

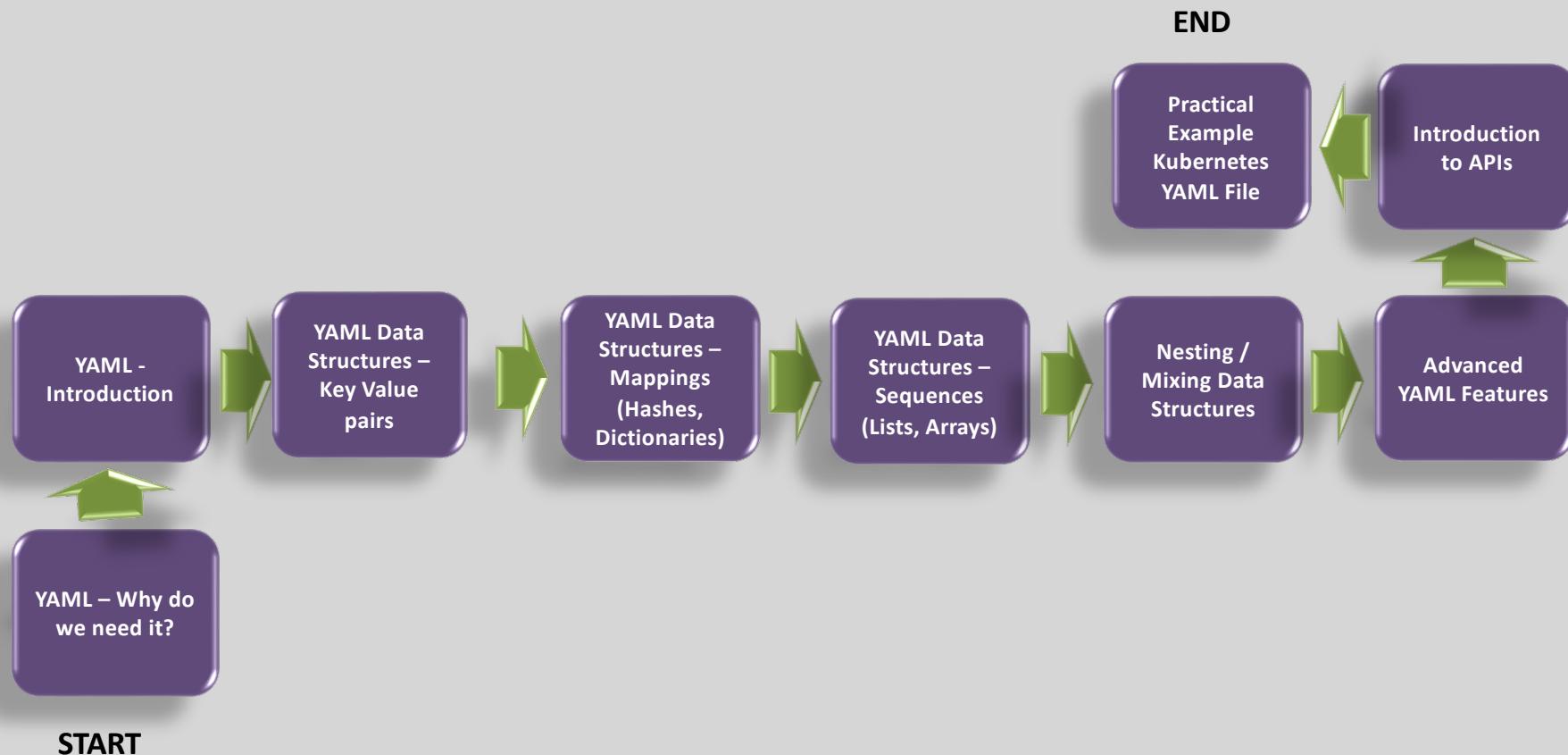
About Your Instructor

- Eissa Abousherif
 - 28 years of IT experience
 - Certified Internetwork Expert and AWS Solutions Architect
 - 17 years working as technical Author and trainer
- You are in the best possible hands to learn



Course Introduction

Content Flow – A Logical Progression – Scaffolding Approach



Audience – Who Is This Course For?

- Multi-Cloud and DevOps career aspirants
- Developers
- SysOps Administrators
- DevOps engineers
- Anyone interested in learning about serialization languages

Pre-requisites

A basic understanding about IT, Cloud, and Development is nice to have but not required.



Why YAML?

