

# YAML Advanced Syntax

---



**Maaike van Putten**  
Software Developer & Trainer

[www.brightboost.nl](http://www.brightboost.nl)



## Overview



**Folding and chomping**

**Nested sequences**

**Nested mappings**

**Combining sequences and mappings**

**Dates in markdown files**

**Repeated nodes**

**Adding data types**



# Multi-line Strings: Folding and Chomping



**Folding**  
**Dealing with new lines in multi-line  
strings**



**Chomping**  
**How to deal with trailing newlines  
in multi-line strings**



# Demo



<https://yaml-multiline.info/>



# Nested Sequences

- [1, 2, 3]
- [A, B, C]

**Sequence of sequences**

**Combining different styles possible**



# YAML Nested Sequences

`flow_list:`

- [of, lists]
- [and, lists]

`block_list:`

- - of
  - lists
- - and
  - lists



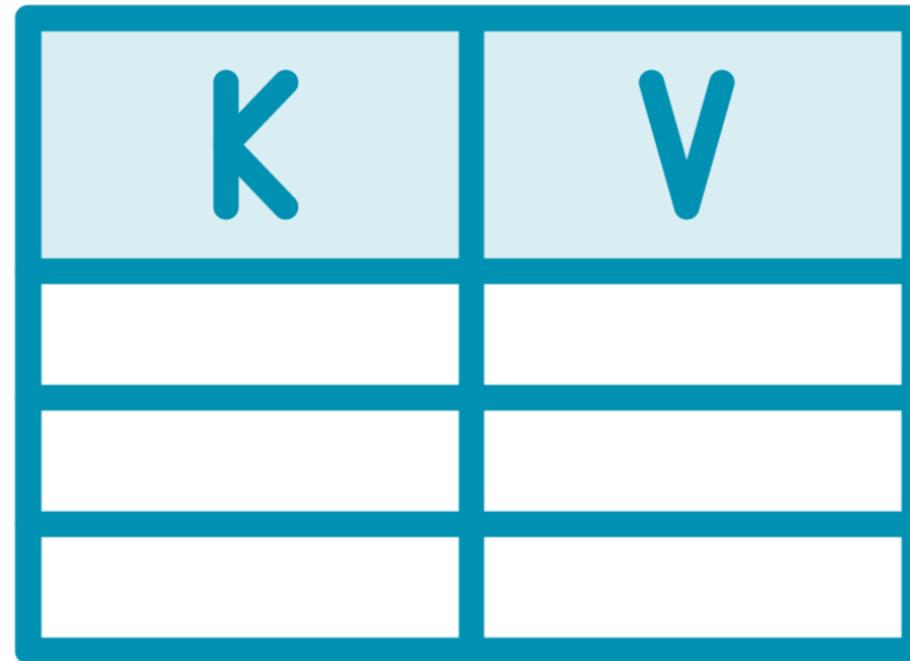
Demo



**Sequence of sequence**  
**Different styles**



# Nested Mappings



**Mapping of mappings**  
**Combining different styles possible**

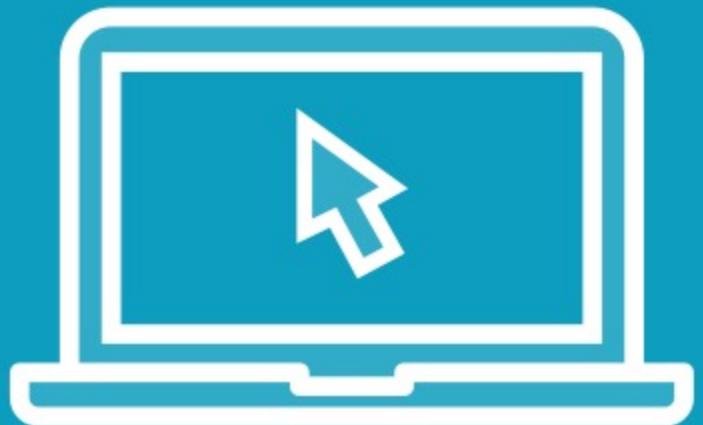


# YAML Nested Mappings

```
person:  
  name: Maaike  
  age: 30  
  address:  
    streetname: Langstraat  
    number: 1  
    zipcode: 1234AB  
    country: The Netherlands  
  flow-mapping: { where: inside, what: mapping }
```



