SkillAssure

YAML Introduction

YAML Introduction

What is YAML?

- A YAML file is used to represent data.

YAML

```
Servers:
    -   name: Server1
        owner: John
        created: 12232012
        status: active
```

XML

```
<servers>
    <server>
        <name>Server1</name>
        <owner>John</owner>
        <created>12232012</created>
        <status>active</status>
    </server>
</servers>
```

JSON

```
{
    Servers : [
        {
        name: Server1,
        owner: John,
        created: 12232012,
        status: active,
        }
    ]
}
```

YAML

- Let's take a close look at YAML.
- Starting with the simplest form of data, **key-value pair**
- In YAML file Key and value are separated by a colon.

Key Value Pair

```
Fruit: Apple
Vegetable: Carrot
Liquid: Water
Meat: Chicken
```

- The keys are fruit, vegetable, liquid and meat and the values are apple, carrot, water and chicken
- You must have a space followed by a colon to differentiate between the key and the value.

SkillAssure

YAML

- Let's take a look at how an array is represented.
- We would like to list some fruits and vegetables.
- Start the list by specifying the name (fruits & vegetables) followed by a colon
- On the next line, enter each item with a dash in the front.

Array/Lists

```
Fruits:
-   Orange
-   Apple
-   Banana
Vegetables:
-   Carrot
-   Cauliflower
-   Tomato
```

- The dash indicates that it's an element of the array

YAML

- Let's take a look at how a **dictionary** is represented.
- A dictionary is a set of properties grouped together under an item
- We will represent nutritional information (i.e. calories, fat and carbs) of two fruits.
- Start by specifying the item (banana & grapes) followed by a colon
- On the next line, enter each item property (key: value) with a few spaces in the front.

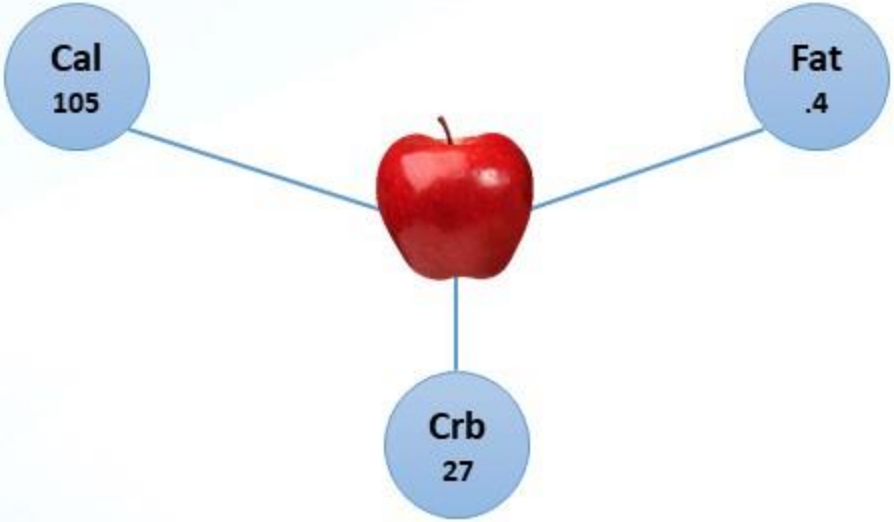
Dictionary/Map

```
Banana:
   Calories: 105
   Fat: 0.4g
   Carbs: 27g
Grapes:
   Calories: 62
   Fat: 0.3g
   Carbs: 16g
```

- You must have equal number of blank spaces before the properties of a single item

YAML

- Let's take a closer look at spaces in YAML.
- Here we have a dictionary representing the nutrition information of apple.
- We have equal number of spaces before each property.
- This indicates that these property (key-value pairs) fall within apple.

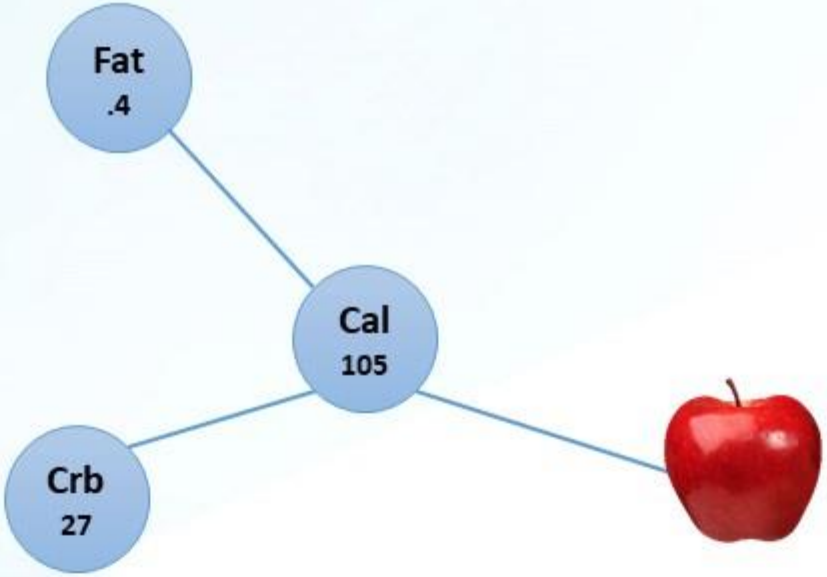


Dictionary/Map

```
Apple:
□□□Calories: 105
□□□Fat: 0.4g
□□□Carbs: 27g
```

YAML

- What if we had extra spaces for fat and carbs?
- Then they will fall under calories and will become properties of calories.