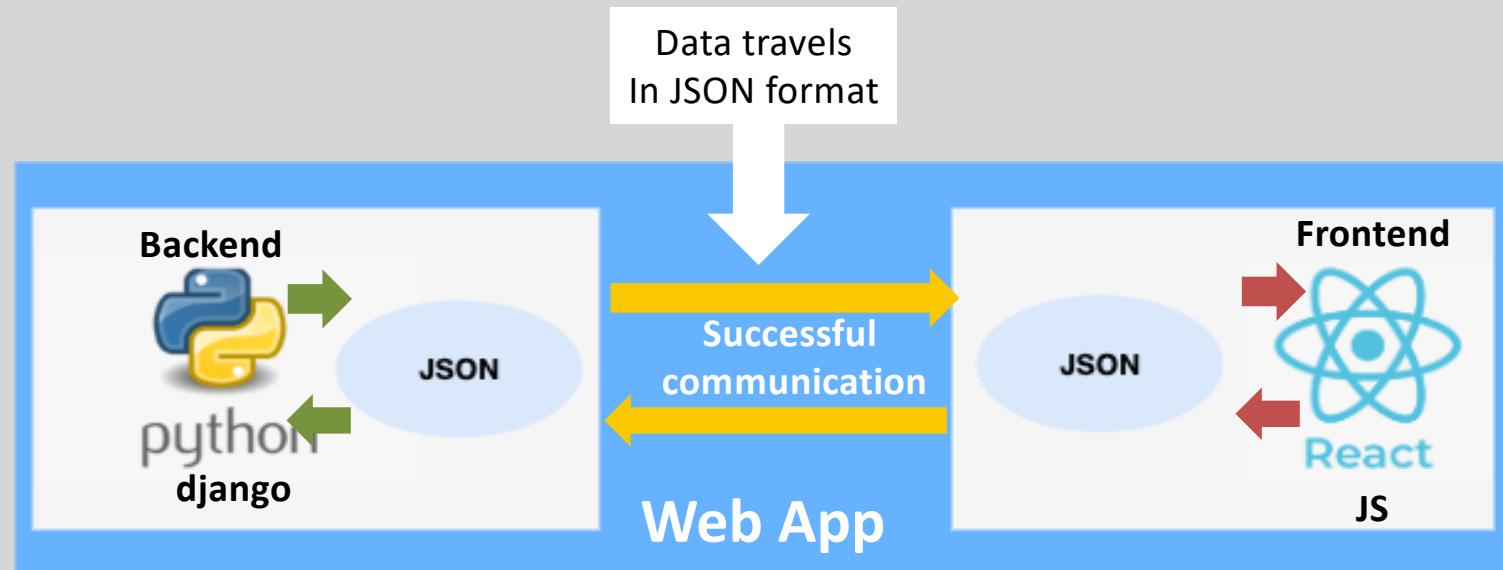
YAML (YAML Ain't a Markup Language) – Why do we need it?

- A fact: There are several programming languages, but only a handful of languages for storing and transferring data.
- Applications need a standard, agreed, language/syntax to communicate.
 - Therefore, a data serialization language is required.
- eXtensible Markup Language (XML), JavaScript Object Notation (JSON), and YAML Ain't Markup Language (YAML) are commonly used data serialization (data representation) languages.



YAML (YAML Ain't a Markup Language) – Why do we need it?

The two application components (tiers) of the web app can now communicate successfully using the data serialization language of choice, JSON in this case.



YAML – Why YAML and not JSON or XML?

Which one seems to be the easiest to deal with?

```
<Servers>
  <Server>
    <hostname>App-1</hostname>
    <manufacturer>HP</manufacturer>
    <tier>Backend</tier>
  </Server>
</Servers>
```

XML

- Tree Structure
- 1996
- Slow
- Complex to read
- .xml

```
{
  Servers: [
    {
      hostname: App-1,
      manufacturer: HP,
      tier: Backend
    }
  ]
}
```

JSON

- Map Structure
- 2002
- Fast
- Easy to read
- .json

```
Servers:
  hostname: App-1
  manufacturer: HP
  tier: Backend
```

YAML

- Map Structure
- 2006
- Fast
- Easiest to read
- .yaml or .yml

YAML (YAML Ain't Markup Language)

- YAML is a digestible, human readable, serialization language that is used to create configuration files and is supported by most popular programming languages (if not all).
- YAML is designed for easier human interaction.
- YAML is widely used in many current DevOps tools including Kubernetes, Ansible, Docker, Prometheus, AWS CloudFormation, and many others.



ANSIBLE



docker



kubernetes



Prometheus



YAML & DevOps Career



ANSIBLE



docker



kubernetes



Prometheus



CHEF



puppet



YAML

Introduction

YAML - Use Cases

YAML works well in any of the following:

- Configuration files,
- Log files,
- Inter-process messaging,
- Cross-language data sharing,
- Object persistence and
- Debugging of complex data structures.

YAML

- YAML uses Unicode printable characters.
- Some characters provide structural information, and the rest is the data itself.
- YAML minimizes the amount of structural characters, which makes the data easily readable.

```
---
person:
# this is alice john's info
  name: alice john
  age: 5
  gender: 'female'
  height: 3.5
  fav_color: "red"
  employed: false
  birthday: 2017-02-27 #YYYY-MM-DD
```

YAML - Data Structures

The three primitive/basic data structures in YAML:

- Scalars (strings, numbers (integers and floats), Boolean)
- Mappings (also called Hashes or **dictionaries**)
- Sequences (also called **arrays or lists**)

To form a complete data serialization language for any native data structure, YAML leverages the above three structures and adds:

- A simple typing system and
- Aliasing mechanism

YAML – Structure - Comments

Comments are not rendered or processed. They can be used in separate lines or following data

```
# this is alice john's info
name: alice john
age: 5
gender: 'female'
height: 3.5
fav_color: "red"
employed: false
birthday: 2017-02-27 # YYYY-MM-DD
```

YAML – Structure & Rules – Start and End of Documents

- YAML uses three dashes (“---”) as the document marker line to separate directives from document content.
- The three dashes (“---”) signal the start of a document.
- Three dots (“...”) indicate the end of a document.
- You can have more than one document in a stream.
- This may or may not be required based on the tool you are using the YAML files with.

```
---
```

```
time: 01:13:10
```

```
actor: Dwayne Johnson
```

```
action: score
```

```
...
```

Single document stream



YAML

Online Tools

YAML – Online Validator Tool

<https://onlineyamltools.com/validate-yaml>

It highlights errors when spotted.