Demo



**Mapping of mapping**  
**Different styles**



# Combining Sequences and Mappings



**Map scalar to sequence**  
**Sequence of mappings**  
**Mapping of sequence of mappings**  
**Et cetera**



# YAML Combining Sequences and Mappings

languages:

- programming:

- frontend:

- html

- css

- js

- backend:

- java

- python

- c#

- data serialization:

- yaml

- json



Demo



**Map scalar to sequence**

**Sequence of mappings**

**Mapping of sequence of mappings**



# Dates



**Scalar**

**Date and timestamp**

**Explicit tag to turn date into string**



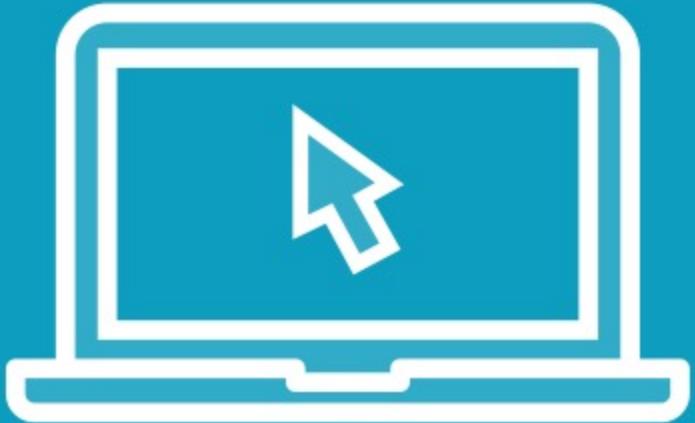
