





# Introduction

 This  introduces the concept of asynchronous programming.

## Introduction

Concurrency can be defined as dealing with multiple things at once. You can concurrently run several processes or threads on a machine with a single CPU but you'll not be parallel when doing so. Concurrency allows us to create an illusion of parallel execution even though the single CPU machine runs one thread or process at a time.

Parallelism is when we execute multiple things at once. True parallelism can only be achieved with multiple CPUs.

A machine with four cores and running one hundred processes is both parallel and concurrent but mostly concurrent because it can only do four things at a time.

## Async.io

async.io stands for asynchronous input output and refers to a programming paradigm which achieves high concurrency using a single thread or event loop. The model is not novel to Python and is implemented in other languages and frameworks too. Most prominent being JavaScript's NodeJS.

## Example



To understand the concept behind `async.io` let's consider a restaurant with a single waiter. Suddenly, three customers, Kohli, Amir and John show up. The three of them take a varying amount of time to decide what to eat once they receive the menu from the waiter. Let's assume Kohli takes 5 minutes, Amir 10 minutes and John 1 minute to decide. If the single waiter starts with Amir first and takes his order in 10 minutes, next he serves Kohli and spends 5 minutes on noting down his order and finally spends 1 minute to know what John wants to eat then in total, he spends  $10 + 5 + 1 = 16$  minutes to take down their orders. However, notice in this sequence of events, John ends up waiting 15 minutes before the waiter gets to him, Kohli waits 10 minutes and Amir waits 0 minutes.

Now consider if the waiter knew the time each customer would take to decide. He can start with John first, then get to Amir and finally to Kohli. This way each customer would experience a 0 minute wait. An illusion of three waiters, one dedicated to each customer is created even though there's only one. Lastly, the total time it takes for the waiter to take all three orders is 10 minutes, much less than the 16 minutes in the other scenario.

Folks with JavaScript background would find `async.io` very similar to how NodeJS works. NodeJS under the hood has a single-threaded event loop that serves all incoming requests.

## Read Me

Advance readers may wish to skip the next few chapters if they are already well-versed with the material that they cover. Upcoming chapters discuss preliminary topics for working with `async.io`.

[← Back](#)[Next →](#)

Miscellaneous Functions

Iterables

☒ Mark as Completed[Report an Issue](#)[Ask a Question](#)

([https://discuss.educative.io/tag/introduction\\_\\_asyncio\\_\\_python-concurrency-for-senior-engineering-interviews](https://discuss.educative.io/tag/introduction__asyncio__python-concurrency-for-senior-engineering-interviews))