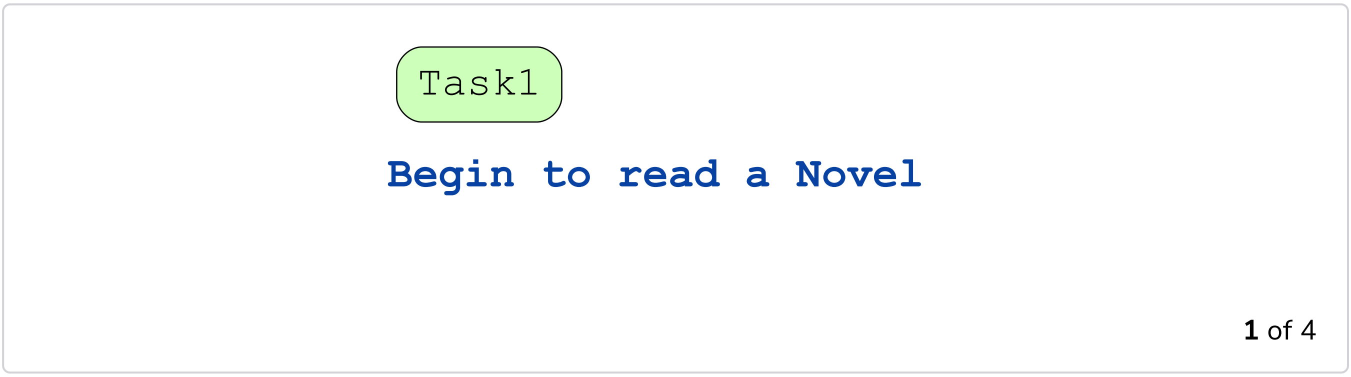
Asynchronous Programming

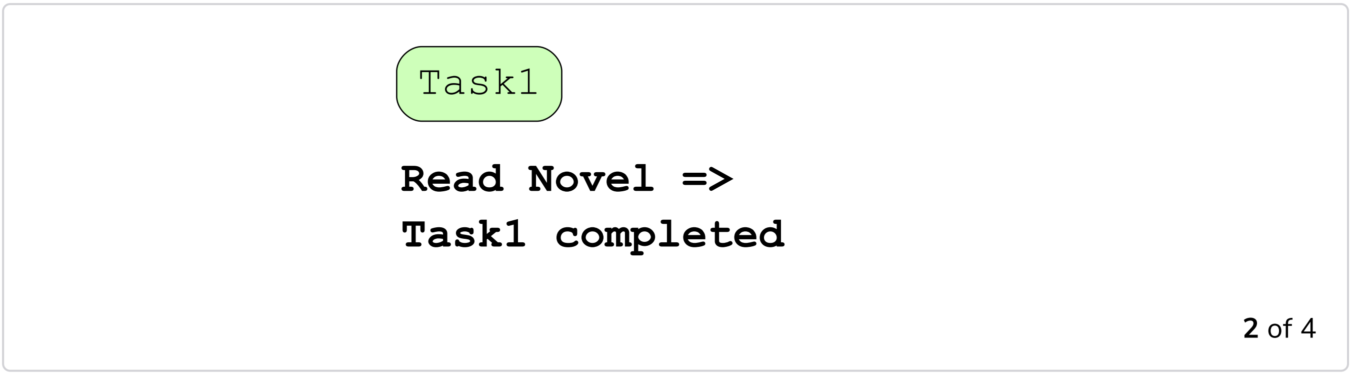
So far we have been doing synchronous programming. In this lesson, we will learn about what is asynchronous programming and why it is important.

WE'LL COVER THE FOLLOWING

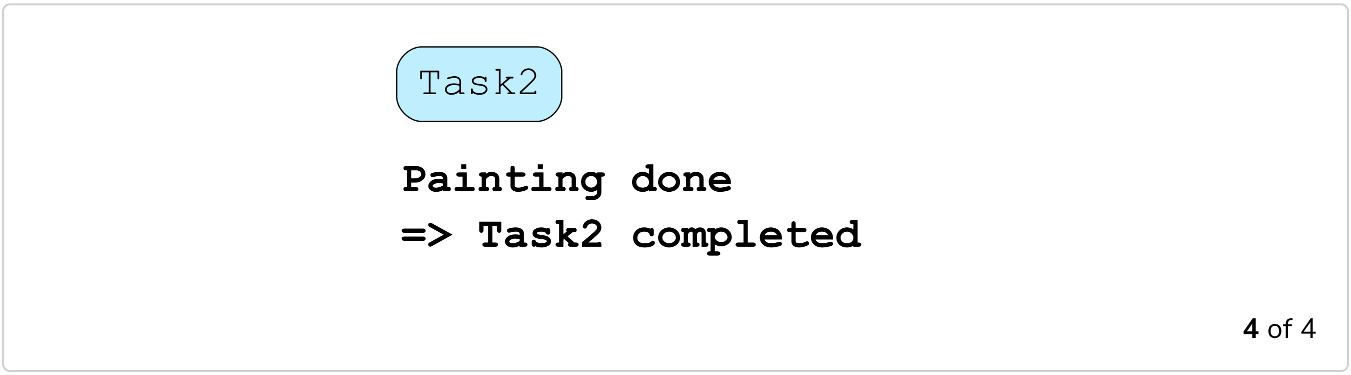
Synchronous Programming

Synchronous program execution is quite simple: a program starts at the first line, then each line is executed until the program reaches the end. Each time a function is called, the program waits for the function to return before continuing to the next line.