The screenshot shows a web interface for validating YAML code. On the left, under the heading 'yaml', is the following code:

```
yaml
class:
- studentName: Wrishika Ghosh
```

On the right, under the heading 'errors', is a green box containing the message:

No errors!
This YAML code is perfectly valid.

The screenshot shows a web interface for validating YAML code. On the left, under the heading 'yaml', is the following code:

```
yaml
class: I
- studentName: Wrishika Ghosh
```

On the right, under the heading 'errors', is a red box containing the message:

✗ Invalid YAML
Parse error below.

YAMLError: end of the stream or a document separator is expected at
line 2, column 1:
- studentName: Wrishika Ghosh

YAML – Online YAML Viewer

<https://jsonformatter.org/yaml-viewer>

View the output of a valid YAML file

{JSON formatter}

JSON BEAUTIFIER JSON PARSER XML FORMATTER

YAML Viewer

Sample

```
1 ---  
2 student: amit gupta  
3 grade:  
4   americanschoo: yes  
5 age:  
6   - hello world
```

Load Data 

YAML Tree

object ► age ►

object {3}
student : amit gupta
grade {1}
 americanschoo : yes
age [1]
 0 : hello world

YAML Viewer

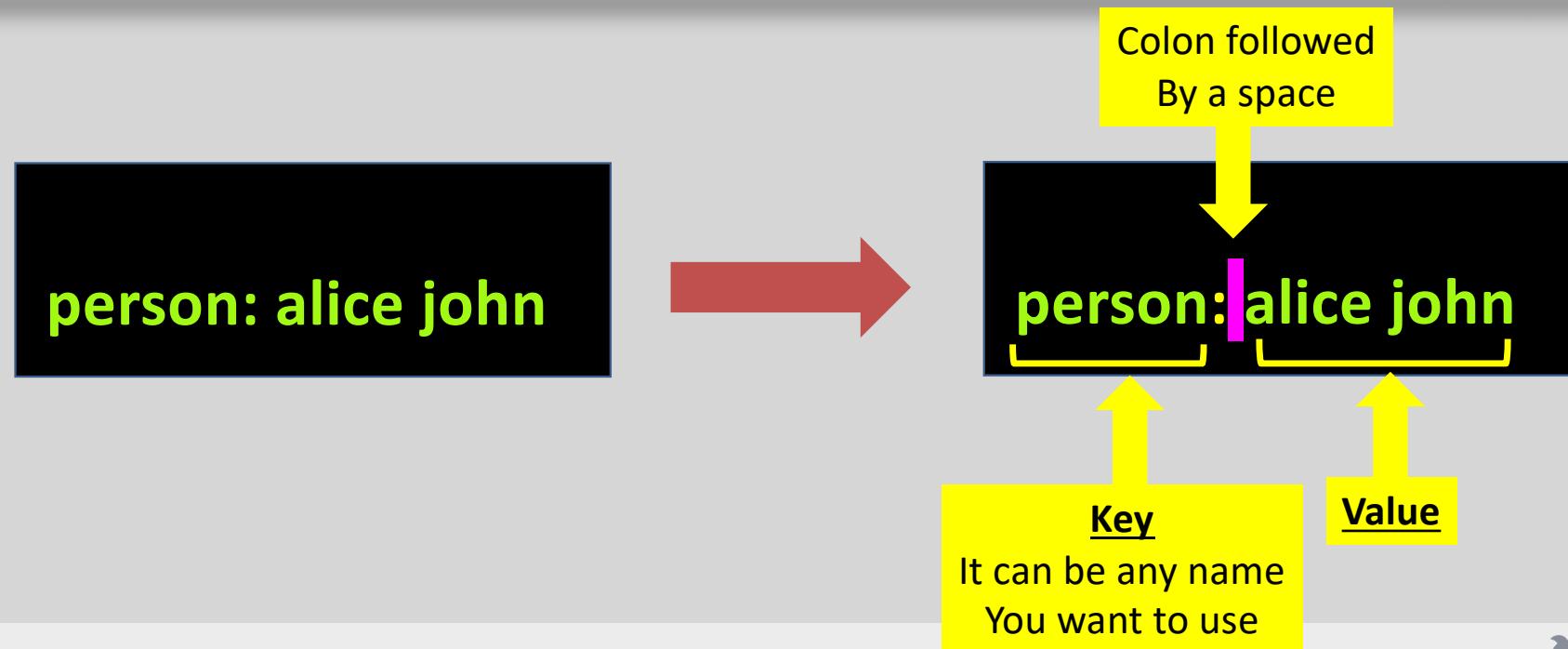


YAML Data Structures

Key/Value pairs

YAML Block Collections : Key/Value Pairs

- Data in YAML is represented as key/value pairs.
- We can set the value directly to the key, or through sequences (also called lists or arrays) and mappings (also called hashes or dictionaries) as we will see later.



