[Course Overview](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live" \t "psplayer)

[Course Overview](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live)

[Hi everyone, my name is Tim Ojo. I'm a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=0) [software engineer at trivago, and I'd like](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=6.96946153846154) [to welcome you to my course, Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=11.062538461538463) [Concurrency Getting Started. Despite](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=13.385461538461541) [Python's popularity and its use in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=14.897) [multiple areas from desktop and web](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=17.992999999999995) [development to scientific computing, many](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=20.61766666666667) [still regard it as a single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=22.019333333333343) [language. In this course, we're going to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=23.697) [learn how to execute tasks concurrently](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=25.628999999999998) [using pure Python. Some of the topics](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=28.083999999999996) [we'll cover include a Python-threading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=31.415400000000005) [API, using multiprocessing to get around](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=33.5459) [the GIL, the new asyncio module, and how](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=36.21563636363638) [to choose the appropriate concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=39.55709090909092) [mechanism for the task. The only prior](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=41.179818181818206) [knowledge needed for this course is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=43.13909090909091) [basic understanding of Python. And by the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=44.681636363636365) [end of this course, you'll be able to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=47.45207142857142) [develop faster and more efficient](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=50.06292857142855) [concurrent apps in Python. I hope you'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=51.913) [join me on this journey to learn Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=54.368769230769225) [concurrency with the Python Concurrency Getting Started course at Pluralsight.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=6dfa95c0-71e0-4a79-95d1-d2c601b9b7af&clip=0&mode=live&start=56.27215384615382)

[Concurrency Concepts](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live)

[Introduction to the Course](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live)

[Welcome to the Python Concurrency Getting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=1) [Started course. My name is Tim Ojo. And in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=4.3165000000000004) [this course, we will explain the forms and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=8.711) [use cases of concurrency, take a look at](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=10.191000000000004) [the thread-based concurrency mechanisms](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=13.83933333333333) [available in Python, use how to use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=16.314) [process-based concurrency to overcome](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=18.8109) [limitations CPython has concerning](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=20.377) [threads, examine the concurrent.futures](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=22.360999999999994) [high-level concurrency API, which was](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=25.844428571428566) [provided to simplify the execution of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=28.7138) [parallel tasks, and finally we'll examine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=30.600199999999994) [the asynchronous programming capabilities](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=33.672999999999995) available from Python 3.5 on.

[But before](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=36.117) [we dive in, let's take a quick detour to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=41.05342857142857) [talk about **Moore's Law** as motivation for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=42.913857142857125) [this course. In 1965, Gordon E. Moore, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=46.955999999999996) [director of research and development for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=51.48399999999998) [Fairchild Semiconductor, predicted in an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=54.301) [article in Electronics Magazine that the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=55.99474999999999) [**number of transistors in a dense**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=58.391000000000005)[**integrated circuit would continue to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=61.15100000000001)[**double approximately every two years.** This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=63.31637500000001) [has been paraphrased to mean that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=65.71849999999999) [processor speeds or overall processing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=67.75066666666666) [power will double every two years. Gordon](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=69.74899999999997) [Moore went on to become the cofounder of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=72.51658333333334) [Intel in 1968, and his prediction went on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=75.47325000000005) [to be popularly known as Moore's Law.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=78.90283333333335) [Moore's Law held true for several decades](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=82.152) [and has been used by the semiconductor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=83.64784615384619) [industry as a guide for longterm planning](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=85.25272727272728) [and as a target for research and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=87.51181818181823) [development. Many of the advancements in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=91.1256) [electronics we have today are strongly](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=92.99654545454545) [linked to Moore's Law. The development of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=94.448) [chips that are increasingly cheaper,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=97.74690909090913) [smaller, more powerful, and more energy](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=100.19981818181819) [efficient has led to all of the modern](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=103.60672727272728) [technology we know today. From the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=106.44900000000001) [internet itself, to better electronic](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=108.12249999999999) [devices, modern healthcare,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=110.87874999999997) [transportation, social media, data](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=112.97) [analytics, and the list goes on.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=116.04999999999998)

[But now Moore's Law is slowly dying. In 2015,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=119.06566666666667) [Gordon Moore stated that he believed that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=122.7185) [the rate of progress would soon reach](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=125.73766666666667) [saturation. And Intel announced that same](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=128.28158333333334) [year that the pace of advancement had](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=130.6607692307693) [slowed. To keep up with Moore's Law today,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=133.48284615384628) [chip makers are turning to multi-core](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=136.63950000000008) [CPUs. That is CPUs with two or more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=138.99964285714302) [independent processing units that can read](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=141.94100000000003) [and execute instructions. So what does](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=144.1735) [this mean for us developers? It means that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=146.64849999999996) [today, more so in the future, we cannot](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=149.50923076923078) [make our programs perform better and solve](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=151.63784615384617) [harder problems by simply running them a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=154.46250000000003) [newer, faster hardware. In order to make](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=157.56849999999997) [our applications run faster and to evolve](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=160.4157499999999) [the way hardware is evolving, we must](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=163.1646666666667) [employ concurrency techniques to allow](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=165.83983333333344) [them to take full advantage of multi-core](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=167.15720000000002) [hardware by executing instructions on multiple cores simultaneously.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=0&mode=live&start=169.56)

[Concurrency Concepts](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live)

[Now that we've seen why concurrency is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=1.574) [important, let's talk about what](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=3.0280909090909085) [concurrency is. Concurrency is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=4.475888888888889) [execution of multiple instruction](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=7.560333333333334) [sequences at the same time. This is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=9.771727272727272) [possible when the instruction sequences to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=12.29681818181818) [be executed simultaneously have a very](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=14.723666666666666) [**important characteristic**, which is that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=17.413) [they are **largely independent** of each](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=19.075222222222223) [other. This characteristic is important](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=21.092666666666666) [both in terms of the **order of execution**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=22.81183333333333) [and in the use of **shared resources**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=25.912799999999997)

[In terms of the order of execution, this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=29.457384615384616) [means that the order of execution of these](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=31.70007692307692) [instruction sequences should have no](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=34.58415384615385) [effect on the eventual outcome. If task 1](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=36.72107692307694) [finishes after tasks 2 and 3, or if task 2](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=40.04560000000001) [is initiated first, but finishes last, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=44.00426666666671) [eventual outcome should still be the same.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=46.76058333333332)

[In terms of shared resources, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=49.511) [different instruction sequences should](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=52.338333333333345) [share as few resources between each other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=53.988800000000005) [as possible. The more shared resources](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=55.64640000000002) [that exist between concurrently executing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=57.805777777777784) [instructions, the more coordination is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=59.913) [necessary between those instructions in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=63.456124999999986) [order to make sure that the shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=66.042) [resource stays in a consistent state. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=67.65200000000003) [coordination is typically what makes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=70.09577777777777) [concurrent programming complicated.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=72.288)

[However, we can avoid many of these](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=73.48366666666668) [complications by choosing the right](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=76.13466666666666) [concurrency patterns and mechanisms](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=77.43299999999996) [depending on what we are trying to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=78.71333333333334) [achieve. This course will help show you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=82.22266666666668) [many of the concurrency patterns and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=84.07100000000004) [techniques offered in Python, and guide](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=85.418) [you on how to choose and use them to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=86.95676923076923) [achieve your goals. But before we dive](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=89.748) [into Python, we should get a general](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=93.23400000000004) [understanding of the different types of concurrency in use today.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=1&mode=live&start=94.9027272727273)

[Types of Concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live)

[The two main forms of concurrency that we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=1.211) [will focus on in this course are **parallel**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=3.395000000000001) [**programming** and **asynchronous programming**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=6.285666666666667) [It should be noted that distributed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=8.791) [computing also offers to us another form](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=10.214636363636359) [of concurrency. However in this course, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=12.163181818181819) [will be focusing our efforts on currency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=14.830818181818186) [that operates on the level of a single](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=18.061571428571433) [machine. Parallel programming involves](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=22.011285714285727) [taking a computational task and splitting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=23.74625) [it into smaller subtasks that are then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=26.214909090909092) [assigned to **multiple threads or processes**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=29.01427272727273) [to be executed on multiple processor cores](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=31.55254545454545) [simultaneously. With single-threaded code,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=34.67581818181819) [if you have multiple processor cores on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=36.350199999999994) [your system, only one core will be charged](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=39.216) [with executing your task, while the other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=41.35266666666666) [cores sit idle or execute instructions for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=43.695750000000004) [other programs. With parallel programming,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=46.67016666666668) [all your processor cores can be engaged,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=48.5376) [theoretically cutting your processing time](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=50.912) [by a factor of the number of cores that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=52.82336363636364) [you have available. Because it's designed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=57.195) [to take advantage of multi-core systems,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=60.16733333333334) [**parallel programming is best suited for**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=61.956)[**tasks that are CPU intensive.** That is,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=63.619636363636346) [tasks in which most of the time is spent](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=65.60757142857142) [solving the problem rather than reading to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=68.3146428571428) [or writing from a device. Another term for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=70.68988888888892) [those types of tasks is that they are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=73.79375000000002) [**CPU-bound tasks**, which means that they](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=77.16108333333338) [will perform better if you could get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=78.36074999999998) [better performance out of the CPU.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=79.451) [Examples of such workloads are string](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=82.65554545454543) [operations, search algorithms, graphics](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=85.92583333333333) [processing, any number crunching](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=88.32516666666668) algorithm, and more.

[If you have a task in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=91.24549999999999) [which most of the time is spent reading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=94.68837499999998) [from or writing to a device, which is more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=96.18487499999995) [commonly known as performing input or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=98.31074999999996) [output, in other words, IO, then that task](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=100.06106666666666) [is more suited for asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=104.05359999999997) [programming. **Asynchronous programming's**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=106.5489333333333)[**concurrency model works by delegating a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=107.67849999999999)[**subtask to another actor such as another**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=109.57)[**thread or device**, and then continuing to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=112.53589999999997) [do other work instead of waiting for a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=115.16823076923076) [response. When the subtask is completed,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=117.89069230769228) [the actor then notifies the main thread,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=120.19099999999997) [which calls a function to handle the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=122.837) [results. This is usually known as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=124.89499999999998) [callback function. In some languages,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=128.46724999999998" \t "psplayer) [instead of executing a callback, the main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=130.54775) [thread is given an object that represents](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=132.85250000000002) [the results from a not-yet-completed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=136.00145454545452) [operation. This object is typically called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=138.49190909090905) [a future, promise, or a task, and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=140.33363636363632) [application can either wait on its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=142.59692307692305) [completion or check back at a later time.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=144.5496923076922) [Examples of **IO-bound tasks** are database](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=148.745) [reads and writes, web service calls, or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=150.34099999999995) [any form of copying, downloading, or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=153.2314444444445) [uploading data either to disk or to a network.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=2&mode=live&start=156.847)

[Concurrency in Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live)

[Python supports both parallel and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=1.31) [asynchronous programming natively. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=4.354375000000001) [**threading** module was introduced way back](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=6.432727272727273) [in **Python 1.5**, and allows you to create](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=7.943090909090908) [thread objects that are mapped to native](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=10.316250000000004) [operating system threads, and can be used](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=12.229) [for the concurrent execution of code.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=14.73823076923077) [However, it should be noted that **in**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=16.889) **[CPython](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=19.119846153846165" \t "psplayer)**[, which is the most common](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=19.119846153846165" \t "psplayer) [implementation of the Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=21.032) [specification, **threads are limited to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=22.167999999999996) [**executing Python code serially by a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=23.588)[**mechanism known as the Global Interpreter**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=26.046615384615393)[**Lock**. Therefore with threading in Python,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=28.505230769230785) [concurrency is limited to what you get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=30.597272727272735) [when the operating system switches between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=33.1345) [threads. We'll discuss this in more detail](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=36.24550000000002) in the next module.

[The **multiprocessing**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=38.597000000000016) [package is an implementation of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=40.71055555555556) [parallelism that uses sub-processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=42.93944444444445) [instead of threads. This technique avoids](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=44.64833333333334) [the global interpreter lock and allows](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=47.10033333333335) [Python to take advantage of multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=49.20399999999999) [processor cores. The multiprocessing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=51.4575) [package was introduced in **Python 2.6** and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=53.69150000000001) [has an API that is similar to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=57.05966666666665) [threading module, but also introduces some](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=59.767933333333296) new APIs.

[The **concurrent.Futures** module](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=62.579909090909105) [was introduced in **Python 3.2**, and it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=65.78227272727275) [provides a common high-level interface for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=68.47581818181817) [using thread pools or process pools.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=70.92)

[And finally in Python 3.4, the **asyncio** module](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=74.994) [was introduced as a provisional package to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=78.54600000000002) [enable asynchronous programming. Starting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=81.95749999999998) [with 3.6, the provisional status was](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=83.94392307692308) [removed, and the API was declared to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=86.40038461538465) [stable. However, if you're still running](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=89.53850000000001) [**Python 3.5** and cannot upgrade versions,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=92.30550000000004) [you shouldn't have any problems working](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=95.00179999999997) [with the examples we will be using in this course.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=3&mode=live&start=96.971)

[Introduction of Demo Application](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live)

[Our main example throughout this course](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=1.506) [would be the development of a **thumbnail**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=2.746) [**maker library**. The function of this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=4.578875) [library is that it takes multiple images](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=8.136124999999996) [of a certain size and produces smaller](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=9.910499999999999) [copies of the original image, while](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=11.600181818181818) [keeping the dimensions intact. How is this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=14.649272727272725) [useful, you may ask? Let's say you've](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=17.380571428571425) [built a service or website that has a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=19.01057142857142) [notion of user profiles and allows the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=21.651846153846158) [user to upload a profile picture of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=24.644076923076934) [themselves. A profile picture taken on a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=26.776500000000006) [smart phone camera can easily have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=28.565) [resolution of 3264 x 1836, and a size of 4](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=32.71425) [MB per image. This image is way too big](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=36.76094117647059) [for your site to serve, especially as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=38.71023529411765) [thumbnail, which is usually a small image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=40.846923076923076) [on the top-right corner of the web page,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=42.928615384615384) [or a small image that accompanies](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=46.3665) [comments. Attempting to serve such a large](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=49.05835714285715) [image will lead to very slow load times](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=51.16853846153845) [and high data usage, two things which will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=53.45733333333333) [quickly drive away users from your site,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=56.259999999999984) [especially mobile users. For reference,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=58.712142857142844) [Facebook uses a 200 x 200 image as the max](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=60.50884615384616) [profile image size, a 50 x 50 image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=64.23807692307695) [throughout the website, and a 32 x 32](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=67.25840000000005) [image for comments on threads. Skype uses](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=70.284) [a 96 x 96 image for the chat window](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=73.81400000000001) [profile picture and a 48 x 48 image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=77.881125) [throughout the rest of the app. Similarly,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=80.02962500000001) [Pinterest displays a profile page picture](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=82.20991666666667) [at 165 x 165 and uses a 32 x 32 thumbnail](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=85.62341666666667) [everywhere else. Therefore for our example](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=90.71574999999997) [application, we will follow a similar](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=92.76172727272726) [model, and resize the images offered to us](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=94.70681818181814) [by our users into smaller sizes for use in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=97.98069230769228) different situations.

[A typical workflow](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=102.1092) [would be to download the images from some](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=103.93684615384616) [form of file store like Amazon S3 to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=105.96576923076925) [image or application server, perform the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=110.34081818181816) [resizing, and then re-upload the resized](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=112.99533333333333) [image back to the file store or some other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=114.32833333333333) [location where they can be served.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=117.34063636363634) [However, to keep this example simple, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=119.889) [will only download the images from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=122.51866666666663) [source location, which in our case will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=124.89663636363638) [DropBox, and then perform the resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=126.89518181818185" \t "psplayer) [operation. We won't worry about the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=129.32500000000002) [re-upload for this example, because these](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=131.941) [two operations are sufficient for us to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=133.81566666666671) [illustrate the different forms and use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=136.57576923076925) [cases of concurrency. **The download of the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=139.30992307692316)[**images is an IO-bound task, and the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=141.8752857142857)[**resizing is a CPU-bound task.** For our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=143.97228571428568) [example application, I have arbitrarily](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=146.65766666666664) [picked target image sizes of 200 x 200 for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=148.891) [the large version, 64 x 64 for the medium](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=152.38224999999989) [size, and 32 x 32 for the small size.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=155.93799999999993)

[Let's take a look at the initial code for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=159.526) [the application. The initial code is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=162.12863636363636) [single threaded, and fairly simple, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=164.24809090909093) [straightforward. We will use it as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=166.55972727272734) [starting point in this course, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=168.0987142857142) [throughout the course, we will refactor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=169.44471428571413) [the code to use different concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=170.9042727272728) [mechanisms to speed up parts of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=172.425) [application. This model of starting with a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=175.38425000000007) [single-threaded program and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=177.48760000000001) [refactoring it is good practice in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=178.608) [general, because it allows you to ensure](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=180.47349999999994) [that your code works properly, debug it,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=182.8988) [handle exceptions, etc., before](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=186.81739999999996) [introducing the complications that can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=188.5272) [inherent in running code in multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=190.29779999999994) [threads. The starter code for this course](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=192.27933333333337) [is hosted on GitHub at the URL showed on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=194.71416666666676) [your screen. If you're not currently a Git](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=198.41960000000006) [user, you can simply download the code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=200.93623076923078) [from the site. If you are a Git user, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=202.893) [you have the option to clone the repository.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=4&mode=live&start=205.75966666666662)

[Demo Code Walkthrough](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live)

[I tend to rely on the command line for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=1.065) [working with Git, but GUIs work just as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=2.854714285714286) [well. The clone command will be *git clone*,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=5.284076923076923) [and then the URL with. git appended to it.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=9.088846153846152) [I'm going to paste in the URL here, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=13.827374999999996) [hit Enter. Done. Before running the code,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=18.44325) [the first thing to do will be to check](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=20.623642857142855) [what version of Python 3 you're running by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=22.702) [typing in the command python3--version. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=25.89866666666666) [you are running on Windows, the command](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=28.25809090909091) [may be python--version instead. As of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=30.921909090909097) [release of this course, the most recent](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=36.31744444444445) [version of Python is 3.6, but for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=41.01879999999999) [purposes of this course, any version of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=44.00019999999997) [Python from 3.5 onwards is acceptable. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=47.18018181818181) [your Python version is 3.4 or less, you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=51.28770588235294) [will need to upgrade your Python version.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=53.82305882352938) [A quick internet search will show you how](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=55.795) [to do this on whatever platform you're](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=57.791923076923055) [running on. If you do not have the luxury](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=59.681000000000004) [of upgrading your Python version, with the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=62.56550000000003) [exception of some of the asyncio code,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=64.53981818181819) [most of the examples in this course will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=66.46107692307692) work for you.