Dictionary/Map

```
Apple:
□□□□Calories: 105
□□□□□□ Fat: 0.4g
□□□□□□ Carbs: 27g
```

YAML

- Let’s represent a list of fruits
  - Banana
  - Grape

Key-Value/Dictionary/List

```
Fruits:
  -  Banana
  -  Grape
```

YAML

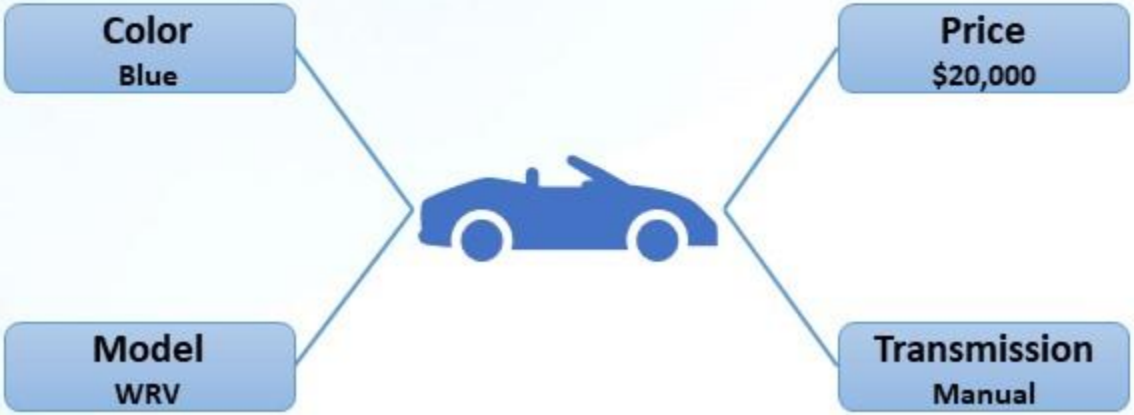
- Let’s represent a list of fruits
  - Banana
  - Grape
- Now represent the nutrition information (Calories, Fat & Carbs) of each of these fruits

Key-Value/Dictionary/List

```
Fruits:
  -   Banana:
          Calories: 105
          Fat: 0.4g
          Carbs: 27g
  -   Grape:
          Calories: 62
          Fat: 0.3g
          Carbs: 16g
```

Dictionary vs List vs List of Dictionaries

- Let's take an example of a car
- A car is a single object and it has properties such as color, model, transition and price
- To store different information or properties of a single object we use a dictionary
- In this dictionary, I have properties of the car defined in a key: value format.