YAML – Words in Keys - Separation

Words in keys can be separated by underscores (most popular), dashes, or spaces

```
Student name: John Smith  
Student-name: John Smith  
Student_name: John Smith
```

YAML – Structure & Rules – Colon and Space “: “

A colon followed by a space (“: “) is used to separate the key from the value in key/value pairs.

```
# this is alice john's info  
name: alice john  
birthday: 2017-02-27 #YYYY-MM-DD
```

One space must be used after
each colon of a key/value pair

YAML Block Collections : Key/Value Pairs (cont.)

A value (to the right from the colon (:)) can be:

- A string e.g., “alice john”
- An integer,e.g., 5
- A float e.g., 50.5
- A Boolean e.g., true, false, on, off, yes, no
- A date e.g., YYYY-MM-DD HH:MM:SS
- Null

```
name: alice john
age: 5
weight: 50.5
female: true
birthday: 200-10-27
flaws: null
```

Value

YAML – Different Key/Value Pairs Examples

All of these are valid YAML key/value pairs

'key with quotations': 'value in quotation'

23: "An integer key with a string value"

'a boolean value': true

key including three spaces: 3

a null value: null

YAML – Strings & Quotes

We can use double quotes, single quotes or none around a string value of a key.

- However, when a string contains special characters (: {}[], & * # ? | - < > = ! % @ '), it needs to be escaped by either single- or double-quotes.

The same
In YAML

```
student_name: Alice John  
  
student_name: "Alice John"  
  
student_name: 'Alice John'
```

```
student_name: "Alice John \n"
```

Use the double quotes
Because of the control character \n

YAML Block Collections : URLs in YAML

To write a URL as a value of a key, better use single quotes to escape the ':' following http or https

```
webpage: 'http://www.dolfined.com'
```



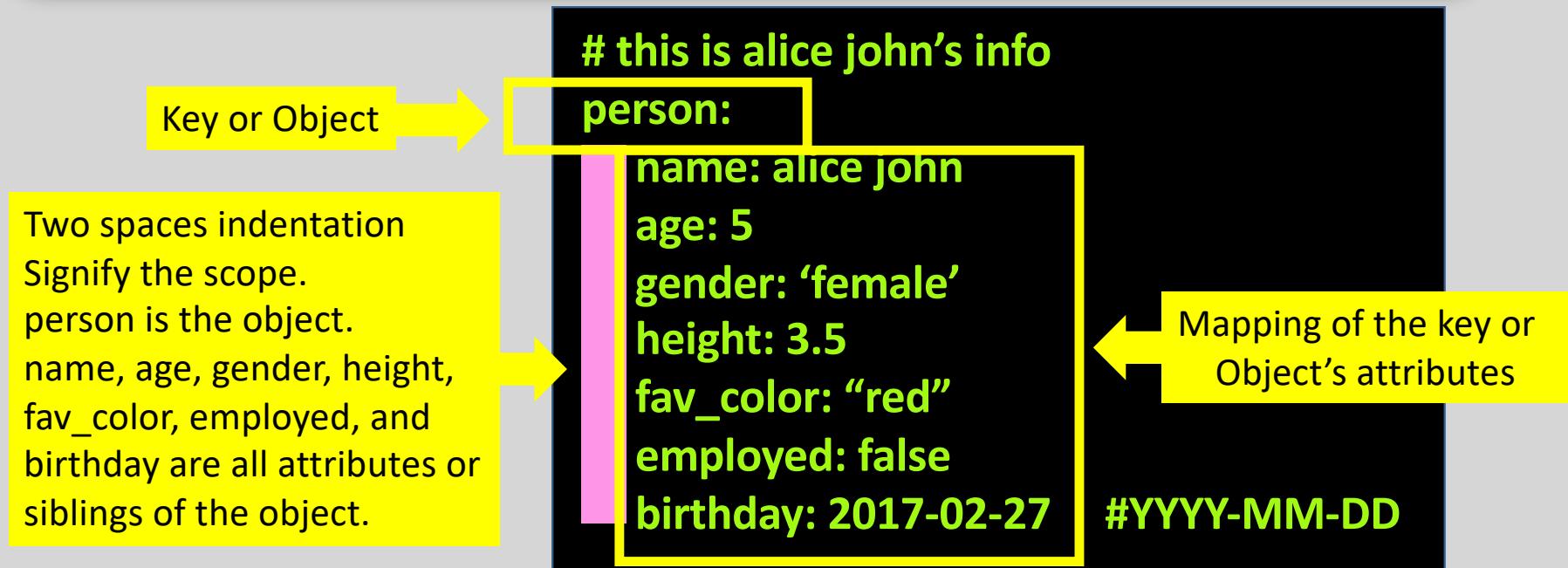
YAML Data Structures

Dictionaries / Mappings /
Hashes

YAML Block Collections : Mappings/Hashes/Dictionaries

You can use a dictionary/hash/mapping to assign multiple values to a key:

- YAML uses indentation for scope.
- Begin each entry on its own line.