[Once you have your Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=68.63338461538456) [version squared away, the next step will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=70.76146153846152) [be to install the Python libraries](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=72.60030769230765) [required for this project. We have only](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=75.97966666666667) [two for this project, **Pillow** and **pytest**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=79.8614) [Pillow is a well-maintained fork of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=83.101) [Python imaging library and provides us](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=84.76254545454547) [with tools to manipulate images. And](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=86.66371428571428) [pytest is a unit test runner that we'll be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=89.37257142857143" \t "psplayer) [using to run our ThumbnailMakerService.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=92.33871428571427) [Both of these libraries are included in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=93.98811111111111) [the requirements.txt file. So you can go](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=96.3058888888889) [to your shell, and install them using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=100.24900000000004) [pip command, pip3 or pip if you're on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=104.6866) [Windows, install -r requirements.txt. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=110.053) [you don't have these libraries installed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=113.6216923076923) [already, it may take a few moments](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=114.90984615384612) [depending on how fast your computer and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=116.631) [network connection is. Since I already](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=119.66200000000005) [have these libraries installed, it lets me](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=121.99650000000001) know that.

[Now let's explore the code. In](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=124.41266666666671) [thumbnail\_maker.py, we define the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=128.71118181818179) [ThumbnailMakerService class. Its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=131.718" \t "psplayer) [initializer takes in the home directory of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=132.619125) [the service, defaulting to the current](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=134.929) [directory if nothing is passed in. In the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=137.35299999999998) [body of the initializer, we set the input](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=139.97715384615378) [directory and the output directory. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=142.76916666666668) [input directory is where we will download](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=146.3275) [images to be resized, and the output](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=147.67499999999995) [directory is where we will place the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=150.02192857142856) [resized images on completion. Our next](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=153.70142857142855) [method is the download images method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=156.4806153846154) [which takes in a list of image URLs to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=158.51046153846156) [downloaded. Our first step is to verify](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=162.14628571428574) [that the list is not empty and the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=164.86828571428578) [folder exists. We then simply iterate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=167.69700000000003) [through the img\_url\_list, and download](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=170.43579999999994) [each image one at a time, saving them to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=173.1267999999999) [the input folder using the filename from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=175.81330769230775) [the URL. I added timers to see how long](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=178.22414285714285) [this process takes. Our next method is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=183.03078571428566) [perform resizing method. As usual, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=187.34300000000002) [first start with validating the inputs,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=189.92550000000003) [checking that the input directory is not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=191.8385000000001) [empty and the output directory exists. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=194.9358) [then specify our target sizes. I'm also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=198.4983076923077) [getting a count of the number of files in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=200.9574615384616) [the input directory, so that I can log](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=204.3945333333333) [that later on. Then in a loop, we load](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=207.7555999999999) [each image into memory, and for each](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=210.93900000000005) [target size, we calculate the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=213.18250000000012) [height of the image that allows the image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=215.08216666666664) [to maintain the aspect ratio. This is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=217.75641666666667) [achieved by figuring out the ratio of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=221.0003333333333) [desired width to the current width, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=223.83840000000004) [then multiplying the current height by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=225.62044444444444) [that ratio. Once we have our target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=227.45311111111113) [height, we can then call resize, passing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=229.98036363636362) [in our target height and width at a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=232.65754545454544) [sampling filter. When the resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=235.67790909090897) [completes, we save the image to the output](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=237.45449999999997) [directory using the same name, but with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=240.1784999999999) [the base size appended to the name. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=243.22823076923078) [last method we will take a look at is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=246.67938461538463) [make\_thumbnails method, which is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=249.58323076923088" \t "psplayer) [essentially the public interface for our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=251.36688888888887) [ThumbnailMakerService. It takes in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=254.9755555555555" \t "psplayer) [image URL list and calls the methods](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=256.6985384615384) [needed to complete the resizing process.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=258.66123076923066)

[You may have noticed by now that the code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=260.5186666666666) [is structured into these two methods on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=263.6227777777778) [purpose. The download\_images method is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=267.1982222222222) [primary IO bound, while the perform\_resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=269.1118571428571) [method is primarily CPU bound. We could've](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=272.10499999999996) [easily written our application to download](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=276.1732727272727) [each image and then resize them within the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=278.67309090909094) [same loop, but this will run a file of a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=281.395) [**best practice in Python concurrency**, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=284.339) [**is not to mix different types of**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=285.6775454545454)[**workloads**. Because Python's concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=287.80674999999997) [mechanisms are optimized for either](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=289.8342499999999) [IO-bound or CPU-bound tasks, mixing those](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=292.6125454545454) [two in one operation often reduces the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=296.14163636363617) [efficiency of Python concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=300.04049999999984) [mechanisms. To run our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=302.2184999999997) [ThumbnailMakerService, we will use a test.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=304.28100000000006" \t "psplayer) [In test\_thumbnail\_maker.py, we have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=307.317) [list of URLs to some example images that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=309.42949999999996) [were uploaded to DropBox. In a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=312.91) [test\_thumbnail\_maker method, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=317.0140000000002" \t "psplayer) [initialized our ThumbnailMakerService](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=318.26300000000003) [class, and then called the make\_thumbnails](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=319.1105000000001) method, passing in that list.

[To run the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=321.99499999999995) [code, we will open our terminal](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=326.31210000000004) [application, ensuring we're in the project](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=329.74230000000017) [folder, and then type in the command](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=332.3431666666668) [pytest to execute the test. Pytest will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=335.184" \t "psplayer) [find your test files based on naming](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=338.2090769230768) [conventions and execute your test methods.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=341.3344285714286) [The execution time of this code will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=344.529) [depend on the speed of your internet](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=345.94461538461536) [connection for the downloading of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=347.5073) [images and your processor speed for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=349.6030999999998) [resizing of them. However, by in large,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=353.26014285714285) [the download takes significantly more time](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=357.0016153846153) [than the resizing. Once the test is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=359.6785384615383) [completed, we can take a look at the log](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=361.88100000000003) [file that logged to verify this. So in my](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=363.8250000000001) [run, the download took 27 seconds, and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=367.3391538461537) [resize took 3 seconds. If we go take a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=370.6811333333334) [look at the outgoing folder, we can see](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=373.8317333333336) [the resized images with their width](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=377.15653333333336) [appended to the image name. Now that we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=380.04333333333335) [have the simple synchronous version of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=383.34081818181824) [code, we will learn how to use Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=385.4217272727274) [concurrency to make this code run faster,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=387.3510769230771) [starting with the use of threading in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=389.12372727272725) [next module. But first, let's look back at what we've covered so far.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=5&mode=live&start=392.11354545454526)

[Module Summary](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live)

[We started out using Moore's law as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=1.236) [motivation for why the industry is moving](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=3.4275384615384614) [towards concurrency, and why every](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=5.6287) [developer needs to be familiar with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=7.242200000000001) [concurrent programming. We then defined](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=9.38211111111111) [concurrency and the conditions necessary](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=11.504888888888885) [for concurrency to be possible. We briefly](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=14.031374999999999) [introduced the different types of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=17.373500000000003) [concurrency, and the workloads that they](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=18.739750000000008) [are suited for. We noted the different](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=20.829250000000005) [concurrency mechanisms in Python. And](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=23.494500000000006) [finally, we set up the example that we'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=25.422823529411765) [be using for the rest of this course.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=28.245411764705885) [That's it for module 1. I hope you learned](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=31.068) [something and are excited for the next module in this course, Threading.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m1&clip=6&mode=live&start=33.64476923076924)

[Threading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live)

[Introduction to Threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live)

[Welcome to the Threading module in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=0) [Python Concurrency Getting Started course.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=3.3775) [I'm Tim Ojo. In the last module, we got a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=5.79) [general introduction to the field of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=9.461428571428574) [concurrency and discussed the different](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=11.7558) [forms of concurrency, which includes both](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=13.820300000000003) [parallel and asynchronous programming. In](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=16.36057142857143) [this module, we'll be diving deeper into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=18.510636363636365) [parallelism using threads. We will learn](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=21.413090909090922) [how to use the threading package in Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=23.311375) [to create and execute threads. We'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=25.056375000000006) [discuss the behaviour and structural](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=26.680888888888887) [characteristics of threads. We'll learn](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=29.35033333333333) [how to do thread synchronization and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=31.436444444444444) [inter-thread communication in Python. And](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=32.88977777777778) [since we're in Python, we'll talk about](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=34.76363636363637) [the global interpreter lock and its effect](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=37.619) on the usage of threads.

[So let's get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=39.622166666666665) [started. The predominant method of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=43.73350000000001) [concurrent or parallel programming for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=45.67180000000001) [past couple of decades has been the use of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=47.239) [multiple threads, multithreading. To](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=50.69799999999997) [introduce threads, let's start with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=52.37188888888889) [defining a process. A **process** can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=54.97633333333334) [defined as the **execution context of a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=58.028230769230774)[**running program**. An alternative and maybe](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=60.634384615384626) [more approachable definition is that a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=62.8474) [**process is a running instance of a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=65.05)[**computer program**. While a computer program](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=67.32888888888887) [is a passive collection of instructions in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=69.09781818181817) [the form of code, a process is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=71.054) [construct we use to define the execution](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=72.86542857142857) [of those instructions. Every executing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=74.64511111111112) [process has system resources and a section](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=76.7506666666667) [of a memory that is assigned to it. It](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=79.32666666666664) [also has security attributes in a process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=82.26988888888889) state.

[A process is composed of one or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=84.4951111111111) [more threads of execution. **A thread is the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=86.58845454545452)[**smallest sequence of instructions that can**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=88.91733333333335)[**be managed**, that is scheduled and executed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=90.8888333333334) [by the operating system. A program can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=94.17822222222226) [composed of a single thread of execution](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=97.30676923076926) [or multiple threads of execution. When](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=99.09661538461546) [multiple CPU cores are available, each](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=101.79066666666665) [thread's instructions can be executed at](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=104.72090909090909) [the same time in parallel of multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=105.93236363636359) [cores. If only a single core is available,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=108.0362727272727) [the threads share time on that core. In](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=111.493) [either scenario, the result is that the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=114.52979999999997) [use of multiple threads allows a process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=116.46092307692307) [to perform multiple tasks at once. For](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=118.10161538461536) [example in a media player, one thread can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=122.52776923076922) [be playing the current song, while another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=124.16592307692306) [is figuring out the next song to play and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=126.05025000000002) [downloading it, while again another thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=129.246) [is responding to user clicks and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=130.81619999999995) [navigation. Another example is a web or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=132.8380909090909) [application server that uses a pool of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=136.2509090909091) [threads to respond to multiple requests](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=139.48475000000002) [simultaneously. Each request is handled by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=142.19525000000004) [a thread from the pool. The thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=144.4065) [executes whatever task is assigned, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=146.57683333333335) [then when it is completed, it returns to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=147.80933333333343) [the pool to wait for the next request.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=149.98466666666664) [Multithreading is supported by virtually](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=152.692) [all operating systems and almost all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=154.6236666666667) [programming languages, and Python is no exception.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=0&mode=live&start=158.169)

[Creating Threads in Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live)

[Python has had support for threading since](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=1.249) [version 1.5.2 via the threading package](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=3.0636153846153853) [in the standard library. This package](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=5.570666666666666) [allows you to create thread objects that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=7.914166666666667) [map directly to operating system threads.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=9.454749999999999) [The simplest way to create a thread is to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=12.063) [instantiate an object of the thread class,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=13.94528571428572) [passing in the thread function, as well as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=15.61) [any function arguments, and then calling](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=18.086) [the **start** method on the thread object you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=20.409285714285716) just created.

[In this example, we start by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=23.81785714285715) [importing the threading package, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=26.71540000000001) [we define the function that we want our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=28.635153846153848) [thread to run. All we're doing in this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=30.85976923076924) [function is printing out some text,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=32.71218181818182) [including the passed-in parameter called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=33.934) [val. The interesting part comes in line 7.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=37.09150000000001) [This is where we construct a thread using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=39.97269230769231) [the **threading.Thread** constructor. Here we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=42.699666666666666) [define our target function and any](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=44.856333333333325) [optional arguments that the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=46.53433333333331) [function requires. In line 8, we schedule](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=48.049) [the new thread to start by calling the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=50.821000000000005) [**start** method, and then in line 9, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=53.2547142857143) [suspend our execution until the new thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=54.9541428571429) [completes, by calling the **join** method on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=57.44990909090908) [the thread object that we want to wait for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=60.209) its completion.

[The full **threading.Thread**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=62.67500000000001) [**constructor** is showed on your screen. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=64.339) [***group*** parameters are reserved for future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=66.3285) [use, so it's always set to None. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=68.2215) [***target*** is the function to be invoked. It](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=70.76) [can specify a name for a thread using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=73.34158333333333) [***name*** parameter. If you choose not to, a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=75.78583333333339) [default name will be used, which will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=78.01980000000002) [the word Thread and a counter appended to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=80.27420000000005) [the thread. The ***args*** parameter is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=83.05966666666671) [argument tuple for the target function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=85.27954545454544) [invocation, while the ***kwargs*** parameter is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=87.085) [a dictionary of keyword arguments for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=89.7358) [function, if you choose to use that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=92.56854545454546) [instead. And lastly, there is a ***daemon***](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=95.05481818181819) [parameter, which is added in Python 3.3.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=97.63257142857145) [that specifies whether the thread will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=100.596) [terminated, if it's parent thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=102.79399999999995) [terminates or not. This way of executing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=104.39808333333333) [tasks with threads is the most common](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=106.83466666666668) [usage pattern for threading. It considers](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=109.34985714285716) [the thread to be a worker executing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=111.49366666666667) instructions in a target function.

[There's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=114.08033333333337) [a **second usage pattern** of threading where](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=116.01008333333334) [we use threads known as the worker](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=118.20166666666671) [function performing some tasks, but as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=120.85414285714286) [unit of work instead. To do this, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=124.12114285714289) [**create a class that inherits from the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=126.77542857142856)[**thread class in the threading package and**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=128.15342857142855)[**overrides the run method**. The \_init method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=130.76200000000003) [can also be overwritten to provide](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=134.04218181818183) [additional state in the object via the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=136.5665454545455) [constructor, but we should keep in mind](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=138.62299999999996) [that whenever you override a \_init method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=140.32399999999993) [in the sub-class of thread, the super](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=142.7106363636363) [class' \_init method must always be called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=145.1428181818182) [first before performing any other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=147.40318181818185) [operations. Here in our overridden run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=149.9087) [method, we are calculating the Fibonacci](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=153.62810000000002) [number for the number passed into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=155.34175) [constructor and printing out the value. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=156.873) [see an example of the usage here. We can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=160.4515) [create objects of our class, and because](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=164.3418333333333) [they are thread objects, we can call **start**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=165.4332499999999) [to schedule the threads to start, and we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=167.3105454545455) [can call **join** to wait for the threads to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=169.62433333333337) [complete. These threads we run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=172.42483333333348) [concurrently on the machine, and then the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=174.24199999999996) program would exit.

[Let's get our hands](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=176.8774999999999) [dirty and try out some of this code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=178.68946153846156) [ourselves using the demo (**script1\_threading**). The first thing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=180.05438461538466) [we'll do is import the threading package.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=186.2675) [Then we'll go ahead and create a new](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=188.546) [method. I'll call it download\_image. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=190.20733333333322) [method will be the target function for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=192.39954545454552) [threads. So I'll cut and paste the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=195.42838461538463) [critical code from the download\_images](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=197.6942307692308) [method to the new download\_image method. I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=200.26014285714285) [can then come in and create a new thread,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=208.168) [setting download\_image as the target and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=210.69700000000006) [passing in the URL as the argument, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=214.04864285714282) [start the tread. I could also go ahead and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=220.73835714285704) [call thread.join here, but that'll be a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=224.51471428571426) [mistake, because what would happen is that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=226.205) [in a loop, the main thread will create a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=228.691076923077) [new thread, start it, and then wait for it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=231.46849999999998) [to complete before continuing the loop.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=233.4079999999999) [This isn't what we want. What we want is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=235.5163333333333) [to create the worker threads, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=238.73353333333336) [wait for all of them to complete. So to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=240.69446666666673) [accomplish that, I'll create a list that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=243.06384615384616) [will hold references to the thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=244.81223076923078) [objects, and then every time I create a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=247.15200000000004) [thread in the loop, I'll add it to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=250.496) [list. Then after all the threads are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=255.51889999999997) [started, I'm going to loop through my](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=257.41869230769214) [thread list and call join on each of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=258.984) [threads, so the main thread is forced to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=261.5599285714284) [wait for all the threads to complete](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=264.70599999999996) [before continuing. I'm ready to run this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=268.70766666666657) [code, but before I do, I'd like to add a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=271.275375) [couple of log messages to my](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=274.124125) [download\_images method in order to get a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=275.34169230769237" \t "psplayer) [better picture of what actually happens.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=276.6474615384617) [And then I'm going to specify a format for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=281.1644166666667) [the log messages that will allow me to see](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=288.2223846153846) [the threadName, the current time, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=292.7016153846155) [logging level, and of course the message.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=297.553) [The last thing I want to do is scroll down](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=304.301) [to the make\_thumbnails method, and comment](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=306.3617692307692) [out the perform\_resizing method, just](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=308.4254000000001) [because I don't really need this part of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=310.8344000000002) [the code to run for this illustration.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=313.6566153846154)

[Alright, so I'll open my terminal, and run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=315.825) [pytest to run my code. The log file will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=320.50992307692303" \t "psplayer) [have my results as they come in. So we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=324.8394545454545) [see that the threads get kicked off within](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=328.2228) [milliseconds of each other. On the left,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=330.70493333333343) [there is the threadNames and the times.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=333.0671) [And because they run concurrently, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=335.683) [entire download completes in 6.5 seconds,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=338.41419999999994) [instead of the 27 seconds that it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=341.0105000000001) [previously took, which is almost a 4 times](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=342.8200000000003) [improvement in performance. Let's talk a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=345.66966666666656) [bit more about how threads work, as this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=347.4096428571429) [would give us the foundations needed to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=348.93935714285726) [understand thread synchronization and inter-thread communication.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=1&mode=live&start=353.513)

[How Threads Work](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live)

[Using our simple example, we can trace the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=2.205) [lifecycle of threads that are created and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=4.668999999999999) [executed when this code is run. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=6.630999999999999) [program starts, there's only one thread in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=8.956166666666666) [existence, the mainThread. The mainThread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=11.984250000000001) [executes the instructions for importing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=14.14777777777778) [the threading library, defining the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=16.31222222222223) [do\_some\_work method, and creating the val](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=18.409499999999998" \t "psplayer) [variable. The mainThread then creates the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=20.87249999999999) [new thread. At this point in time, the new](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=22.39371428571428) [thread is in a new state. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=25.078000000000003) [mainThread calls start on the newThread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=27.441333333333336" \t "psplayer) [object, it goes into the ready state,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=28.77133333333334) [which means that the thread is now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=30.39230769230769) [available for the OS to schedule to run on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=32.105692307692294) [a CPU. After this, the mainThread calls](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=36.10014285714284) [join. Prior to this, the mainThread had](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=39.90364285714282) [been in the running state, but now it goes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=42.37299999999999) [into the blocked state, which means that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=44.77646153846154) [it is suspended and can't execute until](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=46.17376923076923) [something happens. In this case, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=49.37942857142858) [something that needs to happen is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=51.68999999999999) [completion of thread 1. Thread 1 goes into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=53.76899999999997) [the running stage, executes the 2 print](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=56.3879090909091) [instructions, and terminates once the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=58.892) [method is completed. This signals to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=60.95137500000001) [blocked main thread that it can now go](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=63.102999999999994) [back into the ready state and finish](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=64.93499999999999) execution.