Color
Blue

Price
$20,000

Model
WRV

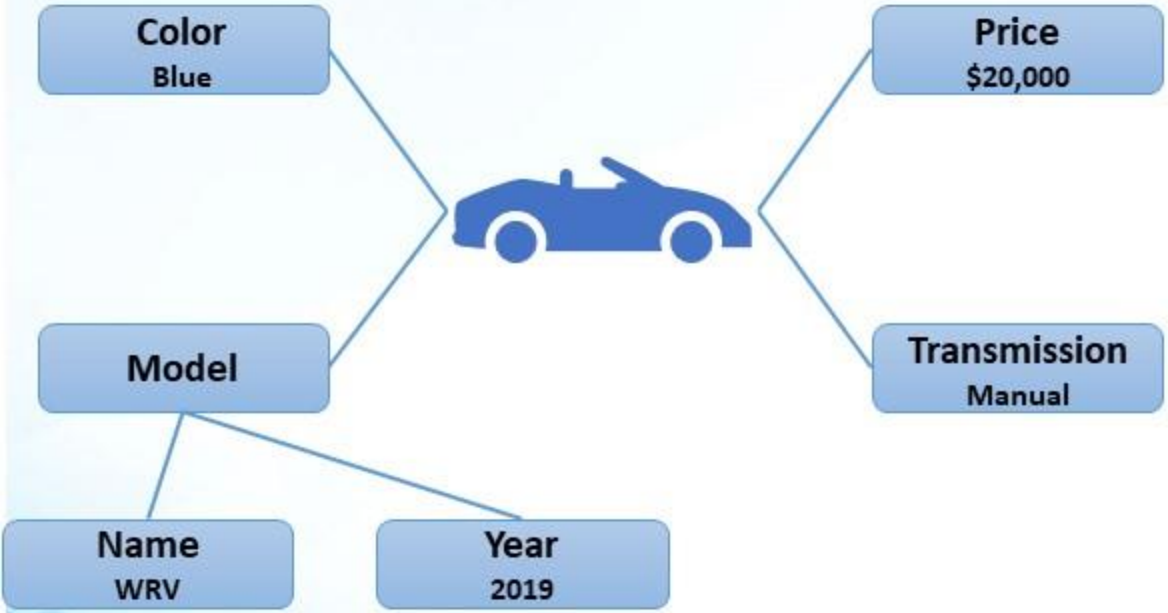
Transmission
Manual

Dictionary in Dictionary

Color: Blue
Model: WRV
Transmission: Manual
Price: $20,000

Dictionary Vs List Vs List of Dictionaries

- Let's now split the model further into model name and make year
- We can represent this as a dictionary within another dictionary.
- The single value of model is now replaced by a small dictionary with two properties name and year.



Dictionary in Dictionary

Color: Blue
Model:
    Name: WRV
    Year: 2019
Transmission: Manual
Price: $20,000

Dictionary Vs List Vs List of Dictionaries

- Let’s store the name of 3 cars.
- To store this, we would use a list or an array as it is multiple items of the same type
- Since we are only storing the names, we have a single list of strings.
- What if we want to store all information about each car.

Blue WRV

Green WRV

Yellow WRV

List of Strings

- Blue WRV
- Green WRV
- Yellow WRV

Dictionary vs List vs List of Dictionaries

- We will then modify the array from a list of strings to a list of dictionaries.
- We expand each item in the array and replace the name with the dictionary we built earlier.

Blue WRV

Color: Blue
Model:
    Name: WRV
    Year: 2019
Transmission: Manual
Price: $20,000

Green WRV

Color: Green
Model:
    Name: WRV
    Year: 2020
Transmission: Manual
Price: $22,000

Yellow WRV

Color: Yellow
Model:
    Name: WRV
    Year: 2017
Transmission: Manual
Price: $18,000

List of Dictionaries

- Color: Blue
  Model:
      Name: WRV
      Year: 2019
  Transmission: Manual
  Price: $20,000

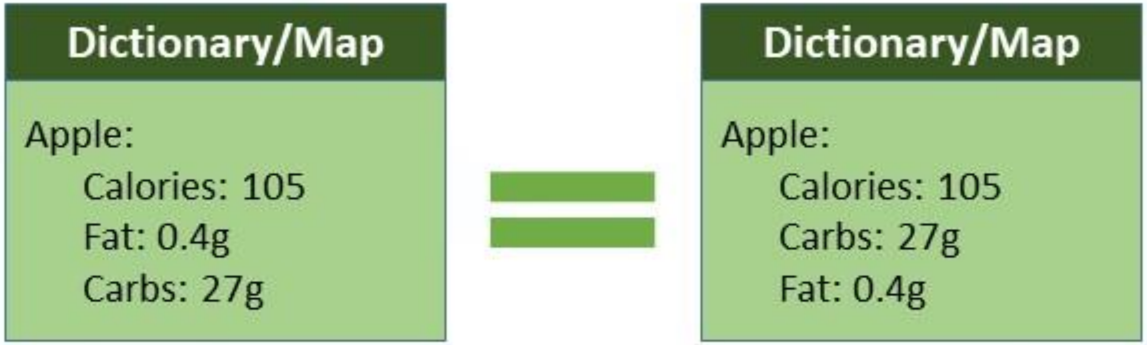
- Color: Green
  Model:
      Name: WRV
      Year: 2020
  Transmission: Manual
  Price: $22,000

- Color: Yellow
  Model:
      Name: WRV
      Year: 2017
  Transmission: Manual
  Price: $18,000

YAML - Notes

Dictionary - **Unordered Collection**

- Order of properties fat and carbs do not match
- The properties can be defined in any order
- Two dictionaries will be same, if the values of each property match



Dictionary/Map

```
Apple:
    Calories: 105
    Fat: 0.4g
    Carbs: 27g
```

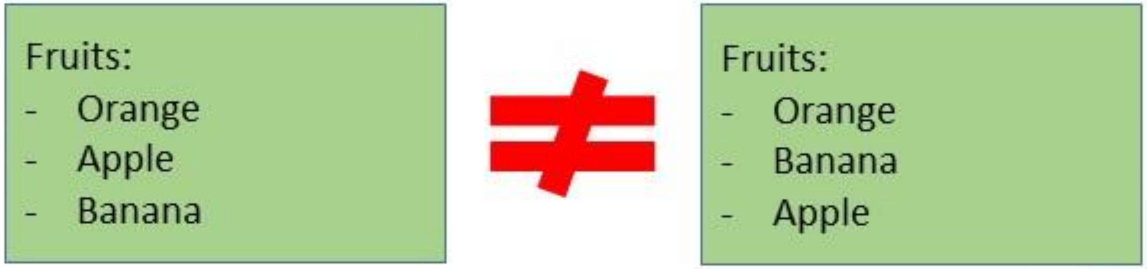
=

Dictionary/Map

```
Apple:
    Calories: 105
    Carbs: 27g
    Fat: 0.4g
```

Lists/Array - **Ordered collection**.