YAML Block Collections : Use of Indentation - Spaces

Equal number of spaces in indentation is key to represent the same level of attributes (scope)

- Siblings of a key or an object must have an equal number of indented spaces.

```
# this is alice john's info
person:
  name: alice john
  age: 5
  gender: male'
  height: 5.5
  fav_color: "red"
  employed: false
  birthday: 2017-02-27
```

```
# this is alice john's info
person:
  name: alice john
  age: 5
  gender: male'
  height: 5.5
  fav_color: "red"
  employed: false
  birthday: 2017-02-27
```

YAML Block Collections : Mappings / Dictionaries / Hashes

A mapping uses a colon and space (" : ") to mark each key/value pair.

- Mappings are un-ordered collection of key/value pairs.

person:

name: alice john

age: 5

gender: 'female'

person:

name: alice john

gender: 'female'

age: 5

The two are the same
Order does not matter

YAML Block Collections : Mappings / Dictionaries / Hashes – A Different Way

We can use curly brackets (“{}”) to represent mappings.

```
person:  
  name: alice john  
  age: 5  
  gender: 'female'
```

```
person: {name: "alice john", gender: "female", age: "5"}
```



YAML Data Structures

Lists / Sequences / Arrays

YAML Block Collections : Sequences / Lists / Arrays

Sequences/Arrays/Lists of items indicate each entry with a dash and space (“- ”).

Two spaces indentation signify the scope.
The hyphen or dash defines it is a key/value pair in a list.

Hyphen (dash)
Followed by a space

students:
- Mohamed Salah
- Amit Gupta
- John Smith

YAML Block Collections : Sequences / Lists / Arrays

The hyphens of the list items can be indented or not, both will work

```
car_parts:  
  - tires  
  - engine  
  - gas tank
```

```
car_parts:  
  - tires  
  - engine  
  - gas tank
```

YAML Block Collections : Sequences / Lists / Arrays

Items or line in a sequence/array/list can be a scalar or it can also be a key/value pair

person:

- name: alice john
- age: 5
- gender: 'female'

car_parts:

- tires
- engine
- gas tank

YAML Block Collections : Sequences / Lists / Arrays – Different Ways

An array can be represented by the square brackets []

```
car_parts:  
  - tires  
  - engine  
  - gas tank
```

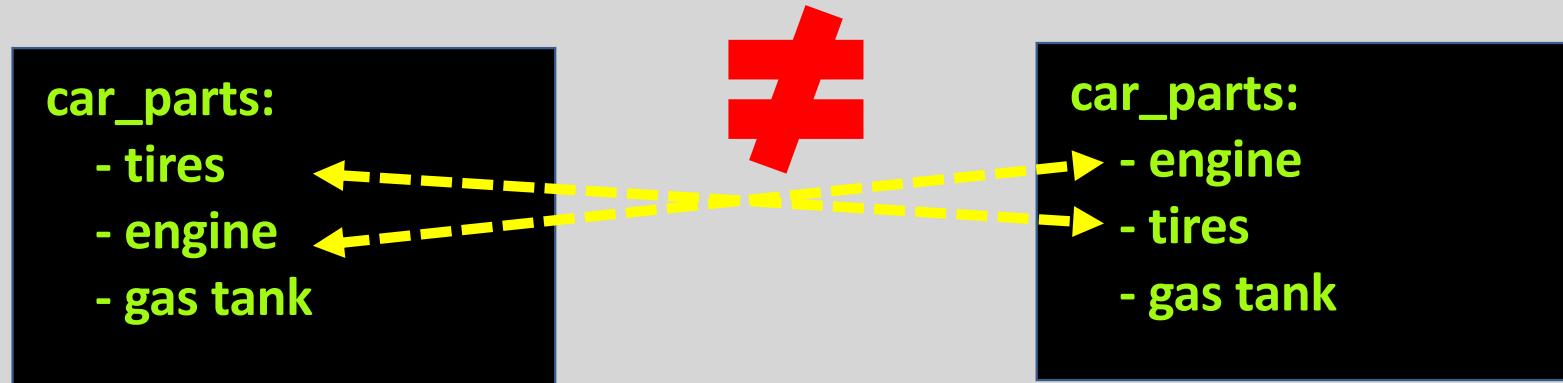
```
car_parts: ["tires", "engine", "gas tank"]
```

```
car_speeds: [80.0, 50.2, 120]
```

```
download_urls: ['http://www.dolfined.com/gzip/', 'ftp://ftp.dolfined.com/gzip/']
```

YAML Block Collections : Sequences / Lists / Arrays

Lists/Sequences/Array is an ordered collection of key value pairs.



The two are NOT the same
Order does matter

YAML Block Collections : Mappings / Dictionaries / Hashes

You can either set a direct value to a key or use a list or map but not both.

```
person: alice john
```

```
person:  
  name: alice john  
  gender: 'female'
```

```
person: alice john  
  gender: 'female'  
  age: 5
```

```
person:  
  - alice john  
  - 'female'
```

```
person: alice john  
  - 'female'  
  - tall
```