[Here's a diagram depicting the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=67.65527272727272) [full **threads lifecycle**. A thread starts](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=70.37) [out in a new state, and from there it goes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=72.7110909090909) [into the ready state when start is called.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=76.14315384615388) [From there it moves back and forth between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=78.513) [the running state and the ready state.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=80.83940000000004) [Whenever it's actively executing on a CPU,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=82.875) [it's in the running state, and if its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=84.94875) [execution gets paused, it goes back into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=87.61223076923078) [the ready state. If a situation arises](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=90.37076923076924) [where a thread has to wait for a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=92.18961538461537) [particular condition to occur before it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=93.4628461538461) [can continue executing, it goes into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=95.6595) [blocked state. For example, it may block](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=98.51199999999999) [while waiting for an IO operation to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=100.70074999999999) [complete or while waiting for another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=102.624) [thread to complete. A thread cannot move](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=105.87533333333334) [from the blocked state back into the ready](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=108.34099999999998) [state until the condition that it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=110.02099999999993) [waiting on occurs. The final state is of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=112.44225) [course the terminated state, which occurs](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=115.00858333333335) [when the thread completes or if an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=116.49608333333336) unhandled exception occurs.

[In a program](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=119.94066666666669) [with multiple threads, each thread has its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=122.77024999999999) [own counter, which maintains the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=125.64316666666663) [instructions that has been executed at the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=127.0331818181818) [current time, its own registers to hold](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=128.3612727272727) [the data being used in computation, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=130.96200000000005) [its own stack, which is memory set aside](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=133.7172) [as scratch space for a thread. Local](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=136.4708) [variables within a function are kept on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=138.77241666666666) [the stack and are therefore thread safe as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=140.42733333333325) [each thread has its own stack space.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=143.19518181818177) [However, threads within a process share](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=146.384) [code, a common memory space, and other OS](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=148.78072727272726) [resources such as open files or network](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=152.07236363636366) [sockets. Therefore any data or area of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=155.09254545454556) [memory owned by a process can be read and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=157.73971428571434) [modified by any of the threads that are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=160.92761538461536) [part of the process. To make things more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=162.79653846153838) [interesting, once a thread is started, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=165.32299999999998) [have little control over how it runs. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=168.07933333333335) [operating system using an algorithm](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=171.02600000000007) [determines what threads run, for how long](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=172.79100000000003) [they get to run before they get suspended,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=175.1010000000001) [and on what processor core they run on. A](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=178.00466666666665) [thread may be running, and then after a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=181.42023076923076) [certain amount of processor time as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=183.40607692307682) [determined by the **OS scheduler**, the thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=185.03536363636363) [gets suspended mid-execution, so that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=187.26263636363632) [other threads that are waiting for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=188.8317) [processor time get to run. The scheduler](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=190.69710000000003) [then uses an algorithm to select which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=193.23261538461546) [threads should run next and performs a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=195.828) [context switch to cycle the running thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=198.98630769230772) [out of the CPU and the selected thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=201.6236923076924) [into the CPU. If the newly-selected thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=205.22775000000004) [is from a different process, a full](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=207.63644444444446) [process switch occurs, which is a fairly](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=209.6798181818182) [expensive process. However, if it's a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=212.78718181818192) [thread from the safe process, the thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=214.6583846153846) [switch is executed, which is a less](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=216.3793076923076) [expensive process. This is one of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=219.0045714285715) [reasons why in most cases parallel](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=221.4013333333334) [programming using threads is preferred](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=223.2533333333335) [over parallel processing using processors.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=226.00111111111107)

[But thread-based parallelism is not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=229.05) [without its perils. Memory sharing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=230.99312499999996) [combined with indeterminate scheduling can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=232.9008) [lead to a situation known as **thread**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=234.75530000000003) [**interference**. The canonical example used](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=237.37359999999995) [to describe thread interference is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=239.12199999999996) [BankAccount example. If two threads are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=241.18" \t "psplayer) [allowed to run this code, and one of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=243.8183636363637) [threads is running the deposit method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=247.42776923076923) [while the other is running the withdrawal](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=249.54392307692308) [method, we could run into a situation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=251.9673333333333) [where thread 1 reads the balance as 0, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=254.35666666666657) [gets interrupted and suspended, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=258.4719999999999) [thread 2 comes in and reads the balance as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=260.5262307692308) [0 as well, it subtracts 50 from 0,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=262.03130769230796) [resulting in -50, thread 2 stores the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=265.22300000000007) [resulting balance as -50 and then exits.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=268.989) [Thread 1 gets switched onto the processor,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=271.3579166666666) [and its stack gets restored. It adds 100](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=274.002) [to 0, resulting in 100, and stores the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=277.74815384615385) [resulting balance as 100 and exits. Here](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=281.3912307692308) [the 50 dollar withdrawal is completely](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=284.59700000000004) [lost as a result of thread interference.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=287.35100000000006) [And the balance that should have been 50](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=289.99718181818173) [ends up incorrectly being 100. **Thread**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=293.00390909090913)[**interference is also commonly referred to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=295.80263636363645)[**as a race condition**, as in both threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=298.4135714285714) [are in a race to read or update the same](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=301.6444285714285) [variable, and you may get different](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=304.7530000000001) [results depending on what threads run the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=306.379) [code at what time. And since this is out](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=309.0033636363638) [of programmer control, sometimes your](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=312.0984545454545) [program results in a correct outcome, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=313.693909090909) [other times it doesn't. This can be very](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=316.19519999999994) [hard to debug, as the data corruption may](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=318.4522857142857) [occur only intermittently in your](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=320.0568571428572) [production environment under heavy load,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=322.452625) [but the code runs fine in your local](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=325.777) [development environment. To solve this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=328.85940000000016) [problem, we use a technique called thread synchronization.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=2&mode=live&start=332.63)

[Thread Synchronization](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live)

[Before we talk about thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=1.912) [synchronization, I must mention that it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=3.004727272727272) [**best practice** in concurrent programming to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=4.316) [**keep accessing shared memory for multiple**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=6.625333333333333)[**threads to a minimum**. The more shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=9.137285714285714) [memory access that is done by threads, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=11.935799999999997) [more complicated your code gets, and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=14.51675) [less concurrent it runs, because of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=16.433) [need to combat thread interference. But](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=18.730624999999996) [sometimes the function of your program](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=21.169) [unavoidably requires threads to read and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=23.773000000000007) write to shared memory.

[Python has a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=26.302) [number of mechanisms that can be used to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=28.50075) [synchronize access to shared resources.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=29.934749999999998) [The most fundamental of these is the **Lock**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=32.22566666666667) [A lock is in one of two states, locked and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=36.449) [unlocked. When put in the locked state by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=40.68263636363639) [one thread, it could only be unlocked by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=43.262999999999984) [the same thread. It cannot be locked or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=45.771846153846155) [unlocked by another thread. For this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=48.586615384615385) [reason, we use the word acquire to signify](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=50.515846153846155) [a thread's putting of a lock into the lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=52.60323076923078) [state, because once the lock is acquired](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=55.2044) [by a thread, it can't be acquired by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=57.32586666666666) [another thread until it has been released](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=60.14650000000002) [by the thread that owns it. So what](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=62.65938461538461) [happens if a thread tires to acquire a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=65.04646153846154) [lock that's being held by another thread?](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=67.41636363636364) [Well, that thread goes into the block](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=69.475) [state, which means that its execution is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=71.67475) [paused and can't continue until the lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=73.94133333333333) [that it is trying to acquire is released](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=76.37500000000001) [by the owning thread. We can use this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=79.8752222222222) [locking and unlocking mechanism to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=83.39554545454546) [synchronize access to shared resources.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=85.33872727272728)

[To do so, we'll need to create a lock for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=88.24176923076924) [shared resource. Before accessing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=91.8794615384616) [shared resource, the thread must acquire](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=93.49881818181818) [the lock for that particular resource.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=95.28245454545454) [This ensures that no other thread can also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=97.45845454545456) [acquire the lock and access the resource](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=100.52291666666666) [at the same time. When the thread is done](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=102.60133333333329) [with the resource, it releases the lock,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=105.18275000000001) [so that other threads are now allowed to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=107.134) [access the resource. But there's a small](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=109.99218181818183) [flaw in that example code. If something](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=112.28311111111108) [goes wrong during the resource access, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=114.29009090909092) [an exception is thrown, we may never](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=115.98090909090914) [release the lock, which would cause all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=117.9693846153846) [the other threads waiting on the lock to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=120.07692307692304) [block forever. We can solve that simply by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=122.52544444444447) [using a **try finally block**. By putting the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=125.66609090909093) [releasing of the lock in a finally block,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=128.50045454545457) [we ensure that it will be run even if an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=130.443) [exception is thrown. But we can take it a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=134.2437692307693) [step further. In Python 2.5, the **with**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=136.83079999999993) [**statement** was introduced to make the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=138.7598666666665) [acquisition and releasing of certain types](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=140.63009090909097) [of objects, such as locks, files, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=142.585) [network connections easier. This example](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=146.14100000000008) [shows how to use the with statement to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=148.21599999999998) [manage the acquisition and releasing of a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=150.41999999999993) [lock. The lock is automatically acquired](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=153.52149999999997) [when entering the with block and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=156.49136363636364) [automatically released when leaving. As](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=159.0075454545455) [with the try finally block using the with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=160.88715384615384) [statement, the lock will always be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=162.5043846153845) [released, even if an exception occurs](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=164.47209090909092) during resource access.

[Let's try this out](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=167.94936363636373) [using our thumbnail\_maker example (**script1\_threading\_lock**). Let's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=172.36149999999998) [say we wanted to get the total number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=175.214) [bytes downloaded for accounting purposes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=176.82500000000002) [or to see how much space we would save](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=177.91738461538463) [after we resize the image. The initial](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=180.4166153846155) [value for our downloaded\_bytes variable](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=184.85550000000003) [will be 0. In the download\_image method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=188.739) [I'm going to do a bit of cleanup by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=191.84450000000004) [creating this dest\_path variable, so I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=195.7771666666667) [don't have to keep computing it over and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=198.25866666666678) [over again. I'll get the size of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=202.14499999999998) [download image into a variable called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=206.60899999999992) [img\_size. Next I'm going to update the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=211.22366666666662" \t "psplayer) [downloaded\_bytes counter by adding the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=216.903" \t "psplayer) [img\_size for this particular image to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=218.7168888888889" \t "psplayer) [whatever the current value is. While this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=221.2930769230769) [may seem like one operation, it's really](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=225.23084615384607) [three. First the Python interrupter reads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=228.86130000000003) [the current value of downloaded\_bytes,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=231.6219000000001) [then it adds that value to the value of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=233.94818181818187) [img\_size, and lastly it stores the summed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=238.16138461538463" \t "psplayer) [value back into downloaded\_bytes. If one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=241.47723076923086) [of the threads running this code gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=244.02183333333335) [interrupted after it's read the value of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=246.29625000000001) [downloaded\_bytes, summed data could be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=248.7306" \t "psplayer) [lost, as some other thread could've also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=250.55510000000004) [updated the same value without its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=253.2447692307692) [knowledge. The solution here is to use a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=255.56953846153843) [lock. We'll call our lock dl\_lock, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=259.17) [then we'll move our critical section of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=262.7081875) [code into a with block for that lock. Now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=268.1975000000002) [we have the guarantee that even if the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=274.68575) [current thread gets interrupted, no other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=276.40374999999983) [thread can modify the downloaded\_bytes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=278.8283333333334) [variable while it has the lock. The last](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=281.79555555555567) [thing we want to do here is to print out](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=283.5300000000001) the downloaded\_bytes value.

[Now I'm going](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=285.88775) [to run this and expect to see the size and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=291.23225000000014) [bytes for each image, and the total number](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=298.7909999999999) [of bytes downloaded. One quick thing to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=303.75899999999973) [**note is when to use locks and when they**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=307.06413333333325)[**aren't needed**. If we're reading the value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=310.308933333333) [of a shared variable that isn't modified,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=312.44708333333324) [like we're doing in line 27, then we don't](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=314.431) [need to synchronize access, since we're](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=317.6433076923076) [just reading the value and never modifying](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=319.79859999999996) [it. Also, if instead of updating the value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=322.34519999999975) [of downloaded\_bytes, we simply replaced](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=325.2468888888889) [it, then we won't need a lock, since these](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=327.1004285714285) [are both atomic operations, which means](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=330.48571428571404) [they happen in a single step, therefore](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=332.7884000000001) [they can't be interfered with. Other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=335.5635) [examples of atomic operations are getting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=338.2605) [an item from a list or dictionary and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=340.75130769230776) [adding in an item to a list or dictionary.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=343.79746153846185)

[Now before we move on to other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=346.715) [synchronization methods, there are a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=348.871) [couple of things to note about the lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=350.2291818181819) [API. The **lock.acquire** method takes in an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=354.2223333333333) [optional ***Boolean argument*** that defines](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=358.0487272727271) [whether the thread trying to acquire the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=359.78358333333335) [lock should block or not, if the lock it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=361.3486666666668) [trying to acquire has already been](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=363.8736666666668) [acquired by another thread. By default,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=365.83046666666695) [this argument is set to true, but if you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=367.7507333333333) [set it to false, the call to the lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=370.5215333333332) [doesn't block the calling thread. Instead](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=372.9935000000001) [it returns false if the lock cannot be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=374.6780769230769) [acquired and true if it's successfully](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=378.23069230769227) [acquired the lock. There is also the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=380.8044545454546) [**locked** method. The locked method allows](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=383.5955454545456) [the caller to determine whether the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=385.5381999999999) [particular lock has been acquired or not.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=387.256) [This may come in handy if there is some](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=389.498) [code that you want to run in case where a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=391.2946250000002) lock has already been acquired.

[In the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=394.556875) [category of locks, there is a type of lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=398.2805714285715) [called reentry lock or **RLock** for short.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=401.05064285714303) [With a regular lock, the lock can only be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=403.5816923076923) [acquired once, even by the same thread. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=407.199) [a thread tried to acquire a lock that it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=410.347923076923) [already owns, it will block. The RLock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=413.864) [fixes this by allowing a thread to acquire](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=416.7560666666667) [a lock is already holds. If a thread tried to do so, it would just continue.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=3&mode=live&start=420.5843333333336)

[Inter-thread Communication Using Queues](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live)

[Another widely-used thread synchronization](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=1.248) [mechanism is the **Semaphore**. A semaphore is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=3.5765000000000002) [a synchronization primitive that manages](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=7.0531) [an internal counter instead of a single](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=8.9666) [locked or unlocked switch. Each time a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=11.643111111111107) [thread call is acquired, the internal](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=14.276000000000002) [counter is decremented. And each time](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=16.484) [release is called, the counter is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=18.877100000000002) [incremented. The internal counter can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=21.455300000000005) [never go below 0, so if a thread makes a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=22.849533333333337) [call to acquire, and the value of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=25.260866666666676) [internal counter is 0, the thread gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=27.887727272727275) [blocked and must wait for another thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=30.218249999999998) [to call release to increment the counter](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=32.47516666666666) [before it can continue. A common analogy](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=35.717222222222226) [is that the **semaphore maintains a set of**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=38.97025000000001) [**permits**. Every call to acquire attempts to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=42.10425000000002) [acquire one of the permits. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=44.94836363636362) [thread is done, it must then release the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=47.76345454545454) [permit. If there are no permits currently](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=50.84927272727273) [available, then the thread must wait for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=52.65700000000001) [another thread to release a permit back](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=54.42453846153846) [into the pool before it can acquire the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=56.84330769230768) [permit, whereas a lock can be used to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=59.61264285714286) [permit only one thread to run a critical](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=62.390357142857155) [section of code at the same time.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=64.63908333333333)

[A semaphore can allow one or more threads to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=66.54192857142857) [run at the same time, depending on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=68.24535714285719) [number used when initializing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=70.70266666666666) [semaphore. If no parameter is passed to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=73.02377777777775) [the constructor of the semaphore like the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=74.77199999999999) [example on the screen, then the internal](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=76.27) [counters defaults to 1. In the next](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=79.40090909090912) [example, we're passing in the number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=82.01407692307691) [permits that can be acquired without](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=83.93692307692305) [blocking. We are also using a **Bounding**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=85.95281818181816) [**Semaphore** instead of the standard](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=88.73372727272721) [semaphore implementation. This is because](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=91.65366666666668) [the standard semaphore implementation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=93.76820000000002) [allows you to call release an unlimited](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=94.90580000000004) [number of times, more times than you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=97.44853846153846) [called acquire. With the bounded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=100.40523076923077) [**semaphore, if you inadvertently attempt to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=102.43590000000002)[**release more times** than you have calls to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=104.80770000000005) [acquire, an **error** will be thrown.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=107.20553846153844)

[Let's use our thumbnail\_maker to illustrate how](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=109.26327272727272) [we can use semaphores for thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=111.31090909090906) [synchronization (**script1\_threading\_lock\_semaphore**). Let's say we have a limit](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=115.402) [of the number of concurrent image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=118.07115384615382) [downloads we are allowed to perform at](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=119.37730769230764) [once. Now I don't believe DropBox has such](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=121.47669230769233) [a limit, but image we had to cap our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=123.99733333333333) [downloads at a maximum of 4 concurrent](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=126.92533333333333) [downloads. One way to do this would be to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=129.20266666666666) [use a semaphore. I'll start by setting the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=132.74545454545452) [value of max\_concurrent\_dl to 4, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=136.34250000000006) [instantiate a semaphore object with that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=139.5925) [value. Then in my download method, I'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=144.74949999999998) [call acquire at the start of the method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=147.70623076923076) [body, and release at the end of it. To](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=151.31066666666666) [ensure that release is always called, I'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=161.2803846153846) [use a try finally block. Now when I run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=166.23907692307682) [this code, what I should see is that no](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=171.17725000000002) [more than 4 threads can execute the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=174.31881250000004) [download code at the same time. Looking at](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=176.96280000000004) [the logs, I can verify this. Initially](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=180.07369230769228) [only 4 threads can download at the same](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=182.57861538461526) [time, because we have only 4 permits. Once](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=185.24709090909096) [thread 4 is done and releases its permit,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=187.70723076923076) [then thread 5 can acquire the permit and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=190.0690769230768) [begin running. One thing to note is that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=192.85833333333335) [instead of following acquire and release](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=196.075) [in the finally block, we can also use the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=197.82340000000005) [with statement just like we did with the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=200.2529285714286) [lock. In the context of thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=202.27521428571438) [synchronization, the with statement can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=204.34209090909084) [used with locks, semaphores, and condition](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=206.519) [objects. We'll talk about conditions](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=209.75128571428576) [shortly, but first let's briefly discuss](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=212.83981818181817) events.