- Order of items matters
- The 2 lists shown are not the same
- Because apple and banana are in different positions.



```
Fruits:
-   Orange
-   Apple
-   Banana
```

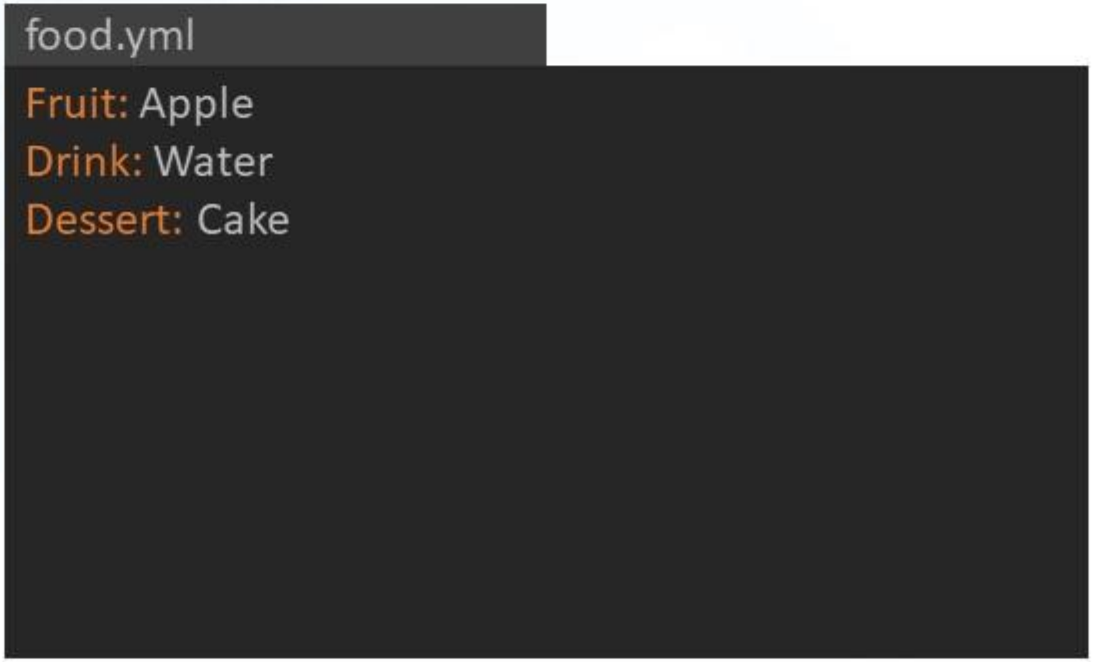
≠

```
Fruits:
-   Orange
-   Banana
-   Apple
```

