## 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 [Playbooks](#) at the same time to see how this is used in practice.

## 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) should begin with `---`. This is part of the YAML format and indicates the start 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
```

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

```
---
# An employee record
name: Example Developer
job: Developer
skill: Elite
```

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

```
---
# An employee record
{name: Example Developer, job: Developer, skill: Elite}
```

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
```

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: Example Developer
job: Developer
skill: Elite
employed: True
foods:
  - Apple
  - Orange
  - Strawberry
  - Mango
languages:
  ruby: Elite
  python: Elite
  dotnet: Lame
```

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

## Gotchas

While YAML is generally friendly, the following is going to result in a YAML syntax error:

```
foo: somebody said I should put a colon here: so I did
```

You will want to quote any hash values using colons, like so:

```
foo: "somebody said I should put a colon here: so I did"
```

And then the colon will be preserved.

Further, Ansible uses “`{{ var }}`” for variables. If a value after a colon starts with a “`{`”, YAML will think it is a dictionary, so you must quote it, like so:

```
foo: "{{ variable }}"
```

## ❗ See also

### **Playbooks**

Learn what playbooks can do and how to write/run them.

### **YAMLLint**

YAML Lint (online) helps you debug YAML syntax if you are having problems

### **Github examples directory**

Complete playbook files from the github project source

### **Mailing List**

Questions? Help? Ideas? Stop by the list on Google Groups

### **irc.freenode.net**

#ansible IRC chat channel

[← Previous](#)

---

© Copyright 2015 [Ansible, Inc.](#). Last updated on May 01, 2015.

Ansible docs are generated from [GitHub sources](#) using [Sphinx](#) using a theme provided by [Read the Docs](#).