Task2
Start Painting
3 of 4



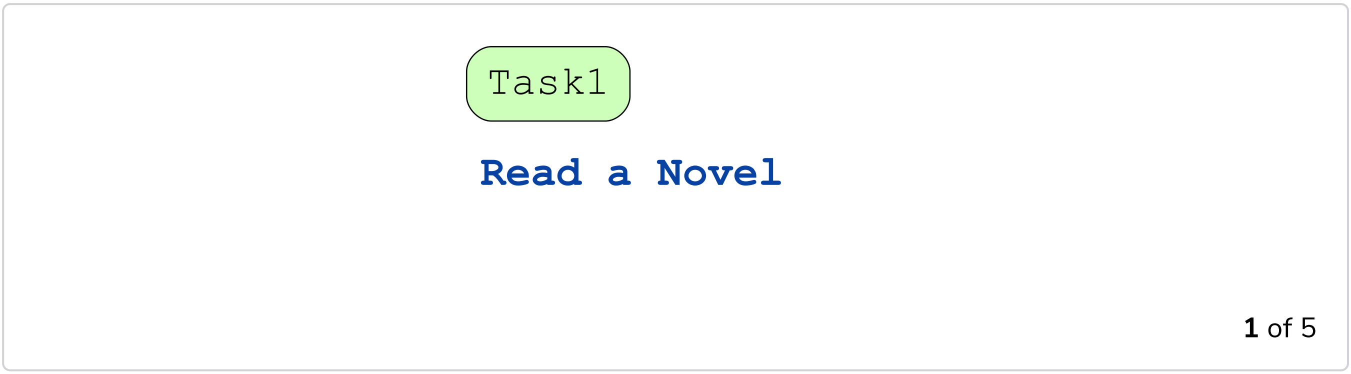
Why Asynchronous Programming?

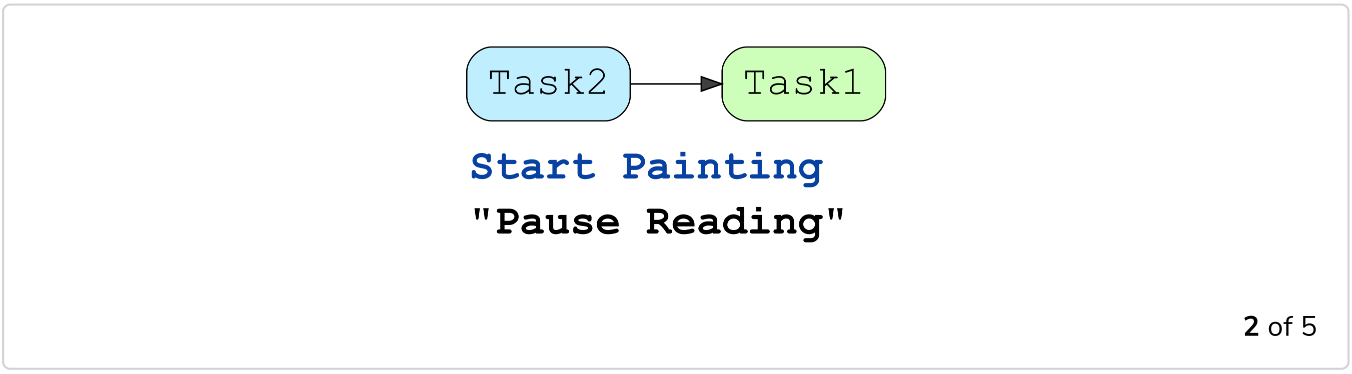
Asynchronous programming allows you to write concurrent code that runs in a single thread.