Coding Exercise

Exercise 1

Update the food.yml file to add a Vegetable - Carrot

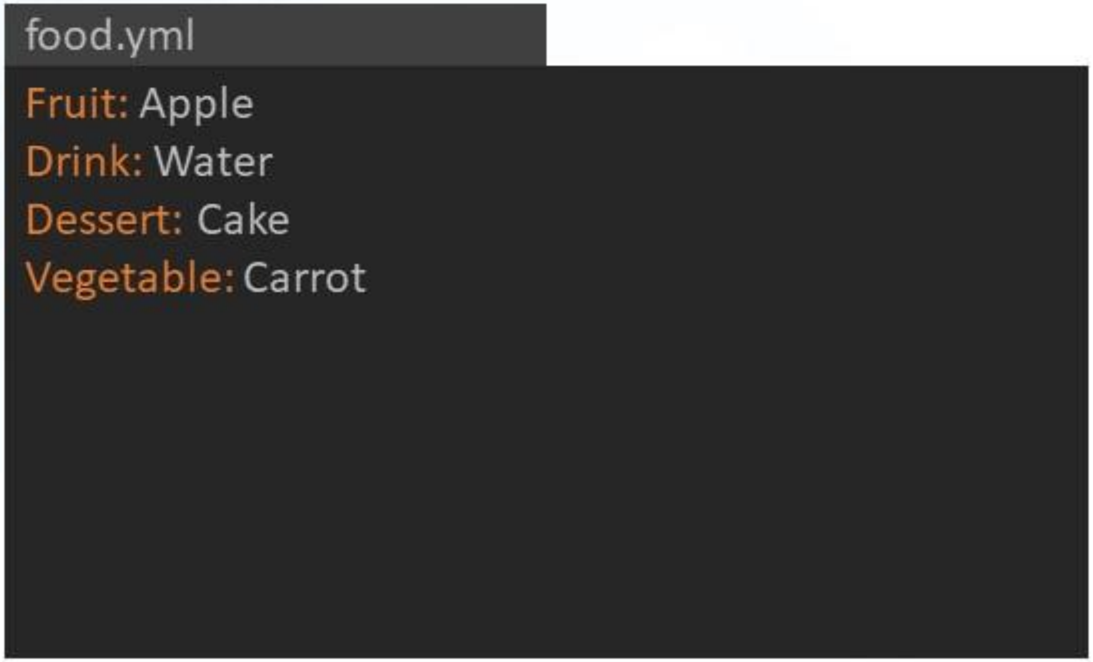


food.yml

```yaml
Fruit: Apple
Drink: Water
Dessert: Cake
```

Exercise 1 - Solution

Update the food.yml file to add a Vegetable - Carrot



food.yml

```
Fruit: Apple
Drink: Water
Dessert: Cake
Vegetable: Carrot
```

Exercise 2

Update the food.yml file to add a List of vegetables – Carrot, Tomoto, Cucumber

food.yml

```
Fruits:
 - Apple
 - Banana
 - Orange
```

Exercise 2 – Solution

Update the food.yml file to add a List of vegetables – Carrot, Tomoto, Cucumber

food.yml

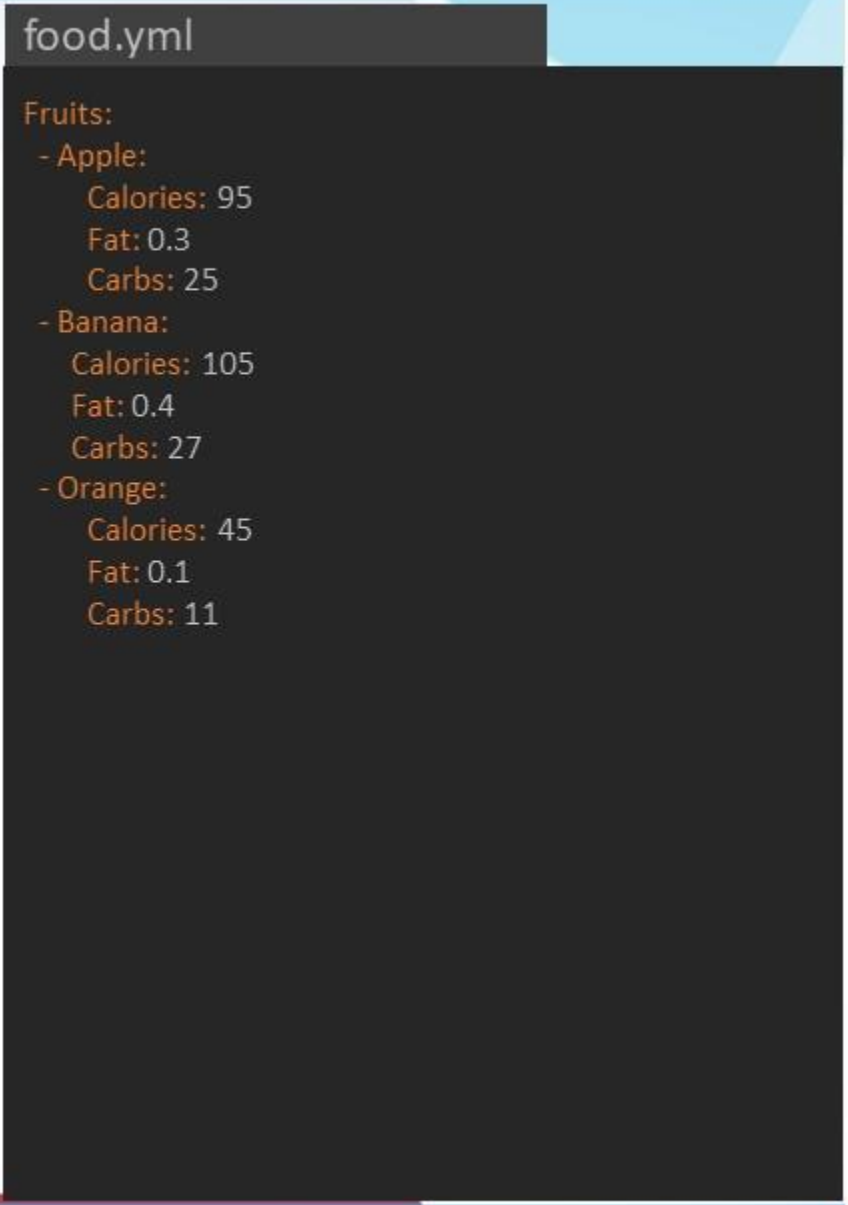
```yaml
Fruits:
 - Apple
 - Banana
 - Orange

Vegetables:
 - Carrot
 - Tomato
 - Cucumber
```

Exercise 3

We have updated the food.yml file with nutrition information for Fruits. Similarly update the nutrition information for Vegetable. Use the below table for information

Vegetables	Calories	Fat	Carbs
Carrot	25	0.1	6
Tomato	22	0.2	4.8
Cucumber	8	0.1	1.9