Setting a direct value
To a key

Setting a map or a list
Key/value pairs to a key

You cannot do both to a key



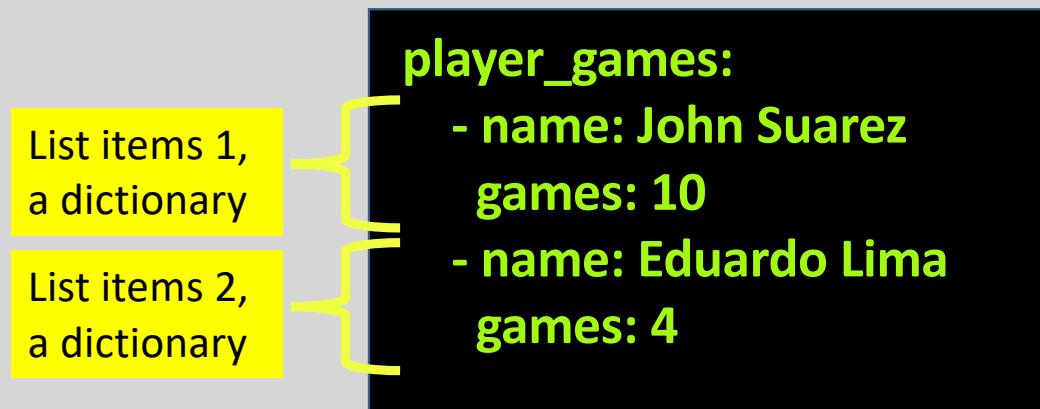
YAML Data Structures

Nesting/Mixing Structures

YAML – A Sequence (List/Array) of Mappings (Dictionaries)

A list item can be a mapping (dictionary) or key/value pairs

- The hyphen indicates this is a list item
- A dictionary list item can have multiple key/value pairs



YAML – A Sequence (List/Array) of Mappings (Dictionaries)

A different way to use the list item preceding hyphen or dash

```
player_games:  
  - name: John Suarez  
    games: 10  
  - name: Eduardo Lima  
    games: 4
```

List items 1,
a dictionary

List items 2,
a dictionary

YAML – A Sequence (List/Array) of Mappings (Dictionaries)

A list item can be a mapping (dictionary) or key/value pairs

- Each dictionary items can have any order
- The list items' order is important

```
player_games:  
  - name: John Suarez  
    games: 10  
  - name: Eduardo Lima  
    games: 4
```

```
player_games:  
  - games: 10  
  - name: John Suarez  
  - name: Eduardo Lima  
    games: 4
```

The two are the same
Order does not matter within a dictionary

YAML – A Sequence (List/Array) Nested in a Mapping (Dictionary)

We can nest or use a list as a mapping value

- Also, we can represent the list using square brackets to write the list inside the dictionary or mapping item

```
player_games:  
  name: John Suarez  
  games_played: 100  
  years_played:  
    - 1998  
    - 2002  
    - 2010
```

```
player_games:  
  name: John Suarez  
  games_played: 100  
  years_played: [1998, 2002, 2010]
```

YAML – Sequence of Sequences - Nested Sequences (List/Array)

An array item can be a list itself, i.e we can nest a list/array inside another list's item(s)

```
player_games:  
  - name: John Suarez  
  - games_played: 100  
  - injuries:  
    - knee  
    - shoulder  
    - shin
```

```
player_games:  
  - name: John Suarez  
  - games_played: 100  
  - injuries: [knee, shoulder, shin]
```

YAML – Nested Dictionaries

A dictionary item can be a dictionary itself

```
player_games:  
  name: John Suarez  
  games_played: 100  
  injury_years:  
    knee: 1998  
    shoulder: 2002  
    shin: 2005
```

```
player_games:  
  name: John Suarez  
  games_played: 100  
  injury_years: {knee: "1998", shoulder: "2002", shin: "2005"}
```

YAML – Mixed Structures

We can mix structures or use them alternatively sometimes

- We can use lists as keys using square brackets

Inline mapping/dictionary

```
- {name: Prashant, age: 18}  
- name: Shiva  
  age: 20  
- [name, age]: [Neeraj, 14]
```

List as a key, list as a value



YAML

Advanced Features

YAML – Multi-line Strings

Greater than (“>”) after a key and a colon means the new lines in the string scalar the follows become spaces.

Pipe (“|”) after a colon of a key means new lines in folded strings will be preserved.

```
name: John Smith  
accomplishment: >  
John was the president of  
kids' world bank from  
1992 through 2020.
```

Lines will be converted to spaces

```
name: John Smith  
accomplishment: |  
John was the president of  
kids' world bank from  
1992 through 2020.
```

Lines/formatting will be preserved

YAML – Multi-line Strings – Passing a Bash Script in a YAML Key/Value Pair

We can use the Pipe (“|”) in a list item to pass a bash (or shell) script

- Formatting will be preserved
- Mind the indentation requirements.

```
bash:  
- |  
  #!/bin/bash  
  yum update -y  
  yum install httpd -y  
  systemctl start httpd service  
  echo " This is server *2* " > /var/www/html/index.html
```

formatting will be preserved

YAML – Complex Keys

Use a question mark (“?”) to denote a key that spans multiple lines.

- Mind the indentation of the (“?”) and the (“：“) for the value

? This is an example of a
key that has multiple lines
: and this is its value

? – First Place
– Second Place
: [“Ahmed Salah”, “Amit Sharma”]

YAML – Aliases/Anchors

Anchors in YAML allow to easily duplicate content across a YAML document and then use it anywhere throughout the document using references.