[**Events** are pretty simple. One](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=216.66454545454545) [thread signals an event, and the other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=219.17533333333327) [threads wait for it. An event object has](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=221.798) [an **internal flag**. The thread or threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=225.02272727272725) [that need to wait for the event to occur](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=227.3677333333333) [do so by calling the **wait** method on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=229.9231333333332) [event. As long as the event's internal](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=231.77630769230777) [flag is false, the threads that call wait](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=233.37090909090907) [will block. A **manager or server thread** can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=237.3141818181817) [then call the **set** method on the event,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=239.7824285714285) [which sets the internal flag to true and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=241.759) [releases the block threads or threads. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=245.25500000000008) [server or manager method can also call the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=248.1229230769231) [**clear** method on the event, which resets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=250.0903076923077) [the internal flag to false and causes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=252.24563636363635) [subsequent threads calling wait to block](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=254.97945454545447) [until set is called to set the internal](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=256.8188461538462) flag to true again.

[**A Condition combines**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=259.137)[**the properties of a lock at an event**. Like](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=262.71959999999996) [a lock, it has an **acquire** and a **release**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=267.0090769230769) [method to synchronize access to some](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=268.764769230769) [shared state, and like an event, it has a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=270.5426153846153) [**wait**, a **notify**, and a **notify\_all** method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=274.1204615384612) [so that threads interesting in knowing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=276.21927272727265) [when the state changes can call the wait](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=277.973) [method and can be notified when the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=280.6610000000001) [condition changes. The typical use case](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=283.70720000000006) [for a condition is in implementing a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=286.57276923076927) [**producer/consumer pattern**. I have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=289.6016153846155) [simplified example of this on the screen.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=291.6292) [In this example, the consumer thread gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=294.341) [the lock, and checks whether an item is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=296.2677500000002) [available, and calls wait if one isn't.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=299.0641999999999) [Calling wait releases the lock, allowing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=302.378) [the producer thread to acquire it, produce](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=304.0478) [the item, and then notify the waiting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=306.37774999999993) [consumer thread. It must also release the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=309.2168333333331) [lock in order for the waiting consumer](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=311.72915384615374) [thread to be able to actually return from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=314.11076923076894) [its wait call. The consumer thread wakes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=317.31330000000014) [up when notify is called, reacquires the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=319.5946666666667) [lock, and then gets and processes the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=321.8228181818182) [produced item. It releases the lock when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=325.0071818181819) it's done.

[This example is as simple as it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=327.64378571428574) [gets with conditions, yet I still have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=330.752) [problem with it, because it's not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=332.4450909090911) [intuitive to the reader of the code, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=333.97571428571433) [as a writer, I have to constantly keep the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=336.3642857142859) [state of the threads and condition in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=338.91507692307687) [mind, which can be complex and error](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=340.846) [prone. What if I don't have to deal with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=343.34762499999994) [that? What if there was an easier way of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=345.9791333333333) [communicating between threads that didn't](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=348.5694444444444) [involve locks, conditions, and thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=350.0705555555554) [states? Well, there is, and it's called a queue.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=4&mode=live&start=353.671)

[Demo: Multiple Reader Threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live)

[The Python standard library **queue** is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=0.958) [construct that makes it easier to exchange](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=2.934916666666667) [messages between multiple threads. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=5.2906666666666675) [queue handles all the locking semantics](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=7.529181818181818) [for us, the programmer, and allows us to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=9.636272727272726) [focus on just four methods, **put**, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=12.78766666666667) [puts an item into the queue, **get**, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=15.740384615384617) [removes an item from the queue and returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=18.817923076923083) [it, **task\_done**, which marks that an item](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=21.410846153846162) [has been gotten from the queue and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=23.425230769230787) [processed completely, and **join**, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=26.550999999999995) [blocks until all the items in the queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=28.789250000000003) [have been processed. There are more than](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=31.559250000000013) [four methods in the queue API, but these](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=33.74942857142857) [four are the most common and important](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=36.0522857142857) [ones. Using a queue to communicate between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=38.734818181818184) [threads brings us closer to a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=41.78122222222222) [**message-passing architecture**, resulting in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=44.890555555555544) [code that's cleaner, safer, and more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=46.47516666666667) [readable most of the time, as opposed to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=48.11966666666669) [the messier world of synchronized access](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=50.493538461538456) [to shared state. Therefore **if the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=52.40984615384614)[**situation fits, I'd recommend using queues**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=54.95680000000001)[**over conditions or events**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=58.13440000000002)

[Here's our producer/consumer example again, but using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=62.3332) [a queue instead. We have a queue object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=64.1462) [and our two threads, t1 and t2 that act in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=66.88993749999999) [a capacity of the producer and consumer.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=70.20181249999997) [In the body of the producer method, we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=73.34600000000002) [see that the thread produces 10 items and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=76.92678571428571) [puts each one in the queue. Then our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=78.97421428571427) [consumer thread simply loops and gets the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=81.21866666666666) [items from the queue one at a time, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=83.286) [calls task\_done each time it completes the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=86.43150000000006) [processing of an item. It's important to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=88.85300000000001) [know that both the **get call and the put**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=91.21200000000003) [**call are blocking calls**. When the queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=94.99139999999998) [object is instantiated, you can specify](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=97.46072727272725) [the maximum number of items that can be in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=98.70818181818176) [the queue. If you don't specify a value,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=101.26911111111116) [like we have in our example, then you have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=103.52249999999998) [an unbounded queue; however, if we had](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=106.23284615384614) [specified a value for ***max\_items***, then the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=108.7291538461538) [put can block if the queue is full. It](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=111.67666666666665) [will block until a consumer thread removes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=115.77084615384615) [an item from the queue, or until a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=118.14276923076919) [user-specified **timeout** threshold is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=120.80412499999998) [reached. The get call can also block, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=122.09362499999996) [it actually blocks more frequently in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=125.62799999999999) [practice. The get call blocks when there](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=128.38114285714283) [are no items in the queue to get. In this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=130.67299999999997) [situation, the thread making the get call](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=133.57546153846155) [is blocked until the new item is put in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=135.18007692307702) [the queue for it to read, or until a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=137.88899999999995) [user-specified time out is reached. We can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=141.11459999999985) [pass a **flag into the get call**, instructing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=142.81330769230772) [it not to block if the queue is empty, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=144.63807142857144) [we'll see an example of that later.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=147.2487857142857)

[Now that you've seen how easy it is to pass](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=149.9502142857143) [messages amongst threads, let's put it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=153.403) [into practice in our thumbnail\_maker](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=155.45079999999993) [example(**script2\_threading\_queue**). In the last couple of sections,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=158.513) [we've spent time building up locks and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=162.0298461538462) [semaphores, but I'd like to temporarily](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=164.153) [back away from that right now, and go back](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=165.92700000000005) [to the state of the code when we first](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=168.07021428571426) [cloned or downloaded the code, but I don't](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=169.17785714285705) [want to lose the code we already have, so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=171.09338461538456) [I'm going to create and check out a new](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=173.72116666666665) [branch. I'll call it threading-intro. And](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=176.78191666666658) [then commit the code that we have so far](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=180.27866666666674) [with the commit message of threads, locks,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=185.87345454545456) [and semaphores. And then I'll switch back](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=189.9875454545455) [to the master branch, which should be in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=192.8172857142858) [the original state it was when I first](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=195.46414285714303) [cloned the repo, but all the changes I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=197.7049230769231) [made earlier are not lost, they're just in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=200.0172307692308) [a different branch. If you're following](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=202.7181538461539) [along and aren't using Git, then the way](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=204.39014285714282) [to do it will be to rename your Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=206.14785714285702) [file with the stuff you've done so far,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=208.70942857142862) [that is maybe](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=211.246) [thumbnail\_maker\_threading-into, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=212.39537500000003" \t "psplayer) [grab an original copy of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=213.54475000000005) [thumbnail\_maker.py, and place it in your](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=216.49069230769234) project folder.

[So the first thing we want](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=220.30515384615393) [to do is from queue import Queue, and from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=224.52607692307694) [threading import Thread. I'm also going to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=229.54453846153845) [set up the format string for the logger,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=234.32392307692302) [just like we had before. Specify the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=239.03722222222217) [threadName, the time, the log level, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=244.13263636363638" \t "psplayer) [the message, and pass it to the config](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=253.29436363636376) [method. Then I'm going to instantiate a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=257.7563076923076) [queue instance in the constructor. I'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=259.908) [call it img\_queue. In my download method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=266.4506666666665) [all I need to do is after a download is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=271.0368571428571) [complete, I'll pop the file name for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=274.324) [file I just downloaded into the queue. For](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=280.0530666666668) [the resize method, now instead of reading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=285.5845714285714) [the folder, I want to go into the loop,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=289.21457142857133) [and in the loop I want to read the files](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=295.05000000000007) [from the queue. Then I mustn't forget to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=304.1300000000002) [mark a task as done after I'm done](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=310.9682857142858) [resizing. Oh yeah, I don't need that piece](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=317.5511428571432) [of code anymore since I'm reading off the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=320.16100000000006) [queue and not the filesystem, and that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=322.36158333333333) [will actually check the filesystem before](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=325.2566666666668) [it runs. Now let's go to our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=327.9478) [make\_thumbnails method, and create our two](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=331.57659999999987" \t "psplayer) [threads. T1, which will target the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=336.1499999999999) [download\_images method, and t2, which will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=341.016" \t "psplayer) [target the perform\_resizing method. As](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=349.82639999999986) [usual, we need to start and join both](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=356.4924) [threads. That's the majority of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=362.83159999999987) [changes that need to be made. For](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=365.21036363636364) [informational purposes, I'm going to add a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=367.4803333333333) [login statement to the beginning and end](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=370.0446666666666) of the resize operation.

[But I see a small](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=374.58671428571427) [problem here though. Up here we have an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=380.12757142857146) [infinite loop, and so our thread will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=382.42928571428575) [never end, which means that our program](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=384.6877692307694) [will also never end. So how do we fix](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=386.987) [this? We could change the while condition](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=390.48620000000005) [to test the flag that gets set by the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=392.584142857143) [producer thread, but then we go back into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=395.1553076923077) [the world of shared state and race](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=397.4937692307693) [conditions, and the chances are very high](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=399.947) [that the resizing thread misses some items](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=402.943) [that are still in the queue. Since what we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=405.55945454545457) [need is a way for the download thread to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=408.4884) [tell the resize thread that there are no](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=410.9376) [more items, we could have the download](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=412.66784615384614) [thread do so by putting a special message](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=413.9472307692307) [into the queue, so when the resize thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=416.55549999999994) [receives the message, it can exit the loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=419.5222727272727) [and terminate. This technique is known as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=423.8524545454544) [the **poison pill technique**. Here's how that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=425.8155555555554) [would look. After downloading all the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=428.0522500000001) [files in the download\_images method, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=431.11675000000025) [could put a non-value or empty string into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=434.2969230769231) [the queue. This message marks that there](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=438.6913846153848) [are no more items to be enqueued.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=442.36125000000004)

[Now in the resize method, we need to check the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=447.1772352941176) [items that we get from the queue. If it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=452.04217647058806) [not a none or an empty string, we process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=456.985) [it; otherwise, we break out of the loop.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=460.0720000000002) [Of course we must not forget to mark the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=462.67360000000014) [task as done. I think we're ready to run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=465.547) [this, we'll run it using pytest as usual,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=469.3964285714285) [and we can switch over to the log file and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=473.608375) [view the progress. So what we have now is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=480.519625) [we have one thread downloading and putting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=484.14146153846167) [items into the queue, this is our producer](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=485.948) [thread, and we have another thread reading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=487.97569230769216) [items from the queue and making](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=490.34628571428567) [thumbnails, this is our consumer thread.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=493.6561428571427) [We only expect a minimal amount of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=496.966) [performance gains, because the downloads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=499.17488888888874) [are still happening sequentially. However,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=501.1358571428571) [we're not waiting for all the downloads to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=503.15341666666666) [complete before we start resizing, so that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=505.10875) [results in some performance improvements](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=507.61472727272724) as we can see here.

[But we can take this a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=509.8306363636363) [step further, and get much more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=513.4710000000001) [significant performance gains by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=515.307) [parallelizing our downloads. Because we're](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=519.1887142857142) [now using a message-passing style, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=522.655090909091) [queues handle all the locking for us](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=524.3203636363638) [behind the scenes, it's just as easy to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=526.4486923076922) [have multiple threads that are reading and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=529.0400769230766) [writing from the queue, as it is to have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=531.4450769230771) [one thread on each side. So we can come up](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=534.5807333333335) [with a design where we have our main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=538.1194000000005) [thread throw all the URLs from the list](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=540.1703333333332) [into a queue, and then have multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=541.7532666666664) [download threads pulling items off of that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=544.4308181818184) [queue, performing the download, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=547.494) [pushing the message into another queue for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=549.4440000000003) the resize thread to process.

[We'll start](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=552.4105) [by adding another queue from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=557.2077777777778) [downloads. The main thread will load all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=563.2671111111111) [the URLs for the images to download into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=565.6574285714289) [this queue, and then spin up some threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=567.498) [to drain the queue and perform the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=570.5096923076921) [download. Let's create the target method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=573.8553333333334) [for the download threads. We'll need to go](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=578.2453333333335) [into a loop and read messages off the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=582.2559999999999) [queue, but instead of going into an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=584.4239999999995) [infinite loop, let's loop until the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=587.1179999999998) [download queue is empty, because we know](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=589.541) [that once the queue is empty, no other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=591.5925384615384) [URLs will be added. Then as usual, we'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=594.3231428571429) [get an item from the queue, but this time](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=598.2117142857145) [we don't want to block and wait if the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=601.0836666666664) [queue is empty, we just want to exit the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=602.7534666666661) [loop. If we pass the block=False flag on a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=608.300142857143) [get call, and the queue happens to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=613.6962) [empty, then the call throws a **Queue.Empty**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=617.3633999999998) [**exception**. So let's catch that exception.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=622.3106000000001) [We don't have to do anything with it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=626.981) [though, we'll just write a log statement.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=632.5852666666663) [My ID doesn't recognize the Queue.Empty](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=637.489) [object, so it's giving me a red squiggly](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=640.0149230769233) [line, but that can be ignored. So you may](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=642.8485) [be wondering, if at line 25 we only go into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=645.8215) [the loop if there are items in the queue,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=648.7239999999999) [then why do we have to guard against the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=650.9833333333331) [queue being empty in line 27? This is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=652.7281333333327) [because since we plan on having multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=655.354769230769) [threads reading off the queue, we could](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=657.842090909091) [have a situation where this is one last](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=660.6567272727275) [item in the queue, and the thread comes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=663.0358571428569) [along and checks if the queue is empty,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=664.9124285714279) [and gets back a result of false. Then it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=667.2110000000001) [goes to read the item of the queue, but by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=669.7048666666666) [that time another thread has come in and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=671.8541999999997) [taken that item off the queue. So that's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=674.2953636363635) [the situation we're guarding against with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=677.1234) [the block=False flag in the try catch.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=679.1405999999997) [Okay, so after this we do our download,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=681.6237692307694) [and put the item into the image queue for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=684.521) [the resize thread to pick up. To save on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=687.0433999999998) [typing, I'm just copying and pasting the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=691.0232727272729) [code. Lastly, we mustn't forget to mark](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=696.3859090909096) [the task as done for every URL we get off](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=698.9155714285712) [the queue. There are no changes to be done](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=702.2113076923076) [in the perform\_resizing method, so I'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=706.4606923076917) [just move down to the make\_thumbnails](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=709.2391818181816) [method. Here we'll start by loading all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=711.990454545454) [the URLs into the dl\_queue. Then in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=717.8255454545457) [loop, we'll create four threads for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=723.1247692307694) [pulling URLs off the queue and downloading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=724.7623076923081) [them. And of course start the threads. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=730.0144) [don't need to call join on the download](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=740.0037333333333) [threads, just on the resize thread. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=742.2576000000003) [is because, as you may remember, a join](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=744.2893076923077) [blocks the calling thread until the join](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=747.0197692307696) [thread is complete. And here we know that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=749.2125384615384) [the resize thread is more important to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=751.4193076923076) [join on, because once the resize is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=753.9318333333332) [complete, then we are done, not when any](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=756.677) [other downloads threads are complete.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=759.6646923076927)

[But we still have a small problem left to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=761.7800000000001) [solve, the resize thread only terminates](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=763.7640000000005) [when it receives the poison pill, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=765.559) [we're not doing that in the download\_image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=768.3694999999996) [method. This is intentional, because if we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=770.5518181818182) [try doing it in the download\_image method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=772.263) [then we would run into a situation where](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=774.777076923077) [one of the four threads completes, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=777.3623636363636) [sends the poison pill, killing the resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=778.8686363636363) [thread before it can process messages from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=781.1067272727271) [the other three. So we need a way to wait](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=783.5746666666666) [for all the download threads to complete](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=786.5613333333336) [before sending the poison pill. This is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=789.0055000000001) [where the join API on the queue comes in.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=791.6127272727272) [What we can do is after starting the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=793.974) [resize thread, we can block the main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=795.6368571428574) [thread until all the downloads are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=797.6299166666666) [complete, by calling dl\_queue.join. And](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=802.1054166666664) [then we put the poison pill into the image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=806.1121428571429) [queue, so that the resize thread can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=808.6064285714285) [terminate when it reaches the end of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=811.0845) queue.

