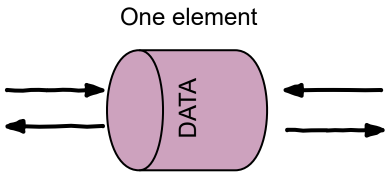
Buffered vs. Unbuffered Channels as Illustrations

Illustrations of buffered and unbuffered channels to explain the differences between the two.

Unbuffered Channels

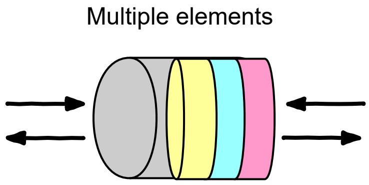
We've been using unbuffered channels in all our previous illustrations. What makes them unique is that only one piece of data fits through the channel at a time.



Unbuffered Channel

Buffered Channels

Buffered channels work just like unbuffered channels, but with one catch—we can send multiple pieces of data to the channel without waiting for another Go routine to read from it.



Buffered Channel

Buffered channels can be useful in concurrent programs as we could run into a situation where one Go routine can perform multiple tasks in the time it takes another routine to perform a single task. In order to make sure the former routine doesn’t have to wait until the latter finishes, we can use a buffered channel.