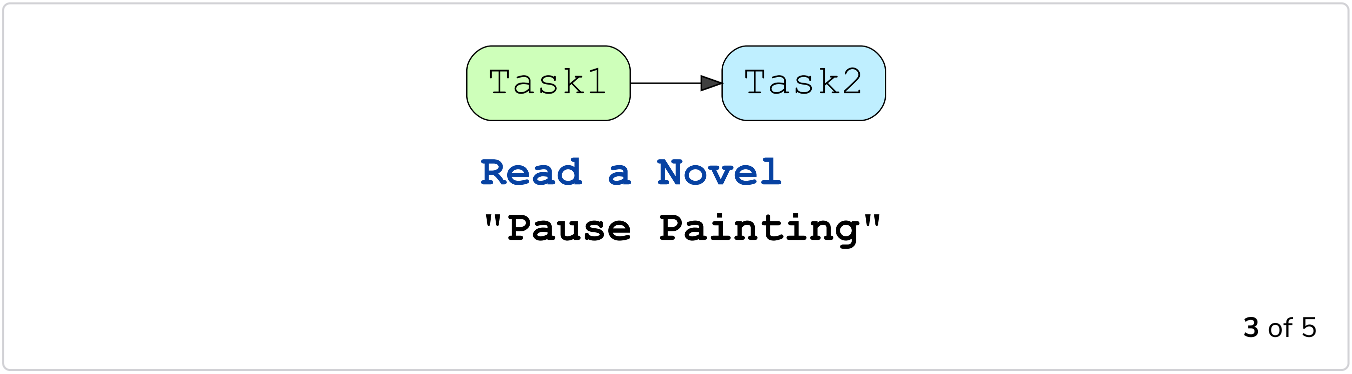
Concurrency

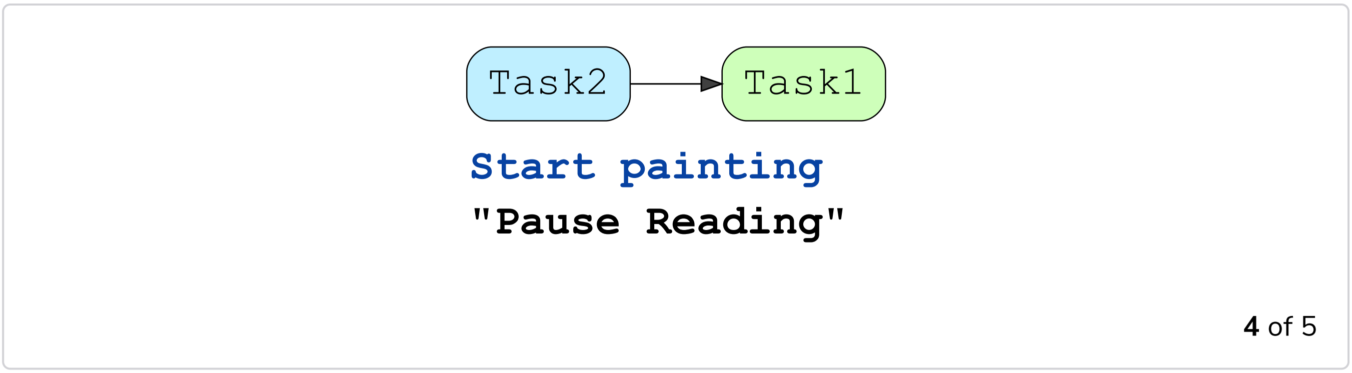
Imagine you are reading a novel while painting. Even if it seems like you are doing both tasks at the same time, what you are doing is switching between the two tasks; while you wait for the paint to dry you are reading your novel, but while you are painting you pause your reading. This is called concurrency.

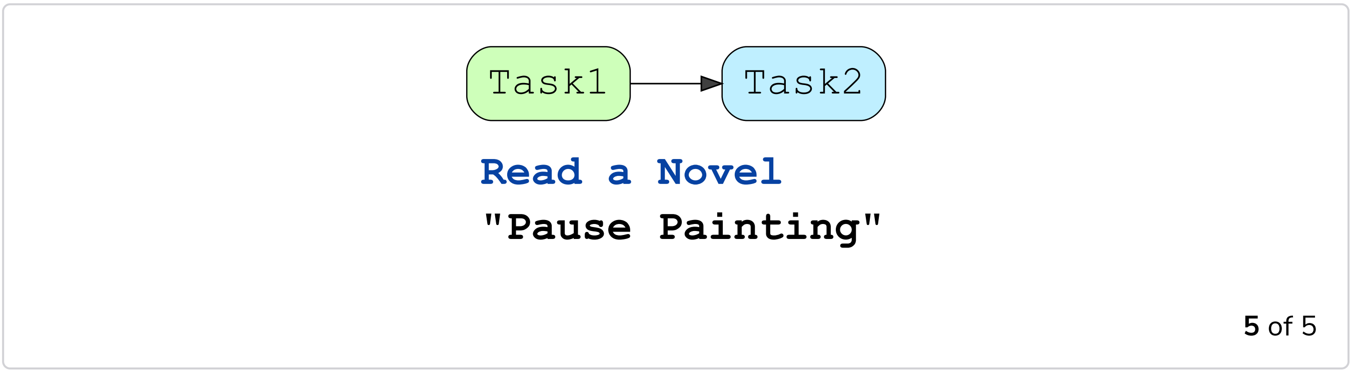
The illustration below shows the context switching between the tasks.











Single thread allows you to decide where the scheduler will switch from one task to another, which means that sharing data between tasks is safer and easier.

Minimum Memory Usage

Every time a new thread is created, some memory is used to allow context switching. If we use async programming, this is not a problem since the code runs in a single thread.

Let’s learn about the components of an asynchronous code in the next lesson.