What is YAML

- YAML means **yet another markup language**. It is a data serialization language that works well with the modern programming language, and it is human-friendly.
- It is used to define the **data structures** that are very easy to understand. These data structures are very easy to manage and maintain by the users.
- YAML uses **space** to define something. If we use a single space or double space, it has different meanings in YAML. Spaces change the meaning of data structure.
- YAML is **case sensitive.**
- YAML is **language independent.** If we define the YAML once, the same YAML file can be invoked in Python file, Ruby, etc.
- In the starting, YAML was very useful and friendly to people who work with data. YAML uses **Unicode printable characters**. In Unicode, some provide the information of structure, and others contain the data itself.

Goals of YAML

1. **Human-readable:** YAML is human readable. Humans can easily read it.
2. **Portable:** YAML can work across multiple programming languages easily.
3. **Consistent:** YAML is consistent, and it is able to support generic tools.
4. **Support various languages:** YAML matches the native data structure of agile methodologies like PHP, JavaScript, Perl, Ruby, and Python.
5. **One-pass processing:** when a programming language goes through the YAML file, it only needs to go through once to complete its task.

6. **Extensive and expressive:** YAML is extensive, that means it should be easily readable by the human, and it is expressive.
7. **Easy Implementation:** The implementation of YAML is easy and useful.

Prerequisite

Before learning YAML, you should have basic knowledge of basic computer functionality, basic mathematics, computer language, and logical operators.

YAML Syntax

In YAML, the documents are a collection of key-value pairs where the value can be complex as a tree or simple as a string. We will use spaces instead of taps while indenting the YAML. Indentation plays a very important part in the YAML file. Indentation is used to group thing that goes together. In indentation, the character will not consider as a part of content information.

In YAML, indentation represents the structure. Dashes (-) are used to represent the lists and colons(:) are used to represent the key-value pair.

Example 1:

```
datacenter:
 location: canada
 cab: 15
```

The above example shows a **key-value mapping indentation**. It has an indentation under the datacenter. The two values under datacenter are part of datacenter mapping. They are associated with each other because they have two space indents before the actual key that is location and cab.

Example 2:

```
host: phl-42
```

This shows a **key-value mapping**. Here, we use a colon for key-value pairs. The key is host, and the value is phl-42.

Example 3:

```
animals:
- dog
- cat
 - mouse
```

This example shows a **list indentation**. It also has indentation because it has a list under animals. It contains the dashes to indicate that it is a list, not just a key-value.

Indicator Characters

Character	Functionality
:	It is used to describe the mapping value.
-	It describes the entry of the block sequence.
,	It describes the entry of flow collection.
?	It is used to describe the mapping key.
!	It describes the tag of a node.
&	It describes the anchor property of a node.
|	It is used to describe the literal block scalar.
#	It is used to describe the comments.
>	It is used to describe the folded block scalar.
{	It is used to start the mapping of flow.
}	It is used to end the mapping of flow.

Character	Functionality
:	It is used to describe the mapping value.
-	It describes the entry of the block sequence.
,	It describes the entry of flow collection.
?	It is used to describe the mapping key.
!	It describes the tag of a node.
&	It describes the anchor property of a node.
|	It is used to describe the literal block scalar.
#	It is used to describe the comments.
>	It is used to describe the folded block scalar.
[	It is used to start the sequence of flow.
]	It is used to end the sequence of flow.
%	It describes the use of directives.
*	It is used to describe the alias node.

YAML Data types

YAML has three types of data types:

1. Scalar
2. List
3. Dictionary

Scalar data type:

Scalar is a simple data type. In YAML, scalar means a simple value for a key. The value of the scalar can be integer, float, Boolean, and string. Scalar data types are classified into two data types:

a. Numeric Data type :- Integer , Floating point numbers , Booleans
b. String

Lists

We can define the list in a single line as follows:

```
---
items: [6, 7, 8, 9, 10]
name: [six, seven, eight, nine, ten]
```

This style is known as block style. We can put the above list in multiple lines as follows:

```
---
items:
  - 6
  - 7
  - 8
name:
  - "six"
  - "seven"
  - "eight"
  - "nine"
```

This style is known as flow style. A list that contains complex objects needs multiple lines.

```
---
items:
  - values:
      value1:
      value 2:
      value 3:
  - other values:
      key: value
```

Any number of valid YAML values can contain by an array. But the value of a list can't be the same type.

Dictionaries

If we want to write a complex YAML file which holds the complex data structure, we will use dictionaries. It is a collection of key: value pairs and each of the key: value pairs can be nested with a lot of options.

Example 1:

```
---
student1: “Rutik”
hobbies:
  - music
  - reading
  - dancing
```

In the above example, student is the first key, and john is the value. Hobbies are the second key, but it is nested, which means it contains a list of values. The value of the key can again be a key: value pair, which we will see in the next example.