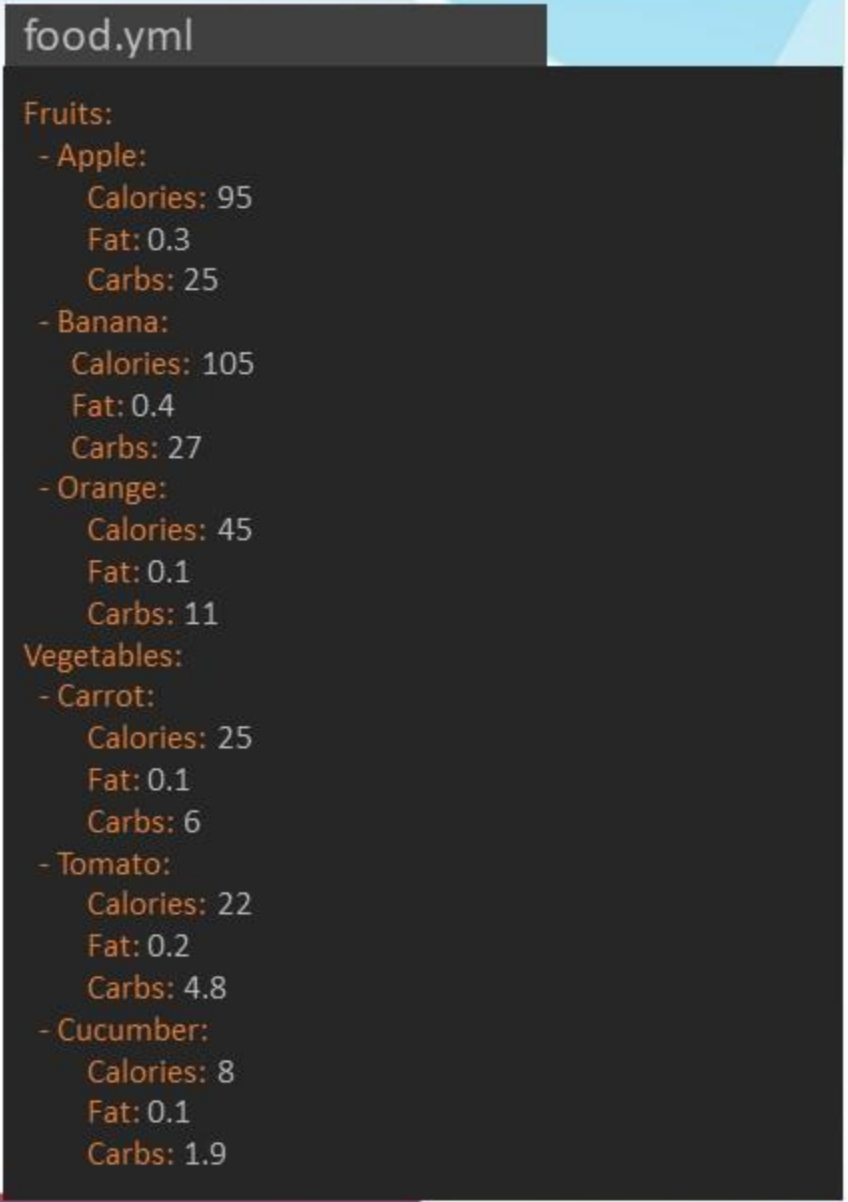
food.yml

```
Fruits:
 - Apple:
    Calories: 95
    Fat: 0.3
    Carbs: 25
 - Banana:
   Calories: 105
   Fat: 0.4
   Carbs: 27
 - Orange:
    Calories: 45
    Fat: 0.1
    Carbs: 11
```

Exercise 3 - Solution

We have updated the food.yml file with nutrition information for Fruits. Similarly update the nutrition information for Vegetable. Use the below table for information

Vegetables	Calories	Fat	Carbs
Carrot	25	0.1	6
Tomato	22	0.2	4.8
Cucumber	8	0.1	1.9