- Anchors are defined as prefixing an ampersand (“&”) with the anchor name
- The anchored data can be referred by using an asterisk (“*”) along with the anchor name.

Anchor

```
anchored_content: &anchor_name This is a value that will be referenced again  
use_reference: *anchor_name
```

Alias

This will be interpreted as:

Anchored_content key will take the value -> This is a value that will be referenced again
use_reference key will take the same value because of the *anchor_name

YAML – Merge (“<<”)

We can use the idea of anchors along with the merge (“<<”) to replicate entire objects (mappings or lists) in another part of the YAML document.

Anchor &id001

- step1: &id001 # defines anchor label &id001

instrument: Lasik 2000

repetition: 1000

spotSize: 1mm

- step2:

<<: *id001

spotSize: 2mm

redefines just this key, refers rest from &id001

Merge (“<<：“)
And alias

YAML – .yaml or .yml are valid extensions

We can save our YAML files as .yaml or .yml

Some well known YAML IDEs/Editors include:

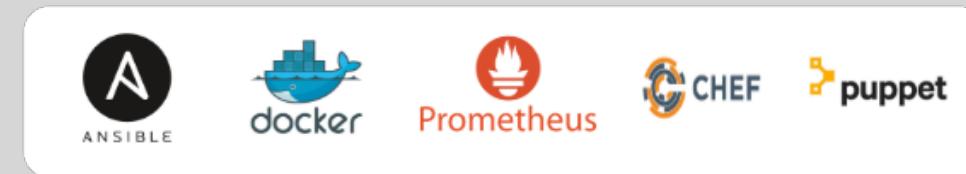
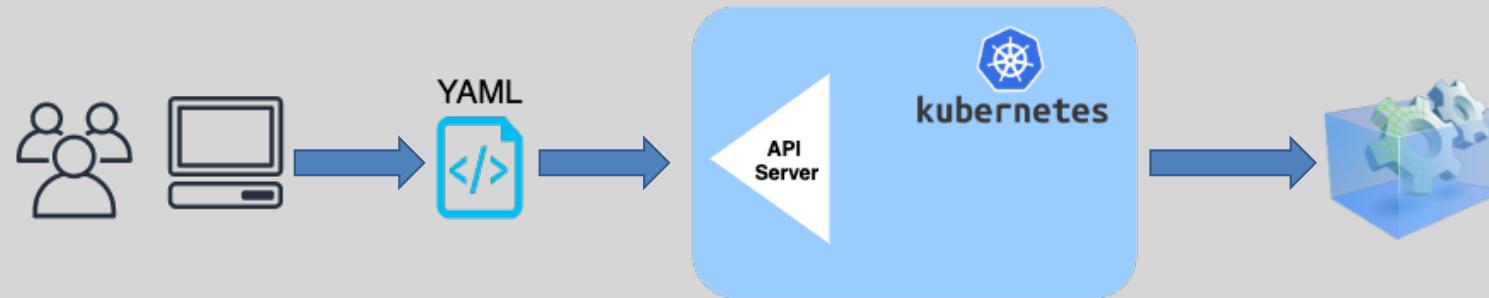
- Microsoft Visual Studio (VS)
<https://visualstudio.microsoft.com/downloads/>
- PyCharm
<https://www.jetbrains.com/pycharm/download/#section=mac>
- Eclipse IDE
- There are many online YAML editors / validators that are free to use



Introduction to APIs

Who Decides The Data Representation YAML Structures (YAML files)

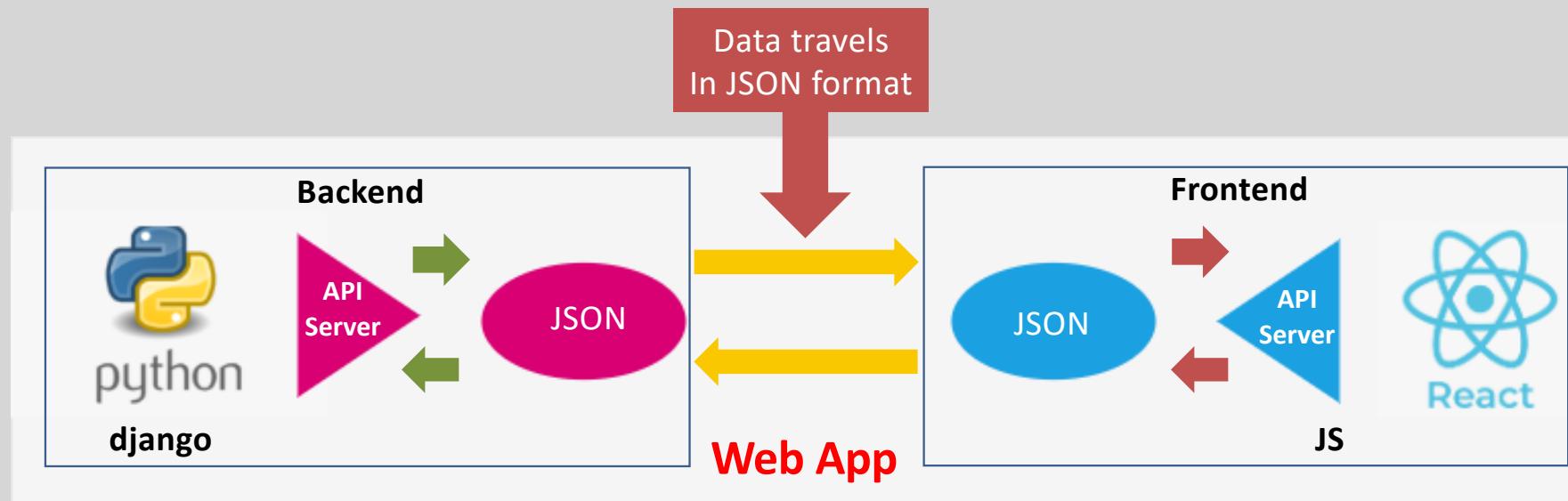
Whoever developed the tool decides on how the data should be represented (key/value pairs, lists, dictionaries, nested structures) using YAML or JSON



