Solution Review: Summing the Integers

This lesson discusses the solution to the challenge given in the previous lesson.

```go
package main
import (
	"fmt"
)

func sum(x, y int, c chan int) {
	c <- x + y
}

func main() {
	c := make(chan int)
	go sum(12, 13, c)
	fmt.Println(<-c)  // 25
}
```

Summing Integers

The function `sum` expects the *two* integers, and a channel to put the resulting `sum` on (**line 6**). So, at **line 11** in `main()`, we make a channel `c`. At **line 12**, we call `sum` with *two* integers and this channel as parameters. At **line 13**, `main()` tries to get a value from channel `c` to print it. This line blocks, until the goroutine in which sum operates, puts its return value on the channel at **line 7**. Only then, it can continue. That's why we made `main()` wait until the completion of `sum()`.

That is it about the solution. In the next lesson, we'll discuss the semaphore pattern, which was mentioned earlier.