Solution Review 2: Declare a Tuple

This lesson gives a detailed solution review of the challenge in the previous lesson.

We'll cover the following

- Solution:
  - Explanation

Solution:

```
fn test() {
    // define a tuple
    let persons = ("Alex",21, "Abe", 22, "Anna", 23);
    // print the values of tuple
    print!("{}:{}, {}:{}, {}:{}", persons.0, persons.1, persons.2, persons.3, persons.4, persons.5
}
```

Explanation

- On **line 3**, a **tuple,** `persons`**, of size** `6` is defined with values.
- On **line 5**, print the value of the `persons` elements using the dot operator notation (`.`) accessed using placeholder `{}` for each value within the `print!()` macro.

```
fn test(){
    let persons = ("Alex",21, "Abe", 22, "Anna", 23);
    print!("{}:{}, {}:{}, {}:{}", persons.0, persons.1, persons.2, persons.3, persons.4, persons.5);
}
```

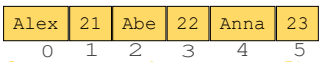
1 of 4

```
fn test(){
    let persons = ("Alex",21, "Abe", 22, "Anna", 23);
    print!("{}:{}, {}:{}, {}:{}", persons.0, persons.1, persons.2, persons.3, persons.4, persons.5);
}
```

Alex	21	Abe	22	Anna	23
0	1	2	3	4	5

Output:

```
fn test(){
    let persons = ("Alex",21, "Abe", 22, "Anna", 23);
    print!("{}:{}, {}:{}, {}:{}", persons.0, persons.1, persons.2, persons.3, persons.4, persons.5);
}
```



Output:

```
Alex:21, Abe:22, Anna:23
```



```
fn test(){
    let persons = ("Alex",21, "Abe", 22, "Anna", 23);
    print!("{}:{}, {}:{}, {}:{}", persons.0, persons.1, persons.2, persons.3, persons.4, persons.5);
}end of program code
```

Output:

```
Alex:21, Abe:22, Anna:23
```


Now that you have learned about data types, let's move on and learn how to perform operations in the next chapter, "Operators".