# Introduction to Tasks

This lesson gives an introduction to tasks (consisting of promises and futures) which are an important part of multithreading in C++.

In addition to threads, C++ has tasks to perform which work asynchronously. These tasks need the `<future>` header, and they are parameterized with a work package. Additionally, they consist of two associated components: a promise and a future; Both are connected via a data channel. The promise executes the work packages and puts the result in the data channel; the associated future picks up the result. Both communication endpoints can run in separate threads. What is special is that the future can pick up the result at a later time; therefore, the calculation of the result by the promise is independent of the query of the result by the associated future.

> 🔑 **Regard tasks as data channels between communication endpoints**
>
> Tasks behave like data channels between communication endpoints. One endpoint of the data channel is called the promise, the other endpoint of the data channel is called the future. These endpoints can exist in the same or in different threads. The promise puts its result in the data channel. The future waits for it and picks it up.
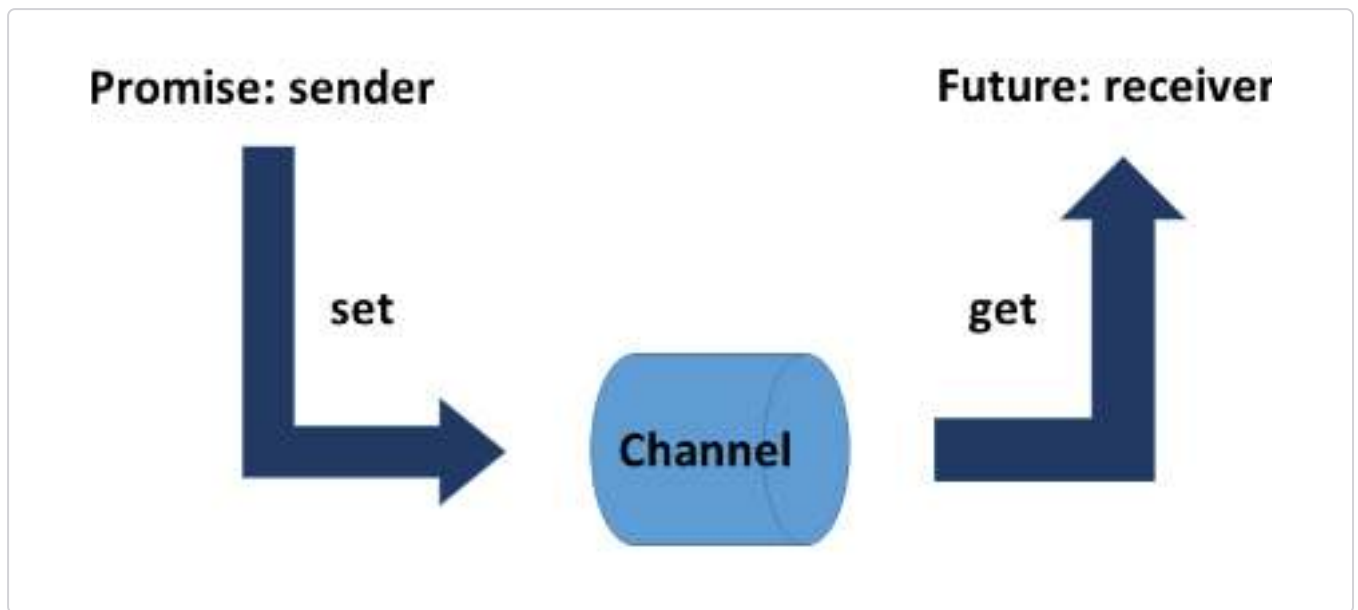
In the next lesson, we'll talk about the differences between threads and tasks used in C++ for concurrency.