\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* OOPS \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

1. On DAY-15, we tried the coffee machine. If you check the code, the code was so complex. The functions

are so many and we are using them randomly in same place in the code make other also hard to understand.

2. Hence, we are using this concept called "OBJECT ORIENTED PROGRAMMING" (OOPS).

3. For simple understanding, assume that you are running a hotel. In hotel there are many roles’ people play

to make job easier and simple.

EX: - RECEPTIONIST will welcome the customers and allocate their seats to diners.

- WAITER will take the order and forward it to CHEF

- CHEF will prepare the food to the customer

-CLEANER will clean the tables, floor, utensils

- SERVER will serve the food to them and

- Ultimately MANAGER will manage all of them and runs the restaurant properly.

4. This is how the OOPS CONCEPTS also works. It will make or manage our program and makes things simpler and helps you

to deal with larger and complex programs.

=> HOW TO USE THE OOPS?

- Let’s say we are creating a virtual restaurant

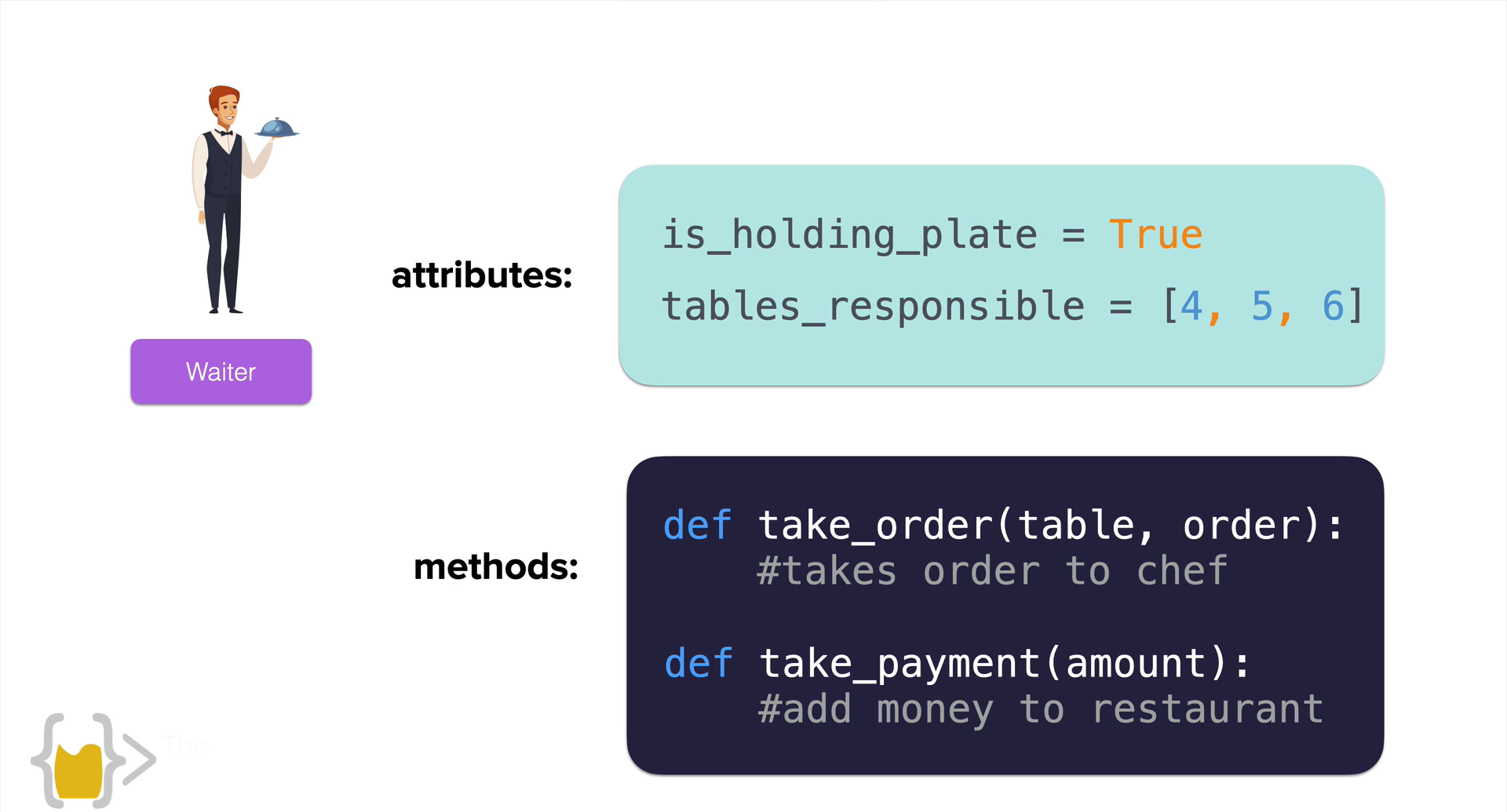
- In that restaurant we need to model the WAITER, CLEANER, CHEF.. etc.

-Let’s say we created a virtual WAITER:

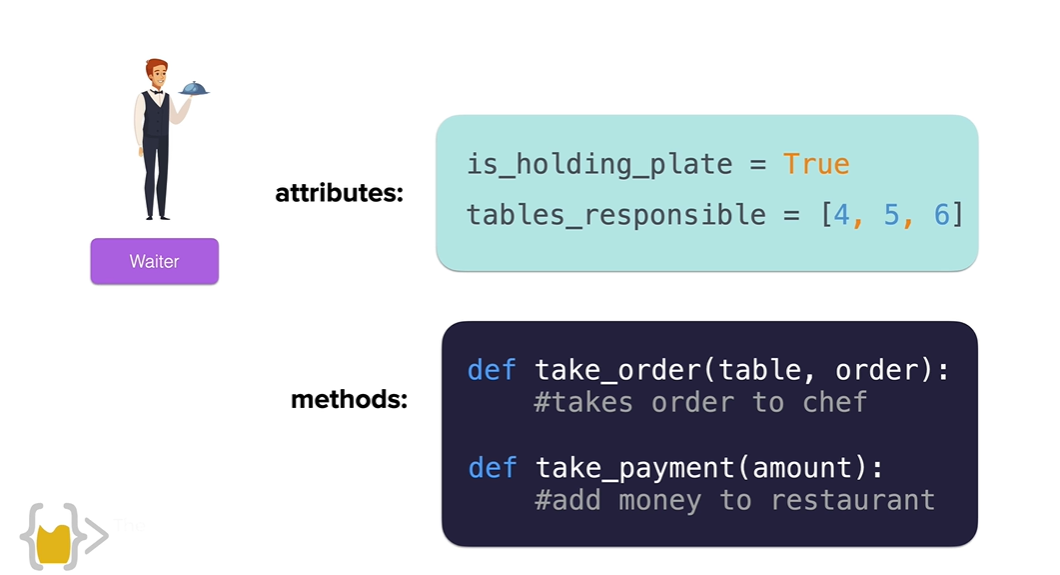
- WHAT A VIRTUAL WAITER CAN DO?

Two things we need to consider.

1. What it HAS?
2. What it DOES?



This we can simple defined as “ATTRIBUTES” and “METHODS”.



* Here, the attributes are nothing but variables attached to the object(WAITER).
* The method is nothing but the function attached to the object. These are modelled by functions. Essentially called as methods
* We can generate multiple versions from the same object or from the same blueprint
* In OOPS, we call it as “CLASS”.
* 