[I'll go ahead and rerun our test.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=814.7365000000001) [This time we expect a much more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=819.5634999999996) [significant performance increase now that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=823.934) [we have four threads downloading, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=824.7112222222223) [that's what we get. Our code runs in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=826.1066666666667) [roughly a quarter of the time. So we see](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=829.2013333333335) [now we can use queues to communicate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=832.0714999999998) [between threads with minimal or no need](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=834.168) [for logs, events, conditions, etc. We were](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=837.836) [even able to implement a pool of four](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=840.8729333333334) [threads to read from the queue and do the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=842.5406666666671) [download in parallel. This approach to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=844.9939090909091) [parallelizing our downloads is better than](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=847.1102727272729) [our previous approach, because we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=849.3657272727271) [control the number of threads that do the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=851.649) [download, and we don't run the risk of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=852.8250000000004) [creating too many threads. In the previous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=854.698909090909) [approach, we were looping through the list](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=857.3563636363633) [and creating and kicking off a thread to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=859.1755384615386) [download to each item from the list. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=861.1234000000001) [is fine if we have a small list of items](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=863.2706000000004) [to download, say less than 100 or in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=866.2343333333334) [low hundreds. In our case, our list is 26,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=869.2793333333337) [so we end up spinning up 26 threads. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=872.4041333333334) [our list size happened to be a 1000, 100](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=875.5653076923077) [thousand, or more, it would be really bad](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=878.8350769230767) [for the system we're working on to spin up](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=881.374705882353) [that many threads. In most systems, you'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=883.9841764705886) [either run out of stack memory space](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=886.5596666666668) [sometime before you hit the 100 thousand](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=889.5445833333335) [threads, or you'll hit an operating system](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=891.9570000000001) [limit, and your program will fail. So](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=894.1624285714286) [**instead of creating a thread for each item**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=895.7310833333333)[**in the list, and you can't anticipate its**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=897.7477499999998)[**size beforehand, the better option is to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=900.4802500000002)[**have a fixed pool of threads for the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=903.247)[**download**. That concludes our discussion on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=906.6256666666665) [the threading module, but before we move](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=908.795272727273) [on from parallelism using threads, we need](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=911.2315833333333) [to discuss the most talked about lock in Python, the global interpreter lock.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=5&mode=live&start=913.6086666666663)

[The Global Interpreter Lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live)

[If you take a look at best practices](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=1.068) [surrounding concurrency in Python, you'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=2.7356363636363645) [notice that it says to use **Python**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=4.018166666666667) [**threading only in cases** where you're doing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=6.318250000000001) [**IO-bound operations**. The reason for this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=9.941333333333333) [is the global interpreter lock or GIL for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=12.457538461538464) [short. The global interpreter lock is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=14.852615384615392) [lock within the Python interpreter code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=17.082363636363635) [that prevents multiple native threads from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=19.182624999999998) [simultaneously executing Python code. And](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=22.132374999999993) [Python interpreters that aren't thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=24.221333333333337) [safe, which is the majority of Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=26.218) [interpreters, the GIL exists to protect](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=28.826083333333344) [internal data structures from thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=31.697181818181818) [interference and race conditions. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=34.21763636363638) [allows Python interpreters to be easier to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=36.486333333333334) [write and to integrate more easily with C](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=38.25266666666666) [libraries that typically aren't thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=41.02163636363636) safe either.

[It also allows interpreters](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=43.22118181818181) [to run much faster for single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=45.45339999999999) [programs, because with the single lock for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=47.482) [the interpreter, there is no need to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=49.62166666666667) [acquire and release locks on all of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=51.400499999999994) [internal data structures individually.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=53.26849999999999) [There's just one global lock to acquire,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=54.794333333333334) [and the interpreter is off to the races.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=57.87) [The downside of the GIL though is that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=61.798) [**only one Python thread can execute at a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=63.72969230769229) [**time**. If this is true, then how are we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=66.33259999999999) [seemingly able to perform operations in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=70.51939999999995) [parallel using Python threads? This occurs](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=73.3537142857143) [because after a certain interval, Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=75.8574) [threads are forced to give up the lock in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=77.61359999999998) [order to allow another thread to run. So](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=80.79750000000001) [instead of having true concurrency, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=83.7839) [is multiple threads running at the exact](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=86.11130000000001) [same time on different processor cores,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=88.45340000000002) [what we actually have it corporative](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=90.221) [multithreading, in which threads give way](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=93.07159999999999) [to one another to allow each other to take](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=95.472) [turns to run on the same core. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=97.722) [actually degrades the performance if](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=100.75727272727273) [you're using multiple threads to try and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=103.30136363636366) [speed up CPU-bound operations, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=105.81122222222221) [IO-bound operations benefit from multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=107.59857142857142) [threads, because during IO-bound](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=110.56642857142853) [operations, the thread releases the GIL](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=112.31110000000001) [and blocks until the IO operation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=114.61330000000004) [completes. As a result, it's not in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=117.46325000000002) [contention for the CPU, and other threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=120.916) [can run during that time. That's the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=123.44766666666665) [reason best practices state that the use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=125.8665) [of Python threads should be limited to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=128.00325) [IO-bound operations, because the global](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=130.60388888888886) [interpreter lock denies the interpreter](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=132.71) [the ability to operate concurrently on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=134.43888888888893) [multiple threads, and instead forces only](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=136.71339999999998) [one thread to operate at a time in order](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=139.3875999999999) to ensure thread safety.

[There is an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=142.90018181818178) [ongoing attempt to remove the GIL from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=145.29075) [CPython interpreter, which has been coined](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=147.6714166666667" \t "psplayer) [the Gilectomy, but until data is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=149.71399999999994) [successfully completed, the Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=151.808) [community has two workarounds for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=153.5524444444444) [parallelizing CPU-bound tasks with Python.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=155.733) [One of the workarounds is to use a Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=160.158) [implementation that doesn't have a GIL.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=162.98024999999993) [Currently, the **only two Python**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=165.07163636363632)[**interpreters without the GIL** are the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=166.98936363636352) [Python implementation on top of Java known](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=169.18284615384613) [as **Jython** and the Python implementation on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=171.6161538461538) [the. NET Framework known as **IronPython**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=174.5938) [Because neither of these interpreters has](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=177.496) [a GIL, executing multiple threads over](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=178.95549999999994) [multiple processor cores is possible, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=182.34199999999998) [the most popular workaround for the GIL is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=186.2125384615385) [the use of multiprocessing. Here we use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=188.17684615384627) [multiple processes instead of multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=190.16190000000003) [threads for parallel execution of code.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=191.8334000000001) [We'll dive into this in the next module.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=194.527)

[But first, let's look back at what we've](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=198.577) [covered in this module. We started by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=201.16699999999992) [introducing the concepts of threads and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=203.31100000000004) [processes, and we talked about to create](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=205.00900000000013) [and manage threads in Python. We then dove](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=207.47763636363632) [deeper into thread components, thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=210.14924999999997) [states, and how the operating system](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=211.56633333333323) [schedules threads. We discussed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=214.02942857142858) [synchronizing multithreaded access to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=216.68828571428577) [shared state using mechanisms such as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=218.53049999999996) [locks, semaphores, events, and conditions,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=220.88549999999987) [and then we looked at using queues to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=223.12381818181822) [simplify inter-thread communication. Our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=226.84422222222224) [final topic in this module was the global](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=228.80511111111116) [interpreter lock, which gives us the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=232.2958181818182) [motivation for our next module, Multiprocessing.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m2&clip=6&mode=live&start=234.5905454545455)

[Multiprocessing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live)

[Processes vs. Threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live)

[In the last module, we introduced the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=0) [concept of processes. We defined a process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=3.4069) [as a running instance of a computer](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=6.089333333333335) [program, and we mentioned that every](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=8.228416666666671) [running instance, meaning every process,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=10.021499999999998) [has a processor state, security](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=11.509) [attributes, a section of memory assigned](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=13.335499999999998) [by the operating system, and possibly some](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=15.703000000000001) [system resources. We also mentioned that a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=19.490000000000002) [process can be composed of one or more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=21.559500000000007) [threads. It is important to know that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=23.667000000000005) [while threads within a process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=25.78800000000001) [automatically share the memory assigned to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=27.315400000000004) [the process, **processes do not share their**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=29.152)[**memory space** with other processes. There](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=32.183) [is an exception to this rule, however. A](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=34.581625) [process can explicitly ask the operating](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=36.495666666666665) [system for a segment of memory that can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=38.16766666666666) [shared with other processes, but again,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=40.865399999999994) [that's the exception and requires special](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=42.822) system calls.

[In running Python programs,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=47.01375) [there is one interpreter per process, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=49.50059999999999) [since the GIL is a lock on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=51.36526666666666) [interpreter, there is one GIL for every](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=53.207399999999986) [Python process. With this in mind, it's an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=54.95276923076923) [easy logical step to say that if we want](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=57.86292307692306) [to get around the GIL blocking multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=60.32716666666666) [instructions from running simultaneously](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=61.74) [on different cores, then we can simply](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=63.81942857142857) [create copies of the process. Each](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=66.8478888888889) [separate process will have its own GIL,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=68.97266666666667) [and can therefore run concurrently with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=70.99333333333337) [other processes on different processor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=73.28450000000001) cores.

[In addition to the benefit of being](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=76.12575000000004) [able to **side step the GIL** and concurrently](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=77.58049999999999) [run on multiple cores, there are a couple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=78.78545454545454) [of other advantages to using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=81.33309090909088) [multiprocessing in Python. The separate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=84.19649999999999) [memory space means that there is **less need**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=86.7699090909091) [**for synchronization** primitives when using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=88.15754545454547) [multiprocessing. Like I mentioned earlier,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=91.18299999999998) [there's a way for processes to share](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=92.83866666666668) [memory with other processes, and when that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=94.4883333333334) [happens, then it becomes necessary to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=96.4601818181818) [synchronize access to shared memory. But](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=98.24008333333333) [in the absence of that, code making use of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=100.23858333333334) [multiprocessing can look simpler and more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=103.31200000000001) [straightforward once you take away the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=105.424) [concerns of synchronization. Another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=107.82400000000004) [benefit is that processes are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=109.4768) [**interruptible and killable** using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=111.74079999999998) [OS-provided APIs, where as threads aren't.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=113.552) [And lastly, since processes are **insulated**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=115.78) [**from each other** by the OS, an error in one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=117.83050000000003) [process cannot bring down another process.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=121.2458)

[This is not to say that multiprocessing is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=123.293) [always the better solution. Processes have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=126.1876666666667) [a **higher memory footprint**, and context](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=128.45816666666667) [**switches are more expensive**. However in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=130.9276666666667) [certain cases, like in parallelizing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=133.59949999999998) [CPU-intensive workloads, multiprocessing is invaluable.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=0&mode=live&start=136.1632499999999)

[The Multiprocessing API](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live)

[The Python multiprocessing API was](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=1.534) [designed to be **similar to the threading**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=2.5949090909090904)[**API**, so that in some cases, switching](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=4.1666923076923075) [between the two is as simple as changing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=6.257538461538464) [one or two lines of code. As a result,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=9.019666666666666) [creating a new process is very similar to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=12.464) creating a new thread.

Here's our previous [example where we created a new thread. To](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=16.593600000000002) [use a process to execute the do\_some\_work](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=18.2248) [method concurrently instead of a thread,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=21.45039999999999) [all we need to do is important](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=23.134) [multiprocessing instead of threading, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=24.4675) [construct a multiprocessing that processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=26.380181818181825) [objects instead. You'll also notice that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=28.532909090909104) [in line 6, we have this extra check for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=31.003142857142855) [\_main. We'll get back to why this is here](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=34.62371428571431) [in a few minutes, but first let's take a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=36.34493333333335) [look at the steps needed to spawn the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=38.15518181818182) [process. We start in line 8 by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=40.77990909090911) [instantiating an object of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=43.98449999999999) **[multiprocessing.Process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=45.90216666666665" \t "psplayer)**[. Here we pass in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=45.90216666666665" \t "psplayer) [our target function and any optional](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=48.202666666666666) [arguments needed. **Arguments passed to the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=50.21516666666667)[**process constructor must be picklable.**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=52.663777777777774)

[**Pickling** is process whereby a Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=54.886) [object is converted into a byte stream,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=57.543500000000016) [and unpickling is the reverse operation,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=60.415) [where by a byte stream is converted back](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=61.69899999999999) [into an object hierarchy. In some](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=64.659) [languages, this is known as object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=67.35839999999999) [serialization and deserialization or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=69.14459999999994) [object martialing and unmartialing.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=71.30999999999999) [Pickable objects include types such as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=74.45" \t "psplayer) [none, Booleans, numbers, strings, byte](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=76.91333333333337) [arrays, collections to pickable objects,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=79.03357142857142) [top-level functions, and classes whose](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=81.255) [attributes are pickable.   
After](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=84.779375) [instantiating our process object, we call](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=87.2418) [start on it to kick it off, and then we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=89.03311764705883) [can call join on the object to block until](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=91.033705882353) [the process completes its work and exits.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=94.0369090909091) [The end results of this code is that our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=97.542) [main Python process creates a worker](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=99.18450000000004) [Python process which executes to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=100.5255) [do\_some\_work method and prints out the two](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=102.66799999999999" \t "psplayer) [strings. The child process then exits,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=105.19449999999999) [which allows the main process to exit.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=106.81949999999996)

[Now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=106.81949999999996)[let's talk about the necessity of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=109.45533333333333) [if\_name block in our example. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=112.008" \t "psplayer) [child process gets started, it needs to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=115.60599999999997) [import the script containing the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=118.30460000000002) [function. If the code for creating the new](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=119.58980000000005) [process in the top level of the script, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=121.68749999999996) [gets executed during the import, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=124.5495) [means that a new process is created, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=127.48499999999999) [in turn tries to import the script,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=130.49363636363637) [causing new processes to keep getting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=132.334) [recursively created until a runtime error](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=133.8802) [occurs. Putting the code for creating the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=135.64427272727272) [new process in the \_main block ensures](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=138.21072727272724) [that it only gets run when the script is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=140.3461428571428) [executed, and not when it's imported.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=142.87)

[Here's the full signature of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=146.908) **[multiprocessing.Process constrictor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=150.0013333333333" \t "psplayer)**[. As](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=150.0013333333333" \t "psplayer) [you can see, it's fully identical to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=151.7768181818182) [threading. Thread constructor, so much so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=153.60736363636366) [that the group parameter exists solely for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=155.72609090909094) [compatibility with the threading.Thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=158.633) [constructor. We have the target parameter,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=161.42050000000003) [which is the function to be invoked, The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=163.29008333333334) [name parameter, which sets the process,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=165.53650000000002) [args is the argument tuple for the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=167.99350000000007" \t "psplayer) [invocation, and kwargs is a dictionary of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=171.5450909090909) [keyword arguments for the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=175.05463636363635) [invocation. The daemon argument sets the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=176.8812) [process daemon flag to true or false. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=178.869) [none the default, this flag will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=182.99336363636357) [inherited from the creating process. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=185.76177777777778) [difference between a daemon process and a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=187.522) [non-daemon process is that when a process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=189.15300000000002) [exits, it terminates all its daemon child](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=191.4443333333333) [processes. If a process has child](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=194.23091666666653) [processes that are not daemon processes,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=196.1735454545454) [the process will by default not exit until](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=198.027) [all its non-daemon child processes have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=201.78500000000003) [exited. Therefore, in situations where we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=204.2776) [want a background process that runs](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=206.11840000000007) [without blocking the main program from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=208.43160000000003) [exiting, setting the child process as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=210.981) [daemon will come in handy. It's important](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=213.44325000000006) [to note that a **daemon child process is not**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=215.81384615384616)[**allowed to create its own child processes**,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=218.56715384615387) [so keep that in mind when determining](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=220.40275000000005) [whether to flag a child process as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=222.116) daemon or not.

[As we mentioned before, one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=225.51309090909083) [of the advantages of using processes for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=228.4453333333334) [concurrency is the fact that unlike](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=230.767) [threads, processes are killable via an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=232.2058) [operating system-provided API. In Python,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=234.2580909090909) [two API calls are used to manage the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=236.99081818181816) [aliveness of a process, the **is\_alive**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=240.73300000000003) [method and the **terminate** method. The code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=243.0670000000001) [below shows the methods in action. In line](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=245.966) [10, we define the process, but we have not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=249.14828571428572) [started it yet, so the print statement on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=250.58957142857145) [line 11 will print out an is\_alive value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=252.75933333333327) [of false. After starting the process in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=255.46226666666652) [line 12, the next print statement prints](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=257.85591666666676) [an is\_alive value of true. In line 14,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=260.261) [we're sending the kill signal by calling](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=262.8394615384614) [the terminate method, and then waiting for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=265.3563333333334) [the process to terminate by calling join](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=268.5250000000002) [on the process object, which means that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=270.5066666666666) [the is\_alive value in line 16 will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=271.5980833333332) [false. From line 17, you'll notice that we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=274.5043636363636) [can find out the exit code of our process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=276.9962857142857) [by reading the exit code attribute. A](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=279.0669285714285) [process with an exit code of 0 means that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=281.5073076923078) [no error occurred. A code of greater than](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=284.7652857142857) [0 means that the process had an error and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=287.4824285714285) [exited with that code, and a code of less](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=289.8309999999999) [than 0 means that the process was killed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=291.613) [with a signal of -1 multiplied by the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=293.87330769230783) signal code.

[**While the option of forcibly**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=297.0758000000001)[**killing a child process by calling**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=299.1244545454547)[**terminate is available in Python, the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=300.151)[**better alternative will be to use the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=302.36172727272725)**poison pill method.** [However, if you do](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=304.8364285714286) [decide to go down a terminate route, there](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=307.04992857142867) [are a couple of things that are important](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=309.4340769230769) [to note. One is that if the child process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=311.5756153846153) [uses any shared resources such as locks,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=315.09192307692325) [semaphores, pipes, and queues, and gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=318.2655555555555) [**forcibly killed**, those **resources** may be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=321.9768888888887) [put **in an inconsistent state**, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=324.01399999999995) [therefore made unusable to other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=325.505) [processes. Therefore it's advisable to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=327.9599999999999) [only consider using process.terminate on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=330.2242000000001) [processes which don't use any shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=332.884) [resources. The other thing to note is that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=335.1845714285714) [on a forcibly killed process, **finally**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=338.33825000000013) [**clauses and exit handlers will not get**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=340.634)[**run**. So if you have critical code in a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=342.85300000000007) [finally clause or exit handler, then it's best not to use a terminate method.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=1&mode=live&start=344.98353846153844)

[Process Pools](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live)

[An easy way to set up a fixed pool of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=0.96) [worker processes that can accept and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=3.4659999999999993) [execute tasks in parallel is to use a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=5.100818181818182) [process pool. This comes built into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=8.155363636363637) [Python standard library in the form of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=9.981692307692306) **[multiprocessing.pool](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=11.682" \t "psplayer)** [class. When creating](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=11.682" \t "psplayer) [a multiprocessing.pool object, you can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=15.732) [specify the number of worker processes. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=17.058) [you leave the parameter set to none, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=19.440833333333334) [**by default** the **number of worker processes**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=22.354769230769232) [will be set to the **number of CPU cores**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=23.73215384615385) [available on the machine at the time. You](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=26.412363636363644) [can also optionally specify an ***initializer***](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=29.814636363636364) [***function*** and ***initialization args***. If set,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=32.41645454545455) [each worker process executes the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=35.114) [initialization function once at startup.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=37.52149999999998) [An interesting side note concerning the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=39.929) [initializer function in args is that they](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=41.890454545454546) [don't have to be pickable. This makes it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=44.04392307692307) [useful as a workaround for when you need](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=46.119615384615365) [to pass a non-pickable object to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=49.118000000000016) [worker processes. The final constructor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=52.226000000000035) [parameter is ***maxtasksperchild***. By default,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=55.0655) [this is set to none, meaning that worker](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=57.65438461538462) [processes live as long as the pool is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=60.2039230769231) [alive. However, if the value is set, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=62.24708333333332) [after executing the specified number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=64.15163636363636) [tasks, a worker process is killed and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=66.44745454545455) [replaced with a new worker process. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=69.07199999999999) [ensures that a long-running worker process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=71.6238) [periodically has to release any system](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=73.80659999999999) resources it holds.

