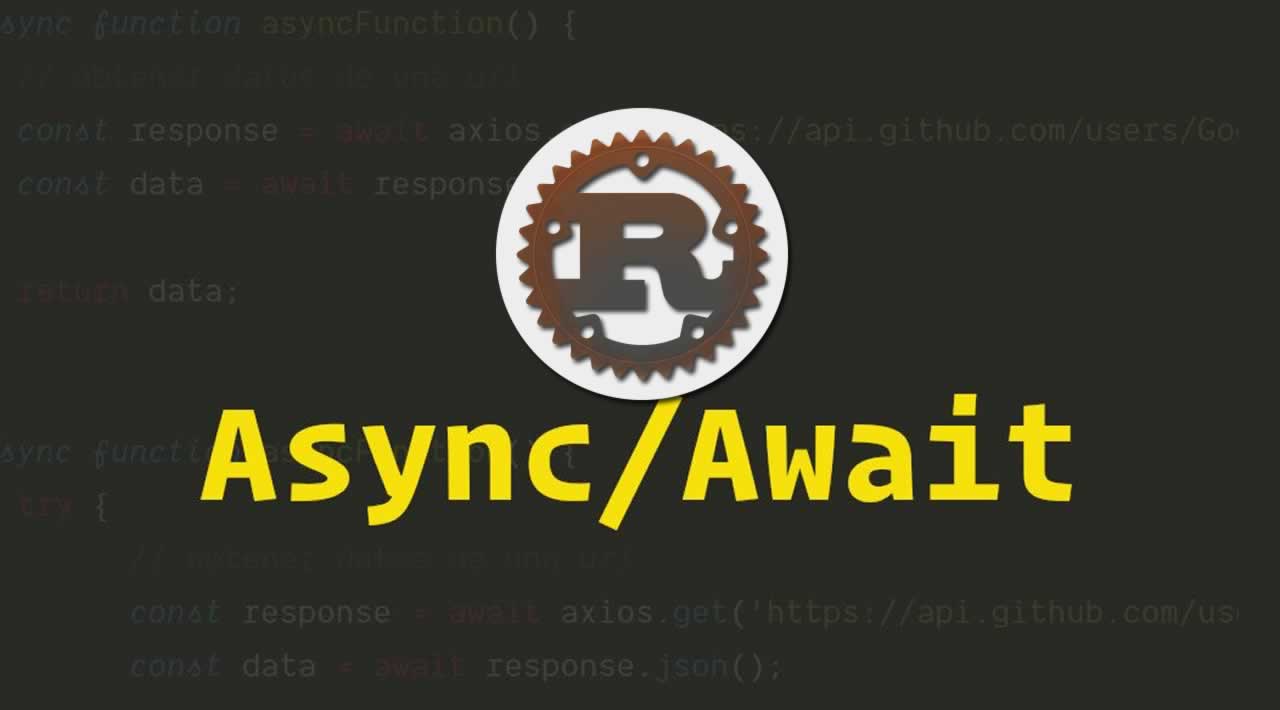
**An Article on Async Rust**



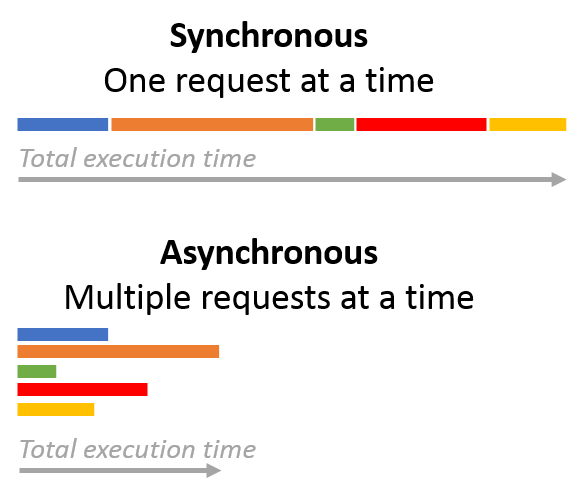
**What is async?**

Async is short for “asynchronous”. Async is a mean of running code concurrently. Also, it is meant to be multiple operations running in a time on a same OS thread.

In other words, Asynchronous programming is a parallel programming in which a unit work of an application run separately from the main application, and notifies to the calling thread after completion.

**Why Async?**

Async programming allows us to run multiple of these IO bound computations at a time on a single thread. Or we can say that, an asynchronous code let us to run several tasks simultaneously on the same OS thread.



**Async Rust**

Asynchronous coding in Rust Programming Language may be little bit different as it done in other programming languages like C# or JavaScript. In Async Rust, there is fearless concurrency while running multiple operations.

In Rust an Asynchronous function starts by .await or by launching a task using an executor.

But the standard library doesn’t come with an executer so, we need an external library to run futures. The executor takes care of executing the futures, polling them and returning the results after completion.

**What is meant by Future?**

In Rust, async fn creates an asynchronous function which returns a Future. To execute the body of the function, the returned Future must be run to completion. They aim to break code into small, composable actions that can be executed by a part of our system.