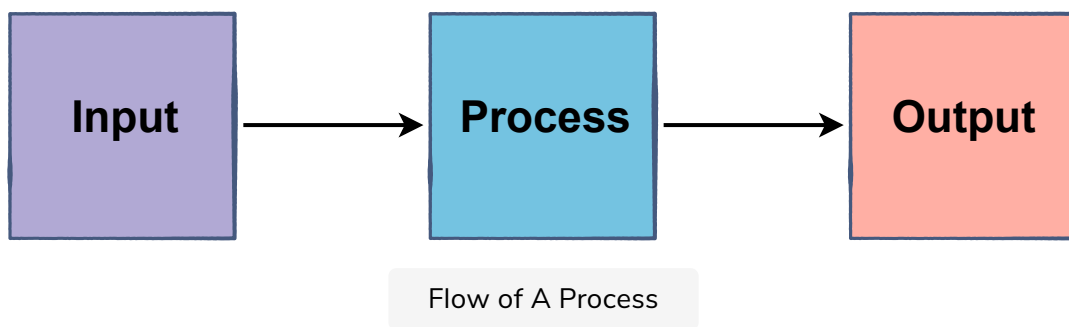


Communicating Sequential Processes

In this lesson, we will briefly discuss Communicating Sequential Processes (CSP).

For a concurrent program, concurrent processes have to operate individually but with a shared data source. However, this leads to problems such as race conditions, which we will discuss in the next lesson. Hence, Tony Hoare came up with an effective solution in 1978, i.e. communication between concurrent processes. He put forth the idea of communication in concurrency in a paper titled **Communicating Sequential Processes**. Have a look at it [here](#). This communication allows us to give a better structure to our concurrent approach and is a simpler solution than using locks, semaphores, etc.

According to this paper, processes have been conceptualized as individual blocks of logic which take in some input and give out some output.



Think of this concept in terms of concurrency. We can say that concurrent processes can synchronize and coordinate with each other by communicating about their input and output. This is what is suggested in the paper. The paper covers two main concepts:

1. Synchronization
2. Dijkstra's Guarded Commands

The paper proposes a formal language for carrying out communication between concurrent processes. If you want to go in more detail regarding the syntax of the language for patterns of interaction between concurrent processes, you should definitely read the paper.

processes, you should definitely read the paper.

"Don't communicate by sharing memory; share memory by communicating"

Go, as well as Erlang and Limbo, has been highly inspired by the concept of communicating sequential processes. Golang makes use of **channels** to achieve communication between concurrent processes. We'll cover channels in detail in the next chapter.

Furthermore, CSP made a breakthrough in Computer Science especially in the field of concurrency. The paper is a must-read for any Computer Science enthusiast.

Now that you've understood the basics of CSP, let's move on to studying data races and race conditions in the next lesson!