Application Programming Interface (API)

APIs is how applications and microservices communicate.

- This is how applications (code) complexity is hidden from the other components.



YAML & DevOps Career



ANSIBLE



docker



kubernetes



Prometheus



CHEF



puppet

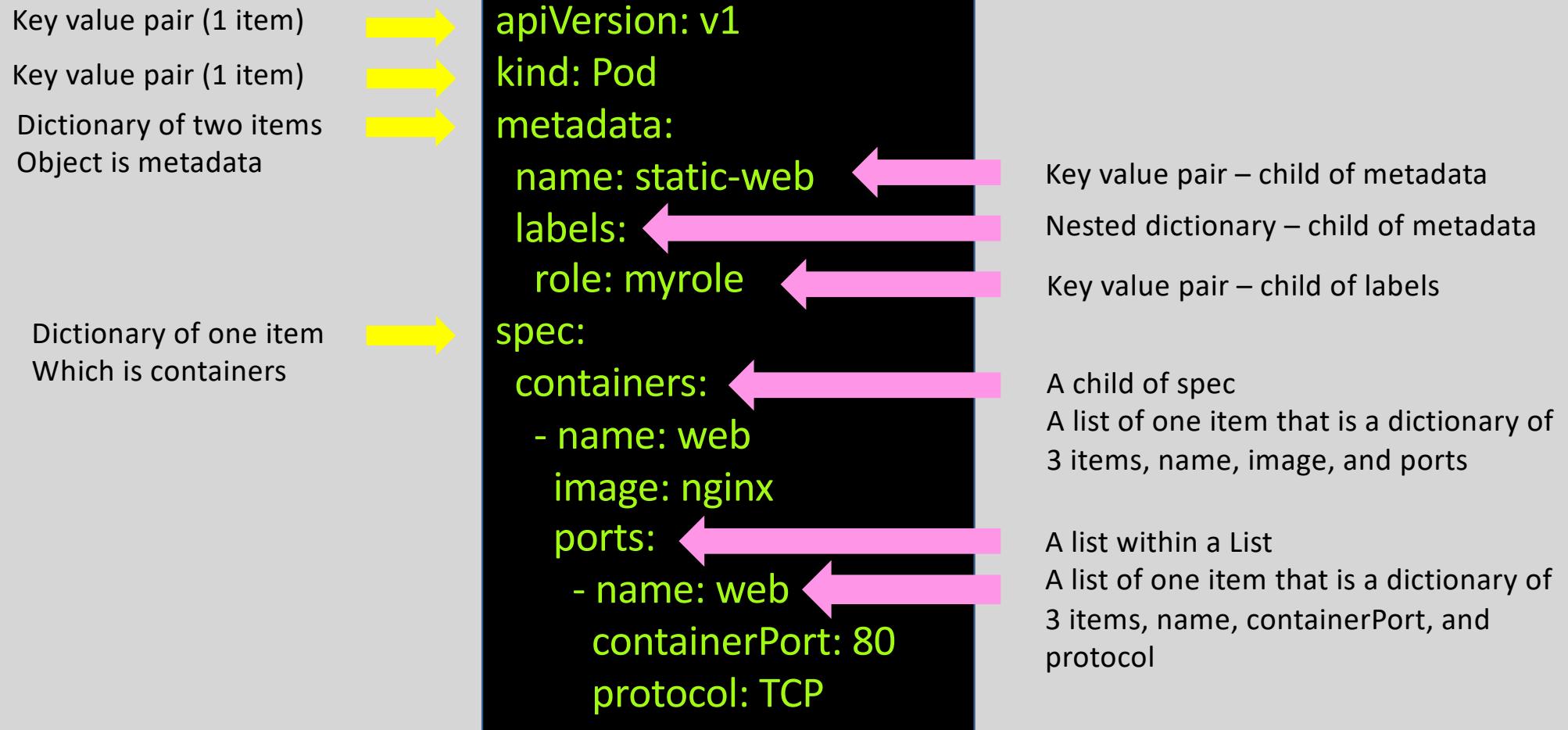


An Example Kubernetes YAML definition file

A Kubernetes YAML file – Let's Analyze it

```
apiVersion: v1
kind: Pod
metadata:
  name: static-web
  labels:
    role: myrole
spec:
  containers:
    - name: web
      image: nginx
      ports:
        - name: web
          containerPort: 80
          protocol: TCP
```

A Kubernetes YAML file – Let's Analyze it





Course End