[The most common usage](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=77.09200000000001) [pattern for using a process pool is to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=80.25771428571426) [define a function to be executed, an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=82.20114285714277) [iterable of items that serve as the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=84.25991666666665" \t "psplayer) [function argument, and then using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=86.462) **[pool.map](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=88.00100000000002" \t "psplayer)** [method to apply the function to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=88.00100000000002" \t "psplayer) [each value in parallel. Here's an example.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=90.77711111111113) [Here we start by inputting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=95.107) [multiprocessing, and then we define our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=96.43745454545453) [target function do\_work. I've also created](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=98.034) [a function start\_process, which will serve](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=101.93066666666665) [as our initializer function. Our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=104.74214285714287) [initializer function takes in no](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=106.187) [parameters, since it simply prints the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=107.65699999999998) [current processes name. We can access the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=109.82627272727274) [process object for the currently-running](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=113.30272727272732) [process by calling the **multiprocessing.**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=114.8407142857143)[**current\_process** function, and once we have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=116.04233333333333) [a reference to the process object, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=118.89433333333336) [name attribute, it gives us the process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=121.36307692307692) [name. In line 10, we get the number of CPU](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=123.4883846153846) [cores by calling the **cpu\_count** method, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=126.9111428571428) [use it to define the number of processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=129.13085714285717) [that we want to have in the pool. We then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=131.0782857142858) [instantiate our pool object, creating a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=134.85519999999994) [list of integers, and call the pool.map](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=137.488) [method. The map method applies a target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=140.54400000000004) [function to every item in the interable in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=142.81800000000004) [parallel by chopping up the iterable into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=145.43709090909093) [chunks and submitting the chunks, as well](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=147.98190909090914) [as the target function to the process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=150.26361538461538) [pool. Therefore, **both the target function**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=152.4400769230769)[**and the iterable must be pickable**. The **map**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=154.89781818181817) [method is a **blocking call**, and upon](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=158.18276923076922) [completion, it returns a list of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=160.66346153846155) [results in the original order of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=163.01536363636365) [iterable. There is a non-blocking version](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=165.1955454545455" \t "psplayer) [of map called **map\_async**, which also allows](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=168.29033333333336) [you to specify an optional callback](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=171.5859) [function, which can act on the results of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=173.72370000000004) [each function call as they become](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=176.06099999999992) [available. For now, we'll continue to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=177.75899999999984) [focus on the synchronous map method. In](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=179.98109090909088) [line 17, we can call **close** on the process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=182.579) [pool, which prevents any more tasks from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=185.01800000000014) [being submitted to the pool and causes the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=186.89675) [worker threads to begin exiting. And then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=189.0160833333333) [we call **pool.join**, which blocks the main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=191.98940000000005) [process until all the worker processes in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=194.78633333333337) [the pool have exited. You should note that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=196.5211666666668) [the join method can only be called after](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=199.17215384615383) [calling pool.close or pool.terminate.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=202.2775)

[Let's put this into practice in our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=205.414) [thumbnail\_maker example (](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=210.25255555555555" \t "psplayer)**[script3\_process\_pool](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=210.25255555555555" \t "psplayer)**[). We'll use a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=210.25255555555555" \t "psplayer) [process pool to execute the more CPU-heavy](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=212.5701) [perform\_resizing method in parallel, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=214.752" \t "psplayer) [I'd like to do this in a branch, so that I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=215.97116666666665) [can come back to our code in its current](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=219.03114285714287) [state. So in order to do this, I'll commit](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=222.07571428571433) [the work I currently have, give it a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=225.1666153846154) [commit message, and then I'll create and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=227.894) [check out my new branch. Now I can start](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=230.24366666666668) coding in this branch.

[The goal I want to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=234.14150000000004) [achieve is that instead of using a thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=237.2188571428572) [to perform the resize operation, I want to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=238.5914285714287) [use a process pool. So I'm going to start](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=241.31227272727264) [by importing multiprocessing. And then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=245.72774999999996) [I'll go ahead and remove t2, which is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=247.9357692307692) [thread responsible for reading the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=249.88323076923072) [filenames from the img\_queue and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=251.7161) [performing the resize, and I'll also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=254.04960000000003) [delete the code where I put the poison](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=256.271) [pill into the img\_queue. Then I'll go](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=258.84700000000004) [ahead and create my process pool object.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=261.13766666666675) [I'm not processing any arguments to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=263.443) [pool constructor, because I'm fine with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=265.85877777777785) [the default value for the number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=267.97433333333333) [processes. I also don't have any special](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=270.4686666666667) [initialization function that needs to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=272.77638461538453) [run. To use pool.map to run the resizing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=274.72776923076907) [operation in parallel, I need the function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=276.9427142857145) [to apply and the iterable of items that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=278.6849285714286) [the function will work on. So first, I'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=281.2843571428573) [create the target function, I'll call it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=284.20609090909085) [resize\_image, and I'll just copy the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=287.2326363636362" \t "psplayer) [interesting parts of the perform\_resizing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=289.8831818181818) [method. I'll also need to copy over the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=292.1013636363636) [target\_size list for the function to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=295.9460666666666" \t "psplayer) [complete. Next I need the iterable of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=299.34713333333315) [function arguments. This will be the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=301.5264999999999) [sequence of filenames that we will need to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=302.7852857142858) [apply the resize\_image function to.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=304.1738571428574) [Currently, when a thread downloads an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=305.98766666666666) [image, it puts the filename into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=308.921) [img\_queue, what we want to do is to add it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=311.11523076923066" \t "psplayer) [to a list instead, so that it can be used](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=313.88688888888885) [as an input for the pool.map method. So](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=316.114111111111) [we'll delete self.img\_queue, and in its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=319.2324999999999) [place create img\_list, and when the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=322.351) [download is completed, we should append to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=325.2110000000001) [img\_list instead of putting to img\_queue.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=328.5818000000001" \t "psplayer) [We can now write our pool.map statement.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=331.493) [There is no return value in this case, so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=334.13242857142865) [we won't be storing any results into a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=337.44025) [list. We can close and join here, and we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=340.9822499999999) are done, or are we?

[If I run this code, I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=346.4693076923075) [will encounter an error, and the reason](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=350.9908000000001) [for this error is that both the target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=352.5877333333335) [function and the iterable arg must be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=355.408) [picklable. Here our iterable is picklable,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=358.16600000000005" \t "psplayer) [but our function isn't, because in this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=360.18944444444435) [case, Python needs to serialize what is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=362.56674999999996) [the function definition, but also all the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=365.07216666666653) [attributes of the thumbnail\_maker service](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=367.432) [object, one of which is the dl\_queue.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=369.076) [Unfortunately, dl\_queue is an object of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=373.321818181818) [threading.queue, and threading.queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=376.7105454545454" \t "psplayer) [objects are not picklable. To work around](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=379.11690909090896) [this, we will need to remove the dl\_queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=381.66461538461544) [from the thumbnail\_maker service object,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=383.32492307692337) [and instead pass it into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=385.17866666666674) [download\_image method. This means that I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=388.056" \t "psplayer) [have to also remove the reference to self](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=390.46527272727263) [for the dl\_queue since it's now a local](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=394.16725) [variable. In my make\_thumbnails method, I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=398.68525000000005) [need to create the dl\_queue, and then I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=402.0960000000001) [need to pass the dl\_queue as an argument](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=406.0488000000001) [in the threading constructor. And also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=409.00026666666696) [here as well, we'll delete the self](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=410.9484615384615) [keyword in these two places, since](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=412.6020769230768) [dl\_queue is a local variable. Now our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=414.9418571428572" \t "psplayer) [pickling problem should be solved, and we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=418.678) [shouldn't have any issues pickling the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=419.5495000000001) [target function for our pool.map. But](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=421.09600000000006) [before I run this, I want to do a few](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=423.757) [housekeeping things. I don't want process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=426.907) [spinup time and shutdown time to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=428.4240769230768" \t "psplayer) [included in our performance measurement,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=429.9764615384613) [so I'm going to move them out of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=431.1315) [perf\_counter context. I also want to find](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=433.266" \t "psplayer) [out how quickly the perform\_resize method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=436.35172727272743) [runs, so I'll create timers for that,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=439.0332222222222) [start\_resize and stop\_resize. And then I'm](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=442.247" \t "psplayer) [going to log how long it takes. Now I'm](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=445.98407692307677) ready to run.

[And we can see that the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=451.1632307692308) [entire operation completes in 9.5](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=454.8497692307693) [seconds, while the resize operation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=457.5787142857142) [completes in 1.5 seconds. Our initial](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=459.96585714285703) [performance measurements for the resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=463.1528333333334) [operation was 3.19 seconds. So we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=465.1524166666668) [achieved a 2x increase in performance by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=467.91727272727275) [parallelizing the resize using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=470.5957272727274) processor pool.

[In addition to the pool.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=473.59479999999996)[map method, there's the **map\_async** method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=476.1194166666667) [like I mentioned earlier. The map\_async](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=477.3619166666668) [method allows you to specify a callback,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=480.1285454545456) [which gets called for every completed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=483.6792222222222) [function execution with the results. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=486.5545555555556) [map\_async method is also non-blocking,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=488.9058888888888" \t "psplayer) [therefore instead of pausing execution](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=490.647) [until the map operation is complete, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=492.5109999999999) [immediately returns an async result](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=495.14575) [object, allowing the call to store a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=497.0726250000001) [reference to the async result object, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=499.6119999999999) [then continue doing other things. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=502.0299090909091) [caller needs the result of the map](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=503.87727272727255) [operation, they call the **get** method on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=506.5819230769231) [result object, which would then fetch the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=510.4077692307693) [results and block only if the results are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=511.8144999999999) [not immediately available. The pool class](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=513.6315454545455) [also has the **apply** and **apply\_async** method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=516.2971818181819) [which allows you to pass a function and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=519.6013333333334) [its args to be executed on one of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=523.544) [workers in the process pool, whereas pool.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=525.7753571428573)[map limits you to applying only a single](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=527.427) [function on a iterable of arguments. With](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=529.0750000000002) [apply and apply\_async, **you can send any**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=531.7057272727272)[**number of different functions and**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=535.228)[**arguments to the process pool**. In this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=536.607) [example, we send the multiply function and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=538.3874999999999) [a tuple containing the args to the process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=539.7927499999997) [pool to run on one of the processes in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=542.3524999999997) [pool. Since we're using the async variant](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=546.2674999999992) [of the apply method, the results are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=548.3575999999999) [returned immediately and can only be fetched when we call **result.get**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=2&mode=live&start=550.8602500000002)

[Inter-process Communication](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live)

[As we mentioned earlier, by default,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=2.703) [processes do not share memory space with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=4.998272727272728) [each other. This means unlike threads,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=7.372272727272726) [sharing of data between processes is more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=8.75609090909091) [involved than one process, writing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=10.927999999999997) [data to memory, and the other process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=13.147999999999993) [reading the data from memory in a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=15.4652) [synchronized manner. Instead, processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=17.73693333333334) [must use OS-supported communication](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=19.406000000000002) [channels if they want to exchange data](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=21.44600000000001) [with each other. **The two communication**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=24.492249999999995)[**channels implemented in Python are pipes**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=27.28354545454545)[**and queues.** The multiprocessing pipe](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=30.434636363636354) [involves two connection objects, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=32.99057142857143) [represent two ends of a pipe. A process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=35.252399999999994) [can write to one of the pipe, while](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=37.28759999999997) [another process reads from the other end](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=39.56626666666667) [of the pipe, and vice versa. By default,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=42.19920000000002) [**the pipe is duplex**, which means that the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=45.18183333333333) [pipe is bidirectional, but you can specify](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=48.08516666666664) [that you only want a unidirectional pipe.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=50.821000000000005)

[In our example, we have two methods that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=53.635) [will be executed by two different](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=56.281400000000026) [processes. In line 15, we create our pipe](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=58.26620000000005) [and specify that it should be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=60.50207692307694) [bidirectional, which it is by default. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=62.135) [get back a tuple of connection objects,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=64.55758333333334) [which represents both ends of the pipe, as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=67.11133333333332) [we pass one end of the pipe to one of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=70.2051875) [processes, and the other end to the other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=72.53924999999998) [process. In the make\_tuple method, we are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=75.07233333333332) [creating a tuple which has a string and a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=78.01991666666662) [randomly-generated integer between 1 and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=81.25528571428569) [9. We then send that tuple into the pipe](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=83.01421428571425) [using the **send** method, and then listen for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=85.58866666666667) [a response using the **receive** method. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=88.40960000000003) [receive method blocks until a message is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=91.32333333333334) [received, and once the message arrives, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=93.76866666666668) [prints it. The make\_string process is on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=96.24545454545455) [the other end of the pipe listening for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=98.76418181818184) [the tuple. Once it receives it, it creates](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=100.63133333333333) [a string, which is composed of the string](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=102.59523076923078) [in the tuple replicated by the integer in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=104.67845454545454) [the tuple. When it's done with its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=107.03409090909088) [computation, it sends a response back to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=109.16453846153847) [the process of the other side of the pipe.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=111.32269230769232)

[The **pipe is** fairly simple construct **with**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=115.399)[**no built-in locking or consistency**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=118.03261538461541) [**guarantees.** Two processes trying to write](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=119.91376923076928) [to the same end of the pipe can cause data](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=121.87928571428569) [in the pipe to become corrupted. For this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=125.01375) [and other reasons, the multiprocessing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=127.20375000000006) [**queue is the more common method of**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=129.11863636363634)**[interprocess communication.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=132.94636363636351" \t "psplayer)** [The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=132.94636363636351" \t "psplayer) [multiprocessing queue uses a pipe, as well](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=134.2426923076923) [as a few locks and semaphores behind the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=135.6615384615384) [scenes for process safety. It also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=138.67042857142857) [implements all the methods of standard](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=141.7440909090909) [library's queue class, with the exception](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=142.98336363636358) [of the task\_done and join methods. Even](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=144.97966666666665) [the queue that. empty and queue.full](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=147.9573076923077) [exceptions are used to raise timeout](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=149.52746153846152) [exceptions in reading and writing to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=151.08855555555556) [queue. There is a type of multiprocessing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=154.16544444444446) [queue called a **JoinableQueue** that also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=156.46172727272725) [implements the task\_done and join methods](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=158.4407) [for **100% API compatibility with the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=161.03090000000003) [**standard library queue**. Therefore the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=163.83299999999997) [JoinableQueue can be used as a drop and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=165.88700000000003" \t "psplayer) [replacement for the standard queue. Both](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=168.07500000000013) [the queue and the JoinableQueue are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=170.14759999999995) [multi-producer, multi-consumer queues,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=172.436) [which means that multiple processes can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=174.35514285714288) [read from and write to the queue at the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=177.50753846153845) [same time without data corruption](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=180.1784615384615) [occurring. To illustrate the use of a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=182.97324999999995) [queue, let's go back to our earlier](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=185.34242857142863) example.

[We can replace the pipe with a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=187.2504285714287) [queue. Instead of the send and receive](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=189.57884615384617) [methods, we have the standard get and put](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=191.67925) [methods for getting a message from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=194.3852499999999) [queue and putting a message into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=196.86033333333333) [queue. As with the standard library queue,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=199.41591666666665) [when the get is called, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=201.34263636363642) [multiprocessing queue by default blocks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=203.0971818181818) [until an item is available to get from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=205.36309090909083) [queue. Here's a full get method signature.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=208.92820000000003) [If the **block flag** is set to false, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=211.08530769230765) [instead of blocking, the get call returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=214.08300000000003) [immediately with the retrieved item or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=217.09300000000007) [throws the queue that **empty exception** if](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=219.13799999999998) [there is no available item in the queue.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=221.21174999999988) [For blocking calls, a **timeout** can also be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=223.01353846153856) [specified. The put method also has the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=225.52533333333335) [optional block and timeout parameter. When](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=229.04709090909088) [put is called on a queue that has a max](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=231.7539411764706) [size set, if there is no free slot in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=234.56335294117636) [queue, the put call blocks until one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=237.05033333333333) [becomes available or until the timeout](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=238.45266666666666) [expires if one is was specified. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=241.24118181818184) [timeout expires, a queue **full exception**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=245.1052) [gets thrown. If the blocked flag is set to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=247.37979999999996) [false instead, if a full queue is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=249.82599999999994) [encountered, the put call immediately](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=251.514) [throws the queue that full exception.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=253.8703636363636)

[Let's take a look at how we can use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=256.698) [inter-processing communication in our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=260.00775) [thumbnail\_maker application (](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=261.66966666666667" \t "psplayer)**[script4\_threads\_multiprocess\_queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=261.66966666666667" \t "psplayer)**[). In the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=261.66966666666667" \t "psplayer) [current state of our code, we are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=263.42576923076916) [collecting all the downloads into a list,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=265.6614615384613) [and then using a process pool to execute](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=267.9225) [resize operations on multiple processes.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=270.54516666666666) [The process pool uses a type of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=272.48099999999994) [multiprocessing queue under the covers,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=275.295) [but to illustrate interprocess](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=277.6127777777778) [communication using queues, we'll back off](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=279.467) [using the process pool and implement the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=280.95390909090895) [queuing ourselves. So let's start by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=282.9584) [committing the current branch, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=285.2546) [go back to the previous state of our code.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=287.9547692307692) [Now what I'd like to do is after the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=292.308) [download threads downloads each image,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=295.13330769230754) [they should put the filename into a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=296.726) [multiprocessing queue instead of a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=299.0849999999999) [standard library queue. Then on the other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=300.5954545454545) [side, we'll have a number of processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=302.5471818181817) [reading items off the queue and performing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=304.8670000000001) the resize. So let's get started.