food.yml

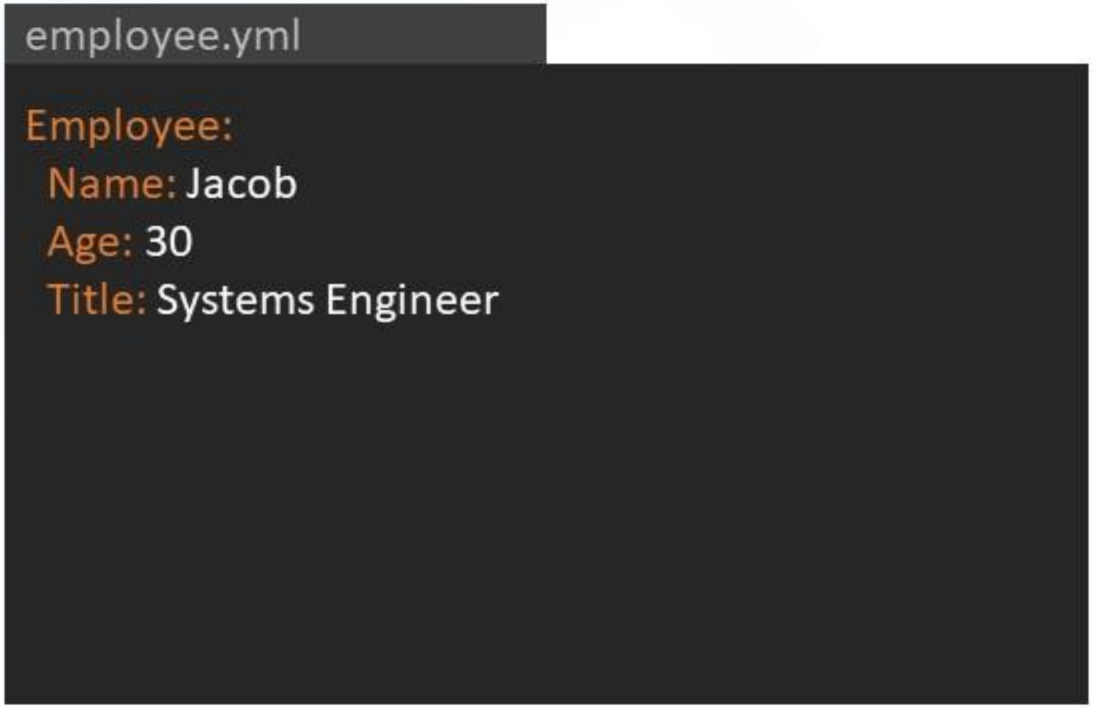
```
Fruits:
 - Apple:
    Calories: 95
    Fat: 0.3
    Carbs: 25
 - Banana:
   Calories: 105
   Fat: 0.4
   Carbs: 27
 - Orange:
    Calories: 45
    Fat: 0.1
    Carbs: 11
Vegetables:
 - Carrot:
    Calories: 25
    Fat: 0.1
    Carbs: 6
 - Tomato:
    Calories: 22
    Fat: 0.2
    Carbs: 4.8
 - Cucumber:
    Calories: 8
    Fat: 0.1
    Carbs: 1.9
```

Exercise 4

Jacob is 30-year-old Male working as a system engineer at a firm. Represent Jacob's information(Name, Age, Title) in YAML format. Create a dictionary named Employee and define properties under it.

Exercise 4 - Solution

Jacob is 30-year-old Male working as a system engineer at a firm. Represent Jacob’s information(Name, Age, Title) in YAML format. Create a dictionary named Employee and define properties under it.

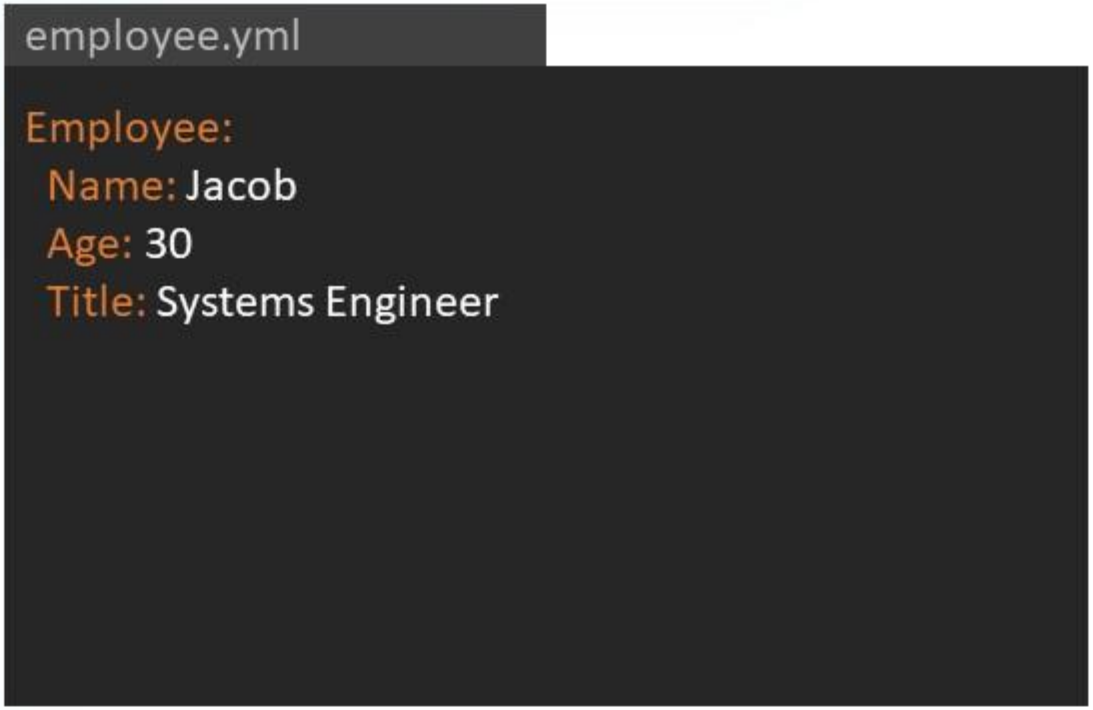


employee.yml

```yaml
Employee:
  Name: Jacob
  Age: 30
  Title: Systems Engineer
```

