**MIT xPRO Data Engineering Certificate**

**YAML**

**What Is YAML?**

YAML, initially named “Yet Another Markup Language” and now often referred to as “YAML Ain’t Markup Language,” is a language to store data and configuration in a human-readable format.

Using markup instead of code for certain things is better because you need to maintain code from time to time. Therefore, it is a best practice to reduce the code to a minimum whenever possible and create it in a manner that it doesn’t need modification for the various data inputs that it takes. This strategy is considered a best practice because it separates the concerns of creating the data and creating the code. Additionally, it reduces code maintenance over time.

Markup languages make it easier to store information in a more accessible format. These files can be exchanged between programs without consuming much bandwidth or resources and can be read by the most common protocols. In other words, markup languages are not associated with any system command.

There are numerous benefits of using YAML over other markdown languages, such as JSON. These benefits are that YAML is generally less verbose, can handle complex data types, supports writing comments in your text using the symbol #, and is more human-readable.

**Basics of YAML Files**

YAML files use Python-style indentation to indicate nesting. Writing in YAML is also fairly simple, as there are no usual format symbols, such as braces, square brackets, closing tags, or quotation marks. YAML files use a .yml or .yaml extension. The general structure of a YAML file is a *map* or a *list*.

*Maps*, similarly to Python *dictionaries*, allow you to associate *key*-value pairs. Each *key* must be unique, and the order doesn't matter.

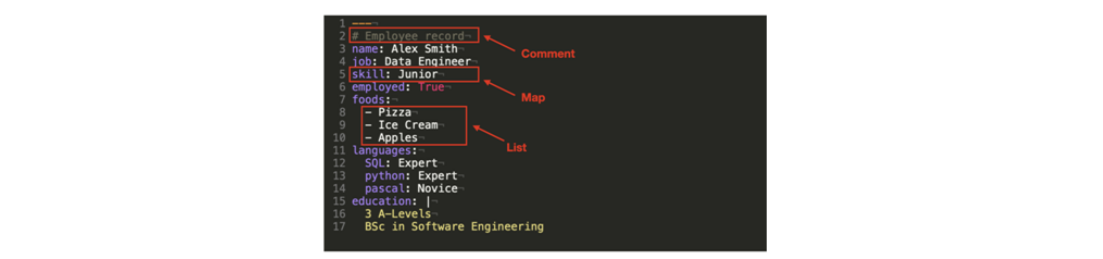
A *map* in YAML needs to be resolved before it can be closed and a new *map* is created. A new *map* can be created by either increasing the indentation level or by resolving the previous *map* and starting an adjacent *map*.

A *list* includes values listed in a specific order and may contain any number of items needed. A *list* sequence starts with a dash (-) and a space, while an indentation separates it from the parent. You can think of a sequence as a Python *list*. A *list* can be embedded in a *map*.

A YAML file can also contain scalars that can be used as values such as *strings*, integers, dates, numbers, or *booleans*.

When creating a YAML file, you’ll need to ensure that you follow these syntax rules and that your file is valid.

The image below shows an example of a YAML file with the components explained above:



The three dashes at the top of the document symbolize the beginning of a new YAML file.

Note that YAML also supports integers, *floats*, and *strings* that can be enclosed in single quotes, double quotes, or no quotes at all.

**What Is YAML Used For?**

One of the most common uses for YAML is to create configuration files to simplify the communication between a *client* and a *server*. It's recommended that configuration files be written in YAML rather than JSON because YAML has better readability and is more user-friendly.