Yaml Cheat Sheet +

Download via https://alta3.com/posters/yaml.pdf

Type / Style	YAML Examples	Example Description

Start yaml document →
Start of YAML Doc →
Key: value pair →
Key: value pair →
Key: value pair →
	`---`
	`  game: tetris`
	`  type: puzzle`
	`  levels: ten`
		Simple key value pairs in Yaml (Dictionary)

Start yaml document→
Key:→
Value of employee→
Value of employee→
Value of employee→
Value of employee→
	`---`
	`employees:`
	`  - Michael`
	`  - Jim`
	`  - Pam`
	`  - Creed`
		List of Dictionaries

Start yaml document →
Start of list of lists →
Item 1 in list time →
Item 2 in list time →
Item 1 in list time.of →
Item 2 in list time.of →
Item 3 in list time →
Item 1 in list time.many →
	`---`
	`- time`
	`  - Is`
	`  - of`
	`    - the`
	`    - essence`
	`  - many`
	`    - quotes`
		List of Lists

Start yaml document →
start playbook tasks →
Ansible comment →
Module name (template) →
Parameter src →
Parameter dest →
Parameter owner →
Parameter group →
Parameter mode →
Task modifier →
First vars set in list →
Second vars set in list →
Register as a variable →
Run as root (True==yes) →
	`---`
	`tasks:`
	`  - name: create file from template`
	`    template:`
	`      src: "{{ item.src }}"`
	`      dest: "{{ item.dest }}"`
	`      owner: goku`
	`      group: saiyan`
	`      mode: 0755`
	`    with_items:`
	`      - {src: 'nvic1.j2', dest: '/tmp/namek1.txt'}`
	`      - {src: 'nvic2.j2', dest: '/tmp/namek2.txt'}`
	`    register: namek_visa_id_card`
	`    become: yes`
		Ansible tasks are a specific dictionary of dictionaries which represents the parameters passed to an ansible module
		
		This task, using the template module, which loops over src/dest files

Start yaml document →
Task description ansible →
Module name (get_url) →
Parameter url file to dl →
Parameter place to save →
Parameter rwx permission →
Ansible option argument →
Param chdir before exec →
Change Directory →
	`---`
	`- name: download file from website`
	`  get_url:`
	`    url: "http://ilvcartoons.watch.com/ctn.tar.gz"`
	`    dest: /home/professorX/ctn.tar.gz`
	`    mode: '0644'`
	`  args:`
	`    chdir: "/home/professorX"`
		Ansible specific Dictionary of dictionaries with dictionary of dictionary
		
		Single task, get url set permissions, and change dir beforehand

Helpful Links	Purpose
http://www.yamllint.com/	Check Syntax of YAML
https://codebeautify.org/yaml-validator	Check Syntax of YAML
http://yaml.org/spec/1.2/spec.html	Extended YAML Documentation
https://github.com/Animosity/CraftIRC/wiki/Complete-idiot%27s-introduction-to-yaml	Introduction to YAML
http://ansible-docs.readthedocs.io/zh/stable-2.0/rst/YAMLSyntax.html	YAML with Ansible
https://learnxinyminutes.com/docs/yaml/	Quick YAML Examples

Facts
Tab characters are not to be used
YAML is cAsE sensitive
End your YAML file with the .yaml or .yml extension
All members of a list are lines beginning at the same indentation level starting with a "- " (a dash and a space)
Three dashes --- is optional in starting a yaml document
Three periods … is optional in ending a yaml document
Use a monospaced font such as `courier new` when writing yaml documents
Ansible playbooks are YAML files
YAML is a superset of JSON

Terms
sequences (arrays / lists) starts with a dash – then a space for each line
scalars (strings / numbers)
mappings (hashes / dictionaries) is a key: value form (the colon must be followed by a space)

Need Telecom or IT Training?
sales@alta3.com \|\| +1-717-566-4428

Visit https://alta3.com/posters for more
Alta3 Posters & Cheat Sheets

Jinja2 Cheat Sheet BY MOSHE ZADKA

Jinja2 is a templating language for Python that is both web framework agnostic and language agnostic.

Language

Operation	Example	Use
`{{ <variable> }}`	`{{ name }}`	Interpolate variable

`{% for <var> in <list> %}`
`... {% endfor %}`
	`{% for x in range(3) %}`
	`{{ x }} {% endfor %}`
		Loop over a list

`{% if <condition> %}`
`{% else %}`
`{% endif %}`
	`{% if x is even %} It is even`
	`{% else %} It is odd`
	`{% endif %}`
		Test for a condition to determine what to render

`{{ <var>.<attribute> }}`
or
`{{ <var>["<attribute>"] }}`
	`{{ user.name }}`
	or
	`{{ user["name"] }}`
		Access attribute or element

`{{ loop.<attr> }}`	`{{ loop.index }}`	Attribute of loop object
`{{ <something> | <filter> }}`	`{{ loop.index | filter }}`	Apply a filter to an expression
`{{ <foo> ~ <bar> }}`	`{{ name ~ " " ~ surname }}`	Concatenate

`{{% set var = expression % }}`	`{% set var = name ~"`
	`"~ surname %}`
		Set variable to be used later

Loop object

`loop.index` -- 1-based index

`loop.index0` -- 0-based index

`loop.first` -- is it the first iteration

`loop.last` -- is it the last iteration

Filters

`tojson` -- convert to JSON

`escape` -- HTML escape

`indent` -- indent string

Tests

`even/odd` -- check if it is even or odd (respectively)

`defined` -- variable is defined

`divisibleby(n)` -- divisible by some number

none -- is None