**Async in Rust:**

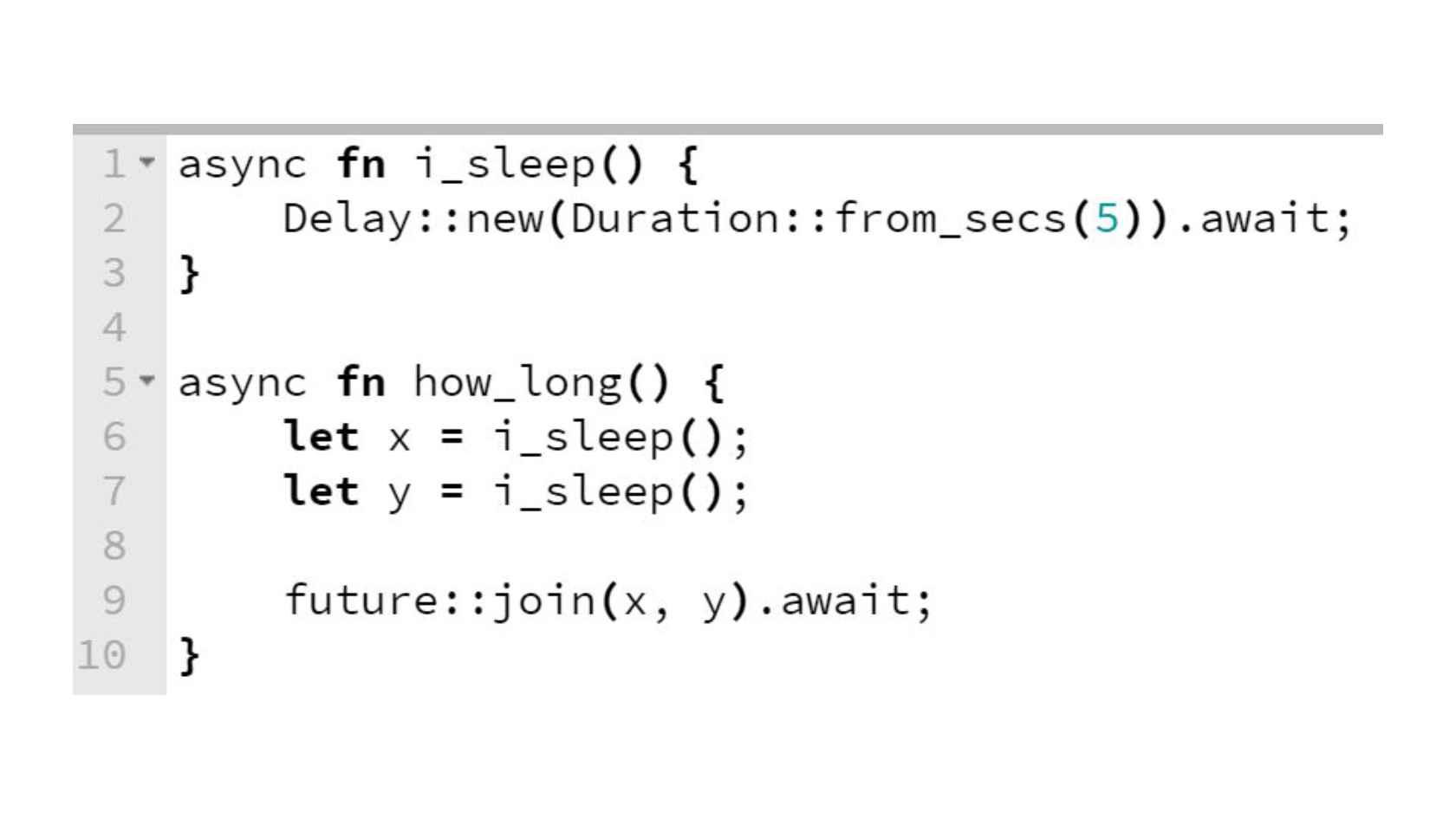
Async program is a form of parallel programming, its allows the atomic work to run separately from the primary (main) application thread. When the atomic work is complete, it notifies the main thread with failed or success status. Moreover, Asynchronous code allows us to run multiple tasks concurrently on the same OS thread.

**Concept:**

Async program basically is a concept which allows not blocking the program workflow when waiting for the results of certain actions.

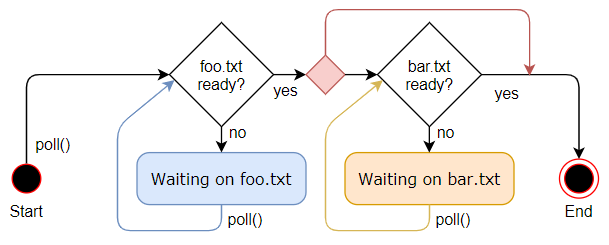
**Async Code Example:**

Async functions require special support from the language or libraries. In Rust, “**async fn**” creates an Async function which returns a Future. To execute the body of the function, the returned Future must be run to completion as mentioned in below code diagram.



**Visual Representation of Async State Machine Basic:**

Below diagram explains how Async works and how Async code differs from traditional Rust programs.



**Conclusion:**

Let’s conclude Async, asynchronous applications have the potential to be much faster and use fewer resources than a corresponding threaded implementation but it has some cost. As we know that threads are natively supported by the OS, and using them doesn't require any special programming model, any function can create a thread, and calling a function that uses threads is usually just as easy as calling any normal functions, but asynchronous functions require some special support from the language or libraries.