# YAML Dates

**timestamp**: 2022-03-22T22:19:56.10+02:00

**simple-date**: 2022-03-22



Demo



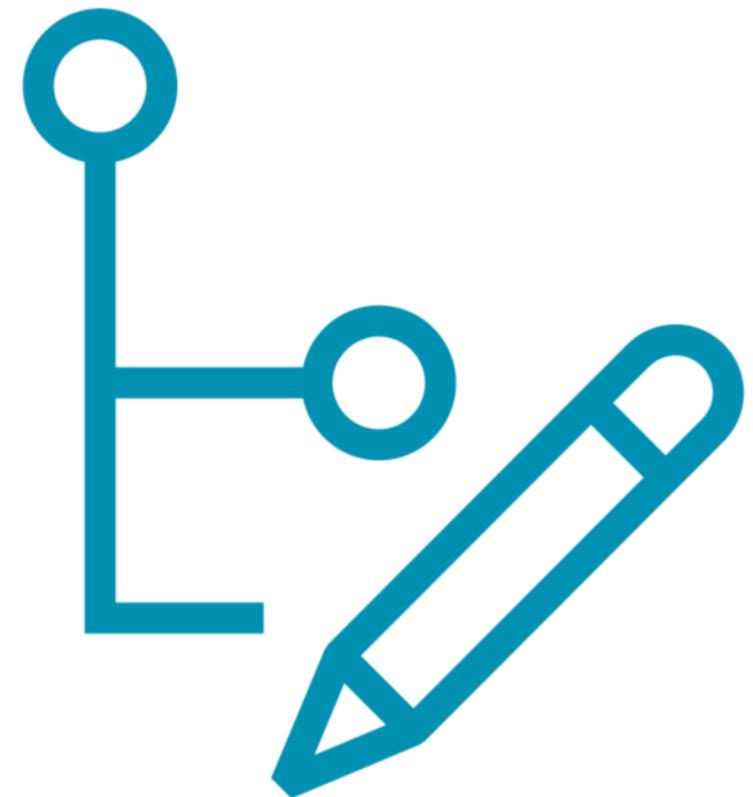
Date

Timestamp

Explicit string tag



# Repeated Nodes



**Cleaner YAML: DRY**  
**Anchor name (`&name`) and next referenced  
with alias (`*name`)**



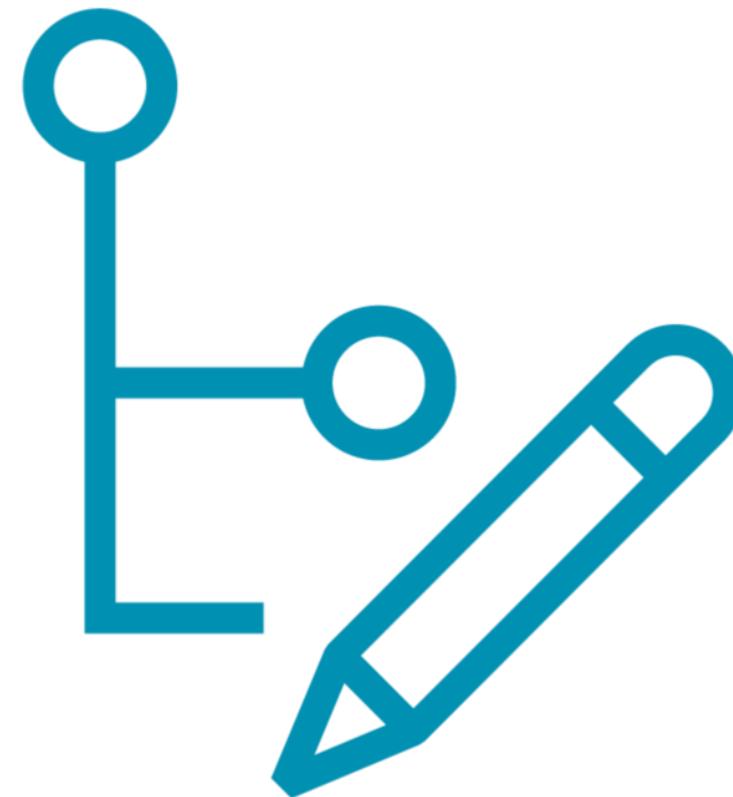
# YAML Repeated Nodes

```
value: &repeated Hi there
new_value: *repeated
```

```
person: &person
name: Maaike
age: 30
address:
  streetname: Langstraat
  number: 1
  zipcode: 1234AB
  country: The Netherlands
another_person: *person
```