[First I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=307.30614285714285) [need to import multiprocessing and change](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=310.33599999999996) [image\_queue from being a queue.Queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=312.25899999999984" \t "psplayer) [object to being an object of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=314.0032) [multiprocessing.JoinableQueue. I'm going](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=315.658" \t "psplayer) [to use a joinableQueue instead of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=318.02049999999997) [regular multiprocessing queue, because the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=321.43300000000005) [joinableQueue has 100% compatibility with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=323.2472222222222" \t "psplayer) [the standard queue and can be used as a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=325.1577777777777) [drop and replacement. So when using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=327.72659999999985) [joinableQueue, I'm able to made as few](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=330.05093333333343" \t "psplayer) [code changes as possible. Now all I have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=332.76506666666694) [to do is in the make\_thumbnails method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=335.50581818181826) [change the thread here to an instance of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=337.588) [multiprocessing.Process. I want more than](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=341.25600000000014" \t "psplayer) [one process performing the resize though,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=343.1242) [so I'll specify a number of processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=344.551) [which will be equal to the number of CPU](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=346.8976153846152) [cores, and then loop that number of times,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=349.63928571428573) [so that we're now creating multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=351.5867142857144) [processes that will read from the queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=353.7625714285715) [and do the resizing. Now here we're only](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=357.13557142857167) [putting one poison pill into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=360.76118181818185) [image\_queue, which means that only one of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=363.30845454545465" \t "psplayer) [our worker processes will terminate. So to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=365.20190909090906) [get all of our worker processes to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=366.8541333333333) [terminate, we need to put as many poison](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=368.3526) [pills as there are workers. We're almost](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=370.51271428571425) [ready to run this code. But first, since](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=373.57771428571414) [we're going to be passing the resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=376.78049999999996) [function to separate processes to execute](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=379.1453333333332) [it, it must be picklable, so I need to put](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=380.61423076923086) [back in the code that we had earlier to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=382.6539285714286) [ensure that the function can be pickled,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=384.94828571428593) [which entailed removing the download\_queue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=387.13020000000006) [from the thumbnail\_maker object, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=389.0672000000001) [instead passing it into the download\_image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=391.948) [method. And now we can run the code.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=396.15999999999997)

[So what we now have is our two operations,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=399.80546666666663) [download\_image, which is IO bound, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=403.015625" \t "psplayer) [perform\_resize, which is CPU bound. Our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=406.82337500000017" \t "psplayer) [IO-bound operation is being parallelized](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=409.2600000000001) [using multithreading, and the CPU-bound](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=411.062) [operation is being parallelized using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=412.86575) [multiprocessing with queues for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=414.89342857142856) [communicating between the different](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=416.7842857142857) [actors. Doing so, we've managed to improve](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=418.3949) [performance by a factor of almost 5x, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=421.05) [still kept our code fairly clean and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=424.05676923076936) [readable. Nicely done. We've accomplished](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=426.4985454545454) [one of the major milestones of this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=427.90490909090903) [course, but there is still a number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=430.0977499999999) [exciting Python features to examine like](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=432.94508333333306) [the asyncio library and the ability to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=434.90391666666665) [share state between processes, even potentially across machines.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=3&mode=live&start=437.148)

[Sharing State Between Processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live)

[Before talking about sharing state between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=0.717) [processes, I should reiterate that as much](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=2.4707142857142856) [as possible, **it is best to keep shared**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=4.239428571428572)[**state to a minimum**, because manipulating](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=6.208) [shared state can be error prone and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=8.670727272727271) [require a lot of complicated](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=11.525) [synchronization steps. Queues and other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=13.53) [message-passing mechanisms are great ways](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=14.997100000000001) [to design systems that minimize or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=16.105600000000003) [eliminate state sharing. However, in cases](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=18.71814285714286) [where program design works best by sharing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=21.593545454545456) [state, Python makes it possible to have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=24.50681818181819) [shared state even between processes.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=28.22463636363636)

[The two options for sharing state between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=31.31390909090909) [processes are **Shared Memory** and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=33.34136363636364) [**Manager Process**. If you only need to share](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=35.28330769230769) [a single variable or array between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=37.644538461538446) [processes, then shared memory is the way](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=39.41250000000001) [to go. The multiprocessing module provides](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=41.47166666666669) [the value and array-shape objects, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=44.34266666666667) [wrap an area of shared memory and can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=47.5308) [inherited by the child processes. To](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=50.54400000000001) [create a value object, we call the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=52.525200000000005) **[multiprocessing.Value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=54.67560000000002" \t "psplayer)** [constructor,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=54.67560000000002" \t "psplayer) [passing in the ctypes typecode, more on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=55.7827) [that later, and the initial value of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=59.23159999999999) [object, and optionally a lock argument](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=62.66299999999998) [that specifies whether or not access to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=65.109) [the value should be synchronized by a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=66.22581818181816) [lock. By default, the lock argument is set](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=67.96661538461538) [to true, which means that an RLock is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=70.90692307692305) [created for synchronizing access to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=74.21138461538459) [value. If set to false, then no lock is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=76.78076923076918) [created, and no synchronization is done.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=79.90671428571426) [You can also use this argument to pass a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=81.93) [lock or RLock object to be used for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=83.81639999999996) synchronization.

[So what's a **ctype**? Ctypes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=86.60962500000001) [are c-compatible data types that are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=89.6585) [necessary for interacting with C libraries](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=92.59549999999996) [from within the Python code. Here's a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=95.0645) [table showing the ctypes and their](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=97.50527272727274) [corresponding Python type. Because the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=99.82509090909096) [multiprocessing value and array objects](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=101.73880000000001) [are ctype objects that are allocated from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=103.62330000000004) [shared memory, the ***ctype or typecode*** needs](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=106.62266666666669) [to specified in the constructor. Here's an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=109.38350000000003) [example of how to create a shared value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=111.86763636363635) [object. For the counter variable, we're](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=114.49818181818178) [initializing it to a shared object of type](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=116.53557142857144) [int. She char I is a typecode for the int](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=119.26928571428576) [ctype. We also haven't specified an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=122.86810000000006" \t "psplayer) [initial value, so it defaults to 0, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=125.01833333333332) [also since we don't specify the lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=127.98769230769231) [variable, it defaults to synchronized. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=131.24753846153845) [next example is a shared Boolean. Here we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=133.91046153846153) [specify the ctype in full and set the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=136.58615384615385) [initial value to false. We also specify](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=139.2027692307693) [that access to this object should not be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=141.44061538461554) [synchronized. Keep in mind that the ***lock***](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=144.37200000000004) [argument is a keywords-only argument. In](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=147.25872727272727) [our final example, instead of using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=149.6831) [built-in lock, we specify the lock object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=152.04279999999994) [that we want to use for synchronizing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=153.79741666666666) [access to our shared object. This comes in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=155.31) [handy if we want to use one lock for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=157.60999999999996) [several objects. We'll take a look at an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=160.65246153846158) [example of shared memory objects in action](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=164.16569230769244) [a little later. But first, let's take a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=166.79462499999997) [look at using manager objects for shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=169.94615384615378) state.

[A **Manager** controls a server](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=172.88776923076907) [process, which maintains objects that can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=175.39527272727273) [be shared amongst other processes. An](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=177.9) [example of shared objects are values,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=180.43527272727275) [arrays, lists, dictionaries, namespaces,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=183.26555555555555) [locks, semaphores, and more. Processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=186.13577777777775) [that access those shared objects do so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=189.40636363636364) [**using a proxy**. To the process, it looks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=192.2089090909091) [like they're interacting with the local](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=194.69454545454545) [object; however, any interactions they](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=196.236) [have with the proxy object is serialized](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=197.8341818181818) [to and from the server process. As a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=200.22714285714287) [result of this, interactions with objects](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=203.5950769230769) [that are shared via a manager are slower](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=205.1463076923076) [than interactions with objects that are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=207.57089999999997) [shared through a shared memory, because](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=209.8346999999999) [**every action involves making a remote**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=212.3865833333334) [**call**. However, the benefit to using a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=214.99608333333347) [manager for shared objects is that you can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=217.4663846153846) [share **much more complex types** than just a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=220.1959230769231) [single value or array. You can share](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=223.03530769230778) [lists, dictionaries, arbitrary objects,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=225.8875) [etc. Another benefit is that with manager](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=227.79749999999996) [processes, we can **share objects across a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=229.73299999999998) [**network**. The manager process may be on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=231.60033333333334) [same local machine as the process using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=234.20966666666666) [its shared object or on a remote machine.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=236.6543076923077) [As long as the proxy object knows how to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=239.522) [connect to the manager process, it doesn't](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=241.85715384615395) [matter. The only difference would be the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=243.80700000000002) [performance of your calls when the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=245.935) [separate processes local versus remote. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=248.39390909090918) [won't discuss remote managers any further](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=250.6619) [in this course, but it's good to know that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=251.97529999999995) [such functionality is available to us when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=255.23471428571423) needed.

[Creating a manager object is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=258.27121428571417) [simply a matter of calling the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=260.3408333333332) [constructor. This spawns the server](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=262.30383333333305) [process that will maintain the shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=264.1102727272727) [objects. This **server process is a child**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=266.3291818181817)[**process of the process that spawns it.**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=268.25992307692314) [Therefore, when the parent process exits,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=270.081) [or if the manager object goes out of scope](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=272.06809090909076) [and is garbage collected, the server](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=275.48342857142853) [process will be terminated. It is also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=278.12257142857135) [important to remember that calling the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=280.6988181818181) [manager constructor spawns a process,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=282.33245454545437) [because if the line where the manager](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=283.92675) [object is instantiated is in the top level](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=286.37616666666673) [of the module, it will get called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=288.6978461538463) [recursively when the new process is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=290.59) [spawned. So care must be taken to ensure](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=292.917142857143) [that the object is instantiated in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=295.1920000000002) \_main block or in the method.

[Here's a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=297.079) [complete list of the **objects that can be**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=300.03814285714293)[**shared** via a manager as of the publishing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=302.1547142857146) [of this course. On the data structure side](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=304.1462307692307) [we have Value, Array, Namespace, List,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=306.01269230769213) [Dict, and Queue, and on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=308.7823636363636" \t "psplayer) [synchronization side we have Lock, RLOCK,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=311.826) [BoundedSemaphore, Event, Condition, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=314.7599999999999" \t "psplayer) Barrier.

[An **example** of how to **use a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=317.63533333333334) [**manager** is shown on the screen. In line 7,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=320.2704615384616) [we create a manager object which starts](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=322.77984615384617) [the server process. We then call the dict](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=325.26215384615386) [method, which creates a dict object on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=327.9456153846153) [server process and returns a proxy for it.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=330.537) [We then pass that dict proxy to our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=332.126) [processes as one of the arguments of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=335.14828571428575) [target function. I'm using list](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=337.76481818181816) [comprehension here for the looping, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=338.6393636363635) [it's tangential to the example. A regular](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=340.30814285714285) [for loop function would function the exact](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=343.55263636363634) same way.

[Now that we've learned about how](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=346.42836363636354) [to share state between processes, let's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=348.23945454545446) [put it into practice (**script4\_multiprocess\_queue\_share\_state**). Let's imagine that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=349.9798) [we're pretty happy with the performance of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=354.1854545454546) [our thumbnail\_maker app, but now we're](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=355.4511818181821) [curious to know how much memory do we save](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=357.79699999999997) [by resizing the original image into three](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=361.4752727272727) [smaller images. To find this out, we know](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=363.74072727272716) [the total size of the files we downloaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=366.0285714285713) [and the total size of the resized files.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=368.216) [Let's start by creating a variable for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=371.88430769230774) [**downloaded file size**, and we'll need a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=375.0142307692307) [lock for this variable, since it will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=376.967769230769) [updated in a download\_image method by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=378.5572727272726) [multiple threads simultaneously. So let's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=379.46) [add lock to the import statement, and pass](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=381.3759999999999) [a lock object to said method. In here, I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=384.9156666666667) [want to do a little bit of cleanup by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=388.78164285714286) [creating an img\_filepath variable so that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=390.6375714285714) [I don't have to repeatedly compute it.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=393.138)

[Next, I'm going to update the dl\_size](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=396.792) [variable within the context of the dl\_size](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=399.46541666666656)\_[lock. I'll use os.path.getsize to get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=402.0999333333333) [the size on disk of the image I just](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=404.9996666666665) [downloaded. So now we've computed the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=408.4978461538461) [total size of the downloaded images. I'm](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=411.0266153846153) [going to go down to the perform\_resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=413.7190000000001) [method and instantiate the dl\_size\_lock,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=416.218) [and pass it to the downloaded\_image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=418.6229090909091) [method. Next, we need to know the **total**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=421.18333333333334) [**size of the resized image**. Since the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=425.63399999999996) [resize operation takes place in different](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=428.7761999999999) [processes, **we** **could** either have each work](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=430.986) [process send the resize\_img sizes back to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=433.89481818181815) [the parent process **using the queue**, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=436.1911538461538) [then let the parent process aggregate the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=437.6708461538459) [sizes, **or** we could use **shared state** and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=439.835) [have each process update the shared\_value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=442.895) [object whenever it completes a resize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=444.81320000000005) [operation. We're going to go with the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=446.8692727272727) [later approach, the shared state approach.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=449.58972727272715) [And now again we have **two options** to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=451.4189999999999) [choose from, using **shared memory** or using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=453.523) [a **manager process**. Since all we need is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=458.17099999999994) [single value and not a list or a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=461.4830000000002) [dictionary, it's best to go with the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=463.2430000000004) [shared memory object in this case. This is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=464.9924999999999) [because the **shared memory** approach is more](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=466.915) [**lightweight** in that we don't need to spin](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=468.2905000000001) [up a whole new process to manage the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=470.5089285714285) [memory, and shared memory access is also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=473.1152142857141) [**much faster**. So let's create a shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=474.9545) [value object that will be inherited by all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=476.5463076923076) [the worker processes. It's of type int,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=478.286) [it's initialized to 0, and it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=482.4707272727273) [synchronized by default. Then in my](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=485.6416) [perform\_resizing method that gets run by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=487.98039999999986" \t "psplayer) [multiple processes simultaneously, I'll do](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=490.062909090909) [the same clean up by creating an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=491.69154545454523) [out\_filepath variable. Now I'm going to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=496.51942857142853" \t "psplayer) [update the shared value with the on disk](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=499.0027999999999) [size of the file I just resized. What](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=500.6283999999998) [about locking, you may ask. If multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=505.2499999999999) [processes are updating the same shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=509.81945454545456) [variable at the same time, isn't there the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=511.49181818181825) [potential for race conditions? Well, yes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=514.6018888888887) [there is. **By default, access to shared**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=517.0533333333333)[**value objects are synchronized using an**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=519.2664999999996)[**internal lock.** In this case though where](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=521.4788181818182)[we have the plus equals, we have two](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=524.4277272727271) [operations, reading the value and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=527.1123333333337) [updating it. **Both** of those **operations** will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=529.1124) [be synchronized internally separately, so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=531.7961999999995) [to ensure that they **run as an atomic unit**,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=533.7402500000001) [**we need to lock both operations**,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=537.272) [preferably using the same internal RLock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=539.7205454545455) [that the shared value uses. We can get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=542.2543846153845) [this lock by calling get\_lock on the value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=546.2014615384611) [object, so we can move this code into a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=550.2840666666668) [with block. And now we're locking the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=555.0294666666673) [update using the resize\_size. value](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=559.0428461538463) objects internal lock.

[The final thing we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=562.0016923076927) [want to do is log the initial size of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=564.5471333333335) [downloads and the final size of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=566.4724666666671) [resized images, and we can go ahead and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=569.8067500000002) [run this. And a few seconds later, we'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=573.9957500000005) [see that we downloaded 5.6 MB of images,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=576.6977142857141) [and after resizing, we now have 199 KB of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=579.5197272727273) [images. Nicely done. We get to save disk](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=585.0612727272733) [space, and now we can transmit perfectly](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=587.642923076923) [sized images to our users, wins all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=589.7126666666667) [around. Now is a good as time as any to wrap up this module.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=4&mode=live&start=592.4053333333336)

[Process Synchronization](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live)

[Before we wrap up the module, I'd like to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=1.358) [bring up something that snuck in to our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=3.027846153846153) [last example, and that's the use of a lock](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=5.298749999999999) [across multiple processes. In the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=8.738437499999998) [following code snippet, we are using a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=10.294384615384613) [lock returned by the get\_lock call to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=11.727230769230761) [synchronize access to the shared memory](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=14.153769230769228) [value across multiple processes. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=17.369307692307686) [reason this is possible is because](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=19.84625) [get\_lock does not return an object of type](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=21.84575" \t "psplayer) [threading.lock. Instead, it returns an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=24.894999999999996" \t "psplayer) [object of type multiprocessing.lock.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=27.660999999999987) [While threading.lock can be used to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=29.7244) [synchronize access to a critical section](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=32.105799999999995) [of code by multiple threads,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=34.040499999999994) [multiprocessing.lock is used to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=35.635" \t "psplayer) [synchronize access to a critical section](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=37.11000000000001) [of code by multiple processes. And it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=39.139818181818185) [not just locks, the **multiprocessing module**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=43.02354545454548)[**contains process-enabled equivalents of**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=45.6525)[**all of the synchronization primitives in**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=46.938500000000005)[**the threading module.** Therefore, when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=49.727499999999985) [synchronizing between processes, you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=51.73444444444445) [should use the primitives for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=52.78733333333335) [multiprocessing module in place of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=55.01372727272727) [threading version. However, since](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=57.88718181818182) [processes have separate memory spaces from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=59.7586) [one another, ideally there should be **very**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=62.49940000000001) [**few situations** other than synchronizing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=65.15299999999999) [access to shared resources where](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=66.893) [significant amounts of interprocess](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=69.04577777777779) [**synchronization is needed. Instead, queues**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=70.768)[**or pipes should be used to pass messages**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=72.94572727272724)**between processes**.

[So to recap what **we've**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=77.394) [**done** in this module, we started out by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=80.48007142857145) [looking at when and why you should use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=82.48178571428579) [**multiprocessing over multithreading**,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=85.87709999999996) [including a discussion on the benefits and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=87.225) [drawbacks of multiprocessing. We then took](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=89.9011) [a look at the **multiprocessing API**, it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=92.32750000000001) [similarities to the threading API, and how](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=95.31300000000005) [to manage processes. We took a look at the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=97.81787500000002) [**multiprocessing pool API**, which makes it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=100.67315384615384) [easier to distribute work over a pool of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=102.53730769230768) [processes, and then we explored](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=105.81483333333334) **[interprocess communication](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=107.934" \t "psplayer)** [with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=107.934" \t "psplayer) **[queues](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=107.934" \t "psplayer)** [and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=107.934" \t "psplayer) [**pipes**. This is the preferred means of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=110.42483333333331) [communication between processes as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=112.74039999999997) [processes don't share memory by default,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=113.952) [but sometimes this is necessary. So we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=116.59690909090912) [looked at ways to **share state between**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=119.27546153846153) [**processes**, which can be done either](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=120.93607692307687) [through the use of **shard memory** or a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=122.7965) [**manager process**, depending on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=126.44249999999995) [complexity of data that needs to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=127.95699999999998) [shared. Lastly, we briefly touched on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=129.36399999999995) [**synchronization between processes**. We now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=133.0625) [two ways of executing code concurrently](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=136.12916666666666) [using threads or processes. As a result of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=139.03366666666665) [the similarities between the two, we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=142.01246153846157) [begin to develop some abstractions that'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=143.8367692307693) [help us standardize the way we interact](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=145.94440000000003) [with threads or processes and deal with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=148.6781) [the results. This is what we'll look at in the next module.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m3&clip=5&mode=live&start=151.15679999999992)

[Abstracting Concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live)

[Abstracting Concurrency Mechanisms](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live)

[Before we jump into asyncio, we need to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=0) [take a quick look at the executing API and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=2.810857142857142) [the future object that gets returned from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=6.228363636363637) [it. We'll start by doing a recap of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=9.283545454545456) [concurrency mechanisms we've looked at so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=11.04945454545454) [far. To create a thread to execute a task,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=13.935999999999998) [we need to define the function that we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=16.582857142857137) [want to run, and then create an instance](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=18.553142857142856) [of the thread class, passing in a target](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=20.881714285714278) [function and any arguments. Then we need](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=23.3825) [to start the thread to begin execution. To](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=25.591100000000004) [wait for a function execution to complete,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=27.915733333333336) [we can call join on the thread. If we want](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=29.943866666666676) [multiple threads to execute that task in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=33.01954545454546) [parallel, then we need to create and start](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=35.46381818181819) [multiple threads in a loop. We'll also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=37.39149999999998) [need to join on those objects in a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=39.03714285714286) [separate loop in order to wait for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=41.10971428571431) execution to complete.

