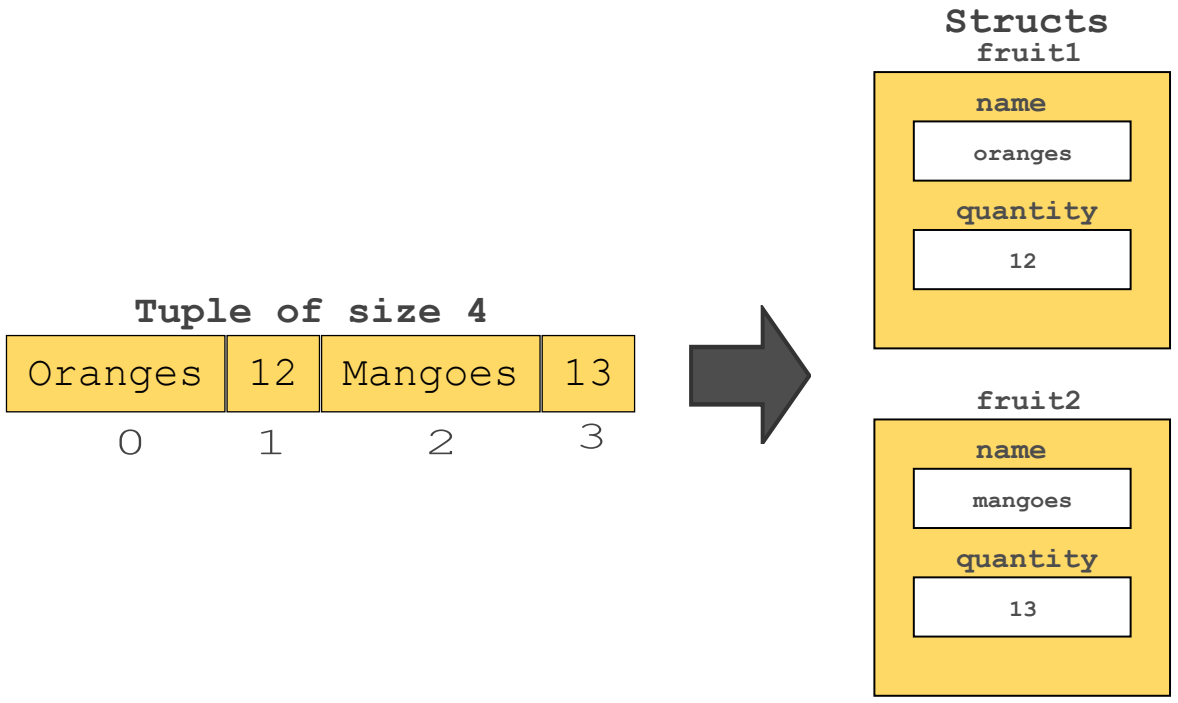
Tuple Structs

This lesson will walk you through tuple structs.

We'll cover the following

What Are Tuple Structs?

Recall tuples in Chapter 3.Tuple Structs are a data type that is a hybrid between a tuple and a struct.



> **Why a tuple struct?**

> In the example above, when it is only a tuple we don't know explicitly what each item in the tuple means. But when it is a tuple struct, we can assign a name to each item.

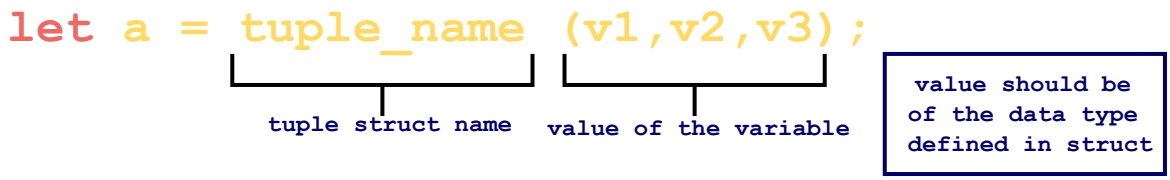
Define a Tuple Struct

Tuples can be of type struct by adding the `struct` keyword before the tuple name, followed by the data type of the variables enclosed within round brackets.

Define a tuple struct

Initialize a Tuple Struct

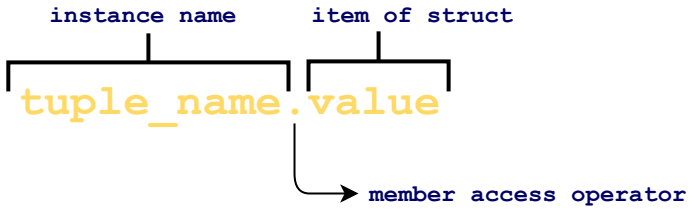
A tuple struct can be initialized like a tuple.



Initialize values of a tuple struct

Access a Tuple Struct

The tuple struct can be accessed using a `.` operator like a traditional struct.



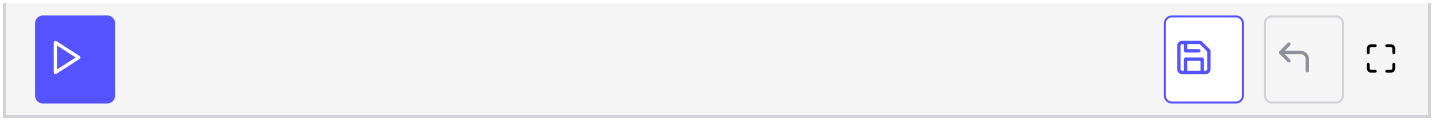
Access a value

Example

The following example declares a **tuple** `struct` named `FruitQuantity`.

```
//define a tuple struct
struct FruitQuantity(String, i32);
// main function

fn main() {
    // create an instance
    let r1 = FruitQuantity("oranges".to_string(), 12);
    // access values of a tuple struct
    println!("r1--name:{} quantity:{}", r1.0, r1.1);
    // create an instance
    let r2 = FruitQuantity("mangoes".to_string(), 13);
    // access values of a tuple struct
    println!("r2--name:{} quantity:{}", r2.0, r2.1);
}
```



Explanation

`main` function

- On **line 6**, a tuple struct `FruitQuantity` is instantiated as `r1` and value is displayed on **line 8**.
- On **line 10**, a tuple struct `FruitQuantity` is instantiated as `r2` and value is displayed on **line 12**

Tuple `struct` `FruitQuantity`

- The tuple `struct` `FruitQuantity` is declared on **line 2**, which can store two items of type `String` and `i32`.

Now that you have learned all about Struct basics, solve a challenge in the next lesson before moving on to the next chapter.