# Repeated Nodes

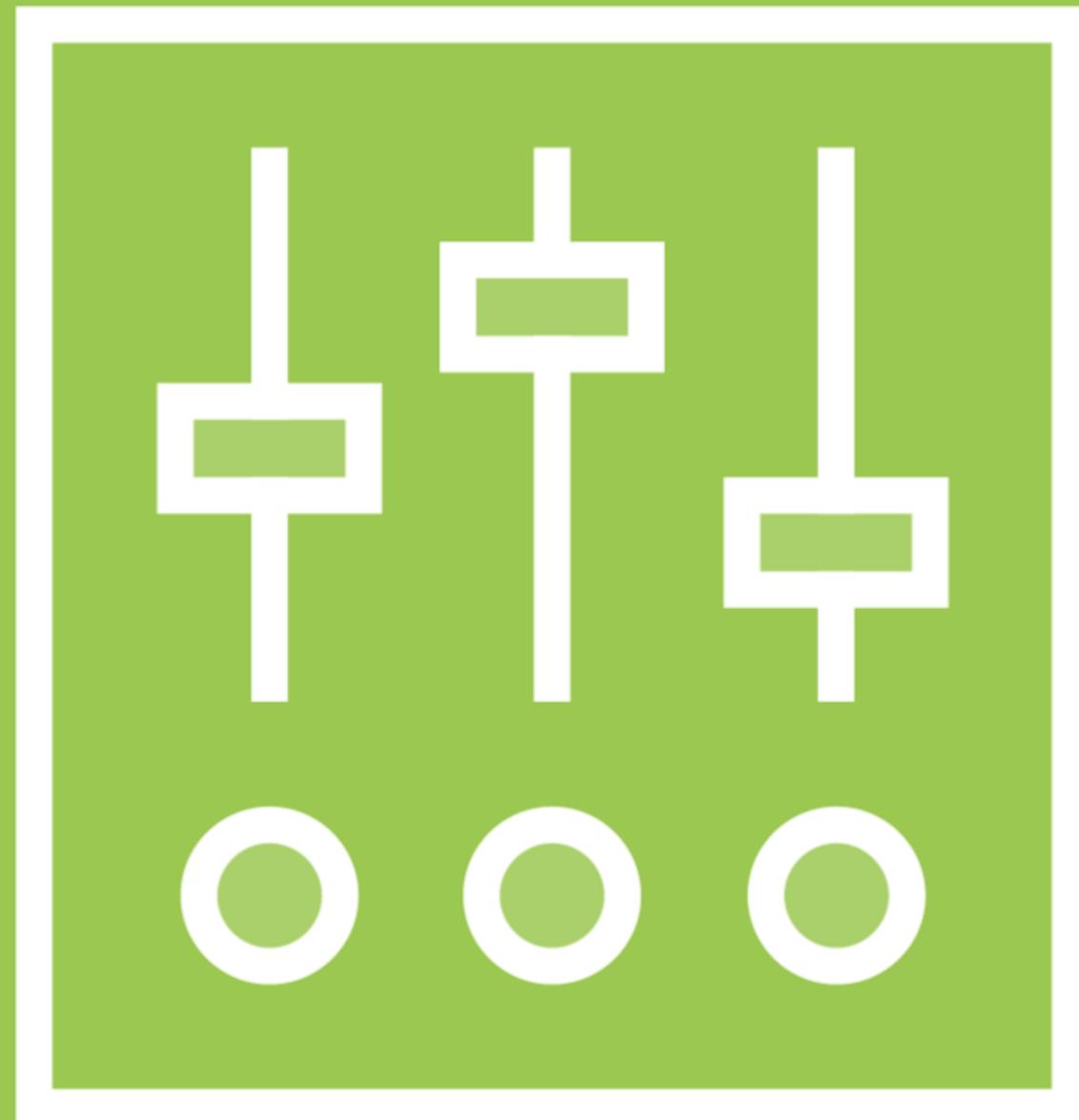


**Cleaner YAML: DRY**

**Anchor name (`&name`) and next referenced with alias (`*name`)**

**Overwrite part of the repeated node with merge key `<<`**





What if You'd Want  
to Modify the Alias?

Overwrite part of the repeated node  
with merge key <<:



# YAML Repeated Nodes Merge

```
person: &person
  name: Maaike
  age: 30
  address:
    streetname: Langstraat
    number: 1
    zipcode: 1234AB
    country: The Netherlands
```

```
another_person: *person
```

```
yet_another_person:
  <<: *person
  name: maria
```



Demo



**Need for repeated nodes**

**Simple repeated node**

**Repeated block**

**Overwrite part of the alias**





## Working with Tags

**Tags give a node a type**

**Untagged nodes are given a type**

**Explicit typing can be done with the !**

**Schemas specify the tags**



# YAML Working with Tags

not-a-date: !!str 2022-03-21

not-a-number: !!str 20

unnecessary: !!int 20



Demo



Comments



# Working with Schemas



**Default: Failsafe schema, JSON schema and core schema**

**Used to create custom data types with language specific schemas**

**Programming languages have their own schema that map YAML to the language native data structure**

**Interoperable schemas use global tags to represent the same data across different languages**



# Default Schemas

## Failsafe Schema

**Three tags: string, mapping, sequence**

**!!str: a string value**

**!!map: a mapping**

**!!seq: a sequence**

**!! Is shorthand for**

**!<tag:yaml.org,2002:....>**

## JSON Schema

**Failsafe tags**

**!!null: a null value**

**!!bool: boolean**

**!!int: integer**

**!!float: float**





## Core Schema

**Extension of the JSON schema**

**Same tags as the JSON schema, but extended tag resolution**

**Recommended default schema**

**Recommended to base custom schemas on this one**



Up Next:  
Parsing and Validation

---