Exercise 5

Update the YAML file to represent the Projects assigned to Jacob. Remember Jacob works on Multiple projects –Automation and Support. So remember to use a list

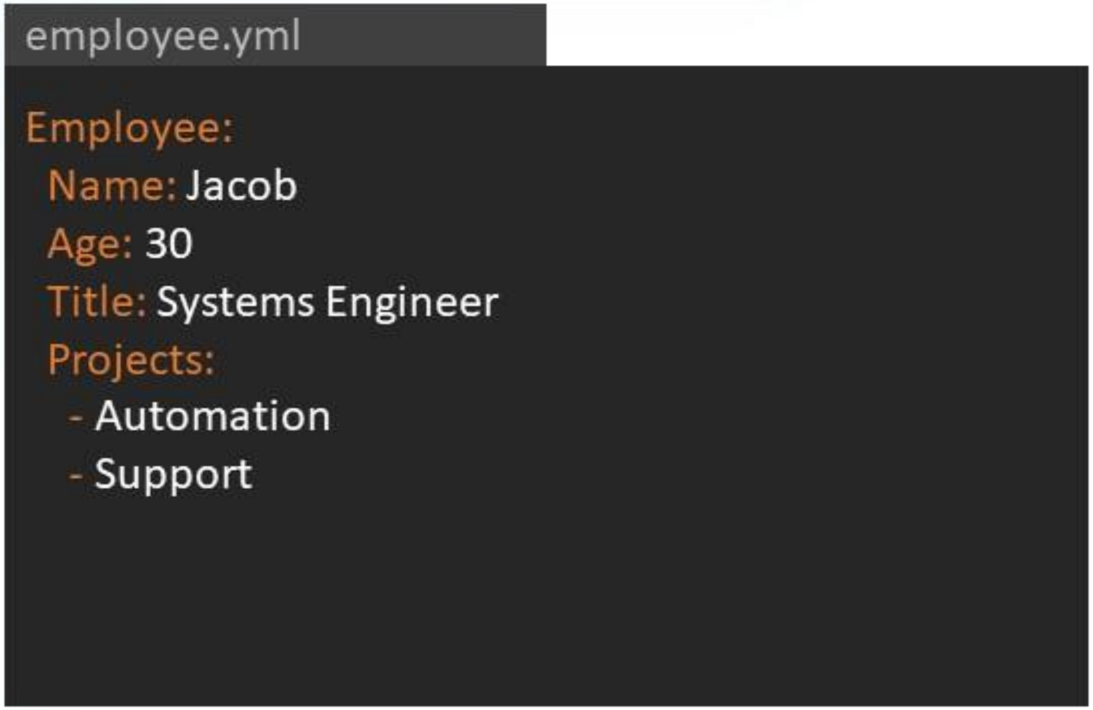


employee.yml

```
Employee:
  Name: Jacob
  Age: 30
  Title: Systems Engineer
```

Exercise 5 - Solution

Update the YAML file to represent the Projects assigned to Jacob. Remember Jacob works on Multiple projects –Automation and Support. So remember to use a list

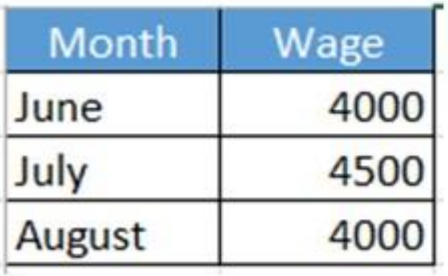


employee.yml

```yaml
Employee:
  Name: Jacob
  Age: 30
  Title: Systems Engineer
  Projects:
    - Automation
    - Support
```

Exercise 6

Update the YAML file to Jacob’s pay slips. Add a new property “Payslips” and create a list of pay slip details (Use list of dictionaries). Each pay slip detail contains Month and wage



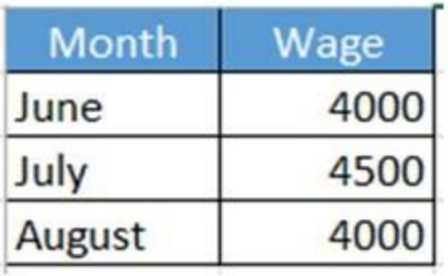
Month	Wage
June	4000
July	4500
August	4000

employee.yml

```yaml
Employee:
  Name: Jacob
  Age: 30
  Title: Systems Engineer
  Projects:
    - Automation
    - Support
```

Exercise 6 - Solution

Update the YAML file to Jacob's pay slips. Add a new property "Payslips" and create a list of pay slip details (Use list of dictionaries). Each pay slip detail contains Month and wage



Month	Wage
June	4000
July	4500
August	4000

employee.yml

```
Employee:
  Name: Jacob
  Age: 30
  Title: Systems Engineer
  Projects:
   - Automation
   - Support
  Payslips:
   -   Month: June
       Wage: 4000
   -   Month: July
       Wage: 4500
   -   Month: August
       Wage: 4000
```

Thank You