

You are taking candies from a bowl one by one. When you find the chocolate, you want to stop searching. Fill in the code below to make this possible:

candies := []string{"Jellybean", "Chocolate", "Gummies", "Licorice"}

for index, candy := range candies {
 if candy == "Chocolate" {





You got it!

What is the purpose of the for keyword in Go?

The for keyword only declares indefinite loops, but not definite and infinite loops.

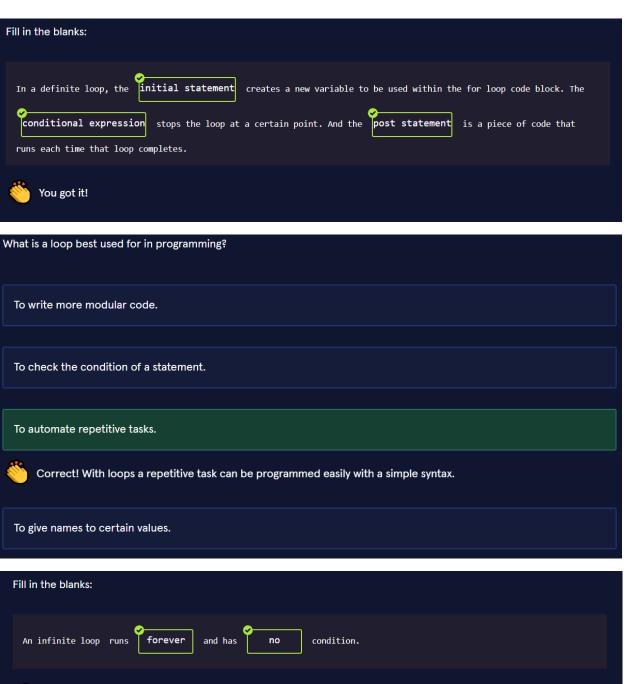
The for keyword only declares definite loops, but not indefinite and infinite loops.

The for keyword is used to declare definite, indefinite, and infinite loops.



Correct! In Go, the language is simplified by using only the for keyword for all types of loops.

The for keyword only declares infinite loops, but not definite and indefinite loops.





You have an address book that maps the names of your friends to their address. Fill in the code below to iterate over this map and print your friend's information.

addressBook := map[string]string{
 "Jannet": "22 Water St",
 "Joe": "241 North Rd",
 "Robert": "86 Stone St",
}

for name, address := range addressBook
{
 fmt.Println(name, "lives at", address)
}



You got it!

