**Introduction**

Imagine objects with complex structures. For example, you have a large dictionary or a list with a bunch of other values. You need to preserve such an object without losing its structure. In other words, you want to serialize it. It's also good if the format of serialization has a simple syntax and is human-readable. So, let's get acquainted with a famous data format called YAML.

**YAML**is a recursive acronym for *YAML Ain't Markup Language***.** It's a human-readable data serialization standard for all programming languages. It is commonly used for configuration files and for storing and transferring data. It’s difficult to escape YAML if you’re doing anything related to software configuration. The usual extensions for YAML files are *.yaml* and *.yml*.

Let's take a closer look at this language in order to appreciate all its simplicity and functionality. We'll go over basic types, structures, and syntax.

**Basic data types**

YAML supports all essential data types like numbers, strings, booleans, etc. It recognizes some language-specific data types, such as dates, timestamps, and special numerical values. So, the list of basic YAML data types includes:

* *integers*like 15, 123
* *strings* like "15", 'Hello, YAML!', which may be enclosed either in double or single quotation marks
* *floats* like 15.033
* *booleans* (true or false)
* *null type* (null)

YAML auto-detects the type of data, but users can specify the type they need using !!. For example, if you need to specify the string *yes*, you need to write !!str yes.

Now you know the basic data types in YAML. Let's figure out the structures and their syntax.

**Maps**

Mapping consists of key-value pairs. For example:

---

object: Book

metadata:

name: Three Men in a Boat

author: Jerome K Jerome

genre: humorous account

published:

year: 1889

country: United Kingdom

The first line is a separator. It's optional unless you’re trying to define multiple structures in a single file. Then there is a set of key: value pairs as a block. Pairs are called *scalars*. The syntax is clean and simple; the usual format symbols, such as braces, square brackets, closing tags, or quotation marks, are unnecessary. Scalars are colon-separated, and there should be a space between the map elements. Note that in YAML, indentation is always done with spaces, not tabs.

**Lists**

The lists in YAML are sequences of objects, as the example below shows.

animals:

- cat

- dog

- bird

The number of items on the list is not limited. Each item on the list should start with a hyphen. Elements are separated from the parent with spaces; after a parent name, there should be a colon. The example above represents a block style. In flow style, the list looks like this: [cat, dog, bird].

These are the structures. Next, we'll learn how to combine them.

**Combination**

Maps and lists can be combined so that one may have maps of maps, or maps of lists, or lists of lists, or lists of maps. Let's consider an example of a to-do map, where keys are weekends and values are lists of things to do during each day:

weekend:

saturday:

- order cleaning

- order a pizza

- watch new series

sunday:

- go to yoga

- hang out with a friend

Also, if you need to denote a string that preserves newlines instead of a list, use | character:

saturday: |

order cleaning

order a pizza

watch new series

Great, now you know how to use structures. The last important thing we should mention here is comments.

**Comments**

One may add comments to the YAML file. Comments start with # and go till newline. They can be made anywhere in the line, for example:

# The comment

metadata: # this is metadata

name: Three Men in a Boat

author: Jerome K Jerome

genre: humorous account

**Conclusion**

To sum up, we've discussed that:

* YAML is a human-readable text-based format that lets you easily specify configuration-type information
* YAML supports integers, strings, floats, booleans, and null data types
* there are two data structures in YAML, lists and maps that can also be combined
* comments can be added to YAML file

Let's get to the tasks!

Mike created a YAML list of his hobbies:

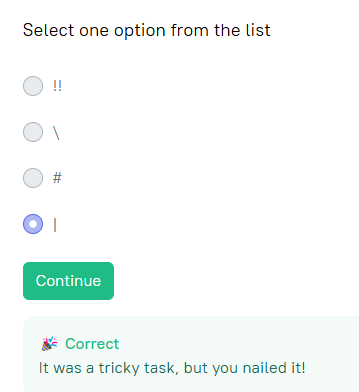
hobbies:

skating

football

guitar playing

playing computer games

He wants to denote a string that preserves newlines. Which symbol should he add to the list?

hobbies: |

skating

football

guitar playing

playing computer games

What data structures are supported in YAML? List and map