[The API for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=43.63600000000001) [processes is similar, with the difference](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=46.61600000000003) [being that the task to be executed should](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=48.561076923076925) [be a CPU-bound task instead of an IO-bound](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=49.77461538461539) [task. Like the thread, we instantiate the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=53.29314285714284) [process object, start it and join it to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=55.78949999999998) [wait for the completion of the task. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=58.67888888888889) [can also have multiple processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=62.21066666666667) [concurrently executing the function, where](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=63.61400000000001) [we create and kick off multiple process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=64.84818181818183) [objects and join on them in a separate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=66.22081818181822) loop.

[In both cases at a high level, what](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=69.14242857142855) [we are doing is **defining a task**, passing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=72.8725) [it to some sort of **executor**, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=75.73250000000002) [getting the **results** back, or waiting for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=78.59446153846152) [it to complete. For IO-bound tasks, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=81.1015384615384) [executor is a thread, and for CPU-bound](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=84.08400000000002) [tasks, the executor is a process. **We can**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=87.42300000000004)[**turn this high-level description into an**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=90.90433333333333)[**interface**. With this interface, all we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=92.98433333333331) [need to do is instantiate an executor,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=95.27725000000002) [submit the function to execute, and get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=98.002) [back the results. For executing multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=100.67739999999998) [tasks concurrently, we can have a map](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=102.88572727272725) [method that applies a target function to a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=105.36309090909084) [iterable or iterable of arguments using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=107.55566666666662" \t "psplayer) [multiple threads or processes in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=109.422) [background. What this buys is that instead](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=112.73999999999998) [of directly dealing with the underlying](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=114.94524999999999) [threads or processes, we simply submit](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=116.67359999999998) [tasks to the interface, and let the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=118.44) [implementation manage the execution of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=120.7206) [task. We don't need to worry anymore about](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=122.67109090909092) [instantiating the threads or processes,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=125.266) [starting them and joining them, all this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=128.20155555555556) [gets taken care of for us. Now if you have](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=131.333) [a program design where you **need a finer**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=133.92614285714285) [**grain control over** the threads or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=135.98042857142855) [processes that will be running your tasks,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=137.45545454545456) [**then the executor interface is not for**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=139.138)[**you**, and you should stick to the process](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=140.96715384615388) [and threading API. But **if your needs are**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=142.99754545454547) [**straightforward, then the executor**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=144.84772727272738)[**interface makes it easier** to parallelize](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=146.60270000000003)[operations. **Another thing** it buys us is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=149.65010000000007) [that we could **easily switch between**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=151.4838)[**threads and processes.** Switching is simply](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=153.55259999999996) [a matter of re-specifying what type of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=155.6132727272727) [executor you want to instantiate, whether](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=158.00790909090898) [going from a thread pool executor to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=159.63874999999996) [process pool executor, or vice versa, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=161.443) [the rest of the API stays the same. Now in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=164.42406666666676) [reality, this benefit may not be very](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=168.33) [practical if you've appropriately divided](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=170.0765000000001) [up your tasks into IO bound and CPU bound.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=171.82463636363633) [Your CPU-bound tasks should run on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=176.323) [processes, and there should rarely be a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=177.7962727272728) [need to switch, and your IO-bound tasks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=179.726) [should run on threads, and there should](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=182.183) [barely be a need to switch either. But](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=184.1573) [there might be cases where your tasks mix](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=185.79683333333332) [IO and CPU, therefore you'd like to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=188.26749999999993) [maintain the flexibility of running on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=190.56709090909092) [either, depending on your platform, etc.,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=192.62672727272732) [or you might foresee a migration to a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=194.57750000000004) [GIL-less interpreter, so you want to keep](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=197.21081818181818) [your options open. Using the executor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=199.89054545454545) [interface will provide the benefit of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=201.97546153846153) [making that switch easier. So now that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=203.84838461538462) [we've introduced the why of the executor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=205.93972727272728) [interface, let's dive into it and how to use it.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=0&mode=live&start=207.90609090909092)

[The Executor API](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live)

[The executor API is provided by the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=1.522) **[concurrent.futures](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=4.3913" \t "psplayer)** [module, and it exposes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=4.3913" \t "psplayer) [only three methods, **submit**, **map**, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=6.912500000000001) [**shutdown**. The submit method schedules the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=9.495500000000003) [passing function to run on one of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=11.65369230769231) [executor's workers and returns a future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=14.418) [object that represents the execution state](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=15.845999999999997) [of the function. We'll talk more about the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=17.528181818181817) [return future object a little later, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=20.448909090909094) [it's important to know that the **call to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=22.322307692307696)[**submit is non-blocking**. The submit method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=24.5438) [immediately returns the future object to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=27.314600000000002) [the caller, so that the caller can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=29.695727272727268) [continue executing and get the results of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=32.0190909090909) [the computation at a later time by calling](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=34.7343846153846) [the **results** method on the returned future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=37.95366666666667) object.

[The **map** method uses the executor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=41.02433333333336) [worker pool to apply the passed-in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=43.393999999999984) [function to every member of the iterable](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=45.325) [or iterables concurrently. Each worker](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=48.55550000000001) [concurrently operates on an item from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=50.39333333333333) [iterable or a tuple of items from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=51.98" \t "psplayer) [iterables until all the items are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=54.745000000000005" \t "psplayer) [processed. Therefore the degree of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=57.247000000000014) [concurrency depends on the number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=58.33018181818183) [workers in the worker pool. The map](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=59.32945454545457) [function returns an iterator that can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=62.11100000000001) [raise a timeout exception if a timeout is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=64.726) [set and the value is not available by the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=66.85971428571429) [timeout period. If the timeout is set to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=69.72375) [none, then there is no limit to the wait](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=73.09375) [time. The ***chunksize*** parameter is only used](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=75.68361538461535) [by the process pool executor. It defines](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=77.518) [how to chop up the iterable into chunks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=79.19449999999995) [when submitting the tasks to the workers](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=81.54423076923075) [in the pool. By default, the value is set](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=84.23276923076918) [to 1, which means that each item in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=87.283) [iterables get send individually to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=90.13" \t "psplayer) [worker on the task. But for larger](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=91.79319999999996) [iterables, there might be some efficiency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=94.00654545454547" \t "psplayer) [gains that may be had by setting this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=96.21563636363642) value to a number larger than 1.

[The third](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=98.6855384615385) [method is the **shutdown** method. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=101.412) [shutdown method is used to signal to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=103.52400000000003) [executor that no more tasks will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=105.8214285714286) [submitted to it, and that it should free](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=107.37792857142864) [up any resources that it's currently using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=109.20958333333334) [once the currently-running tasks, if any,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=110.84) [are done executing. Once an execute is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=113.70466666666667) [shut down, any subsequent attempts to call](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=116.52499999999998) [submit or map on it will result in a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=118.95399999999994) [runtime error. The ***wait*** parameter](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=122.77460000000004) [specifies whether the shutdown method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=124.78549999999998) [should block or not. If set to true, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=126.68966666666662) [the method will not return until any](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=129.51223076923077) [pending tasks have completed, and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=131.33115384615385) [executor pool has been shut down. If set](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=133.2196363636364) [to false, the shutdown method returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=136.02155555555558) [immediately, but it should be noted that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=138.5642222222223) [your application will not exit until all](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=140.78200000000004) [the pending tasks are done executing.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=142.875)

[Like several other resources in Python,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=146.23690909090914) [**executor instances** can be called within a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=148.4263) [with context, in which case it would be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=150.30440000000007) [unnecessary to call the shutdown method,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=152.62150000000003) [as it would automatically get called when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=154.402) [leaving the with block. The executor class](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=157.43045454545452) [is an abstract class, so to use the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=160.46923076923073) [executor API, we need to instantiate one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=163.95784615384602) [of its concrete subclasses,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=166.4038888888889) **[ThreadPoolExecutor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=167.663" \t "psplayer)** [or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=167.663" \t "psplayer) **[ProcessPoolExecutor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=167.663" \t "psplayer)**[.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=167.663" \t "psplayer) [For IO-bound tasks, we should instantiate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=172.311) [a ThreadPoolExecutor. When we instantiate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=175.69950000000003) [the ThreadPoolExecutor, we can specify the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=177.5655) [maximum number of works and the thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=179.03849999999994) [name prefix. the maximum number of workers](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=181.61050000000006) [determines, like the name implies, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=184.103) [maximum number of worker threads in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=186.18439999999998) [thread pool. If the value is set to none,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=188.7351999999999) [then it will default to the number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=190.69460000000007) [cores in the machine, multiplied by 5. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=192.101) [thread name prefix allows you to specify](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=195.25300000000004) [the prefix for the thread names. This can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=197.8673333333333) [make debugging easier when you print out](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=200.4876923076923) [the thread name in the application logs.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=202.3906153846154) [In this example, we use a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=205.75) [ThreadPoolExecutor to attempt to download](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=208.02018181818187" \t "psplayer) [two HTML pages concurrently. We start by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=209.912) [importing the necessary modules, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=213.35599999999994) [defining the load\_url task, which opens a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=215.7753636363636) [connection to the specified URL and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=217.97654545454537) [downloads the HTML content in that URL.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=220.42016666666672) [Then in a width block, we instantiate an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=223.401) [object of the ThreadPoolExecutor with a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=225.28536363636357) [max\_workers parameter of 2, and refer to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=227.56099999999998" \t "psplayer) [it as executor. We then submit two tasks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=231.22199999999995) [to the executor, one to download a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=234.7875384615385) [legitimate web page, cnn.com, and another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=237.5826923076924) [to download a page we know doesn't exist.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=241.01328571428573) [The submitting of the tasks is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=245.055) [non-blocking and returns a future object.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=247.13049999999993) [So both tasks get submitted immediately](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=249.206) [one after the other. To get the results of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=252.28580000000005) [the computation, we call the result method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=255.62669230769237) [on the future object. This is where we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=257.4294615384615) [actually get the downloaded bytes array](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=260.48583333333335) [returned by the load\_url method, and refer](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=262.8582857142857) [to it using the data variable. If the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=265.4712857142857) [download is not complete, then the call to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=268.611) [result will block. We expect request 1 to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=272.2109999999999) [succeed and request 2 to fail, because](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=274.7191818181818) [it's being made to a nonexistent URL. But](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=276.679125) [notice that the try catch isn't around the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=280.8670769230769) [submit, but rather it's around the result](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=282.4436923076922) [method. This is because exceptions that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=284.2979) [happen during the execution of submitted](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=286.4297) [functions don't get raised to the caller](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=288.87454545454534) [during the submit or map operation.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=291.7322857142857) [Instead, they get raised when the result](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=293.834) [method is called in the case of a submit](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=295.5436153846156) [method or when the resulting iterator is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=298.11524999999995) [being iterated over in the case of the map](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=300.6964999999998) [method. The last thing I want to point out](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=303.06219999999996) [here is that because we make use of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=305.1735) [width statement, we don't need to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=307.4279999999999) [explicitly shut down the executor. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=309.09833333333324) [happens automatically for us when we exit](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=311.1419090909091) the with block.

[For CPU-bound tasks, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=315.21527272727286) [should instantiate a **ProcessPoolExecutor**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=319.4435000000001) [The only parameter in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=321.926) [ProcessPoolExecutor constructor is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=323.21711111111114" \t "psplayer) ***[max\_workers](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=324.25" \t "psplayer)*** [parameter, which determines](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=324.25" \t "psplayer) [the maximum number of worker processes in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=326.45044444444443) [the pool. If this value is set to none,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=329.88700000000006) [then by default it will be set on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=332.9740000000002) [number of processor cores on the machine.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=335.4522307692309) [The ProcessPoolExecutor uses](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=337.143) [multiprocessing under the hood, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=338.79428571428576) [means that just like we discussed in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=341.31054545454543) [last module, the **target function and the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=343.82690909090906)**[iterables must be picklabe](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=346.6787777777778" \t "psplayer)**[. In this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=346.6787777777778" \t "psplayer)[example, we're using a ProcessPoolExecutor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=349.1732222222222) [to concurrently generate a hash of each](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=350.9637777777777) [input text. We start by importing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=354.26964285714286) [necessary modules. Next, we define a list](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=357.89514285714284) [of byte strings and our callable function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=361.1683076923077) [generate\_hash. The generate\_hash function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=364.30053846153845" \t "psplayer) [returns a sha2 hash of the passed in byte](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=365.8423076923077) [string. Then we create a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=369.1252307692309) [ProcessPoolExecutor instance in a with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=371.126" \t "psplayer) [block, and use the map method to run the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=373.17099999999994) [generate\_hash function on all the items in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=376.2292307692306" \t "psplayer) [a text iterable concurrently using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=379.02566666666667) [ExecutorPoolWorkers. The resulting](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=382.4476666666667" \t "psplayer) [integrator is zipped together with a text](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=383.61436363636363) [iterable into a tuple, and then we print](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=385.70163636363645" \t "psplayer) [out the byte string and the hash. It's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=388.188) [important to note that the instantiating](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=390.6725555555555) [of the ProcessPoolExecutor was done **inside**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=392.3498888888887) [**of a \_main block**, as opposed to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=394.45081818181814) [ThreadPoolExecutor, which didn't need to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=397.76814285714283" \t "psplayer) [be. As we discussed in the last module,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=400.863857142857) [this is done to protect your application](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=403.1866923076924) [from infinitely creating new](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=405.1605714285714) [ProcessPoolExecutor instances in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=407.2148571428571" \t "psplayer) [process of creating new worker processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=409.13892307692305) [until our runtime error occurs. When a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=411.82969230769214) [newly forked process importing the script](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=414.7152727272727) [from the parent process, it won't execute](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=416.64509090909075) [the code inside of the \_main block.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=418.7878181818181) [Therefore we avoid the infinite recurrent problem.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=1&mode=live&start=420.849)

[The Future Object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live)

[Let's talk in more detail about the future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=2.256) [object, which is returned by the **submit**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=3.8700000000000006) [method. As an FYI, the map method, even](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=5.872153846153846) [though it doesn't **return a future object**,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=9.059230769230771) [also uses an iterable of future objects](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=11.347818181818178) [internally in its implementation, so it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=13.507333333333333) [certainly a worthwhile exercise to take a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=15.723333333333334) [few moments to understand what a future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=18.405333333333328) object is and how to use it.

[A **future**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=21.126) [**object** is an object that acts as a proxy](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=23.84) [for a result that is initially unknown,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=25.473499999999998) [typically because a computation of its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=27.23869230769231) [value **is not yet complete**. In Python, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=29.811769230769244) [future object takes on the extra](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=33.27136363636364) [responsibility of encapsulating the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=35.90209090909092) [execution state of the computation,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=37.455285714285715) [allowing the developer to manage that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=39.146) [state to some extent and be notified or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=40.34749999999998) [perform some action when the computation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=42.465399999999995) [completes by making use of callbacks. By](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=45.21459999999998) [being strictly non-blocking and returning](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=49.785666666666664) [a future object, the executor API drives](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=51.954) [us to think about programming](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=54.70820000000001) asynchronously.

[In asynchronous program,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=56.524700000000024) [we typically focus on the main thread. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=58.18280000000001) [**actor** that actually executes the task is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=61.58190909090909) [concealed from us. The main thread simply](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=64.22727272727273) [submits the task to the actor, which **may**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=66.84930769230768)[**be another thread, process, or OS**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=69.82592307692302)[**function**, and then continues executing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=72.19649999999997) [until it needs the result from the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=74.195) [execution or until the execution is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=77.40099999999998) [completed. In Python, the future object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=80.14899999999997) [enables asynchronous programming. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=83.57387499999997) [**executor represents the actor** and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=85.634) [immediately returns the future object so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=87.034) [that the main thread is not blocked and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=88.84625) [can go on doing other things. When the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=91.61525) [main threads needs the result, then it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=94.04199999999999) [calls the **future.result** method to get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=95.26699999999997) [back the result of the function. If the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=97.86836363636367) [function is not yet completed, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=101.03666666666666) [future.result will block until it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=102.43666666666667" \t "psplayer) [completes or until a timeout occurs if one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=104.191) [is specified. Also important to remember](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=107.475) [is that as I've mentioned earlier, if the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=110.07080000000002) [execution of the task resulted in an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=113.06284615384615) [exception, then when you call the future.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=114.21730769230767)[result method, the exception will get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=116.34815384615383) raised at that time.

[The **future object has**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=119.29046153846151)[**a number of methods** that allow you **to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=122.19085714285715) [**manage the execution of a task.** The **cancel**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=124.06857142857147) [method allows you to attempt to cancel the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=126.17385714285714) [execution of the task. If the function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=128.85728571428567) [call is currently being executed and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=130.9320909090909) [cannot be cancelled, then the method will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=132.39827272727268) [return false; otherwise, the call will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=134.6001818181818) [cancelled and the method will return true.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=137.24937500000001) [The **done** method is usually checked when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=141.403) [the function call has completed execution](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=142.74127272727281) [or was successfully cancelled. In both](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=144.90866666666668) [cases, it returns true, otherwise it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=148.899) [returns false. The **exception** method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=152.454) [returns the exception raised by the call](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=154.61600000000004) [if any. If the execution is not yet](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=156.89566666666678) [completed and a time out is specified,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=159.06176923076927) [then the method will wait until the time](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=160.829) [out expires. On expiry of the timeout, a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=163.63940000000008) [timeout error is raised. If there are no](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=166.84699999999995) [exceptions raised by the task execution,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=169.5342) [then the method returns none. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=171.06) **[add\_done\_callback](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=174.3708000000001" \t "psplayer)** [method attaches a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=174.3708000000001" \t "psplayer) [callback function to the future that will](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=176.578" \t "psplayer) [be executed on completion or cancellation.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=179.88038461538468) [The callback function accepts only one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=182.711) [argument, which is the future object.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=185.01650000000006) [**Multiple callback functions can be**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=187.322)[**attached** to a future object and will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=188.62472727272726) [called in the order of which they were](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=190.98690909090908) [added. If the future has already been](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=194.18254545454545) [completed or cancelled, the attached](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=195.89266666666674) [callback functions are called immediately.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=197.61771428571427" \t "psplayer) [And if an exception is raised within a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=200.292) [callback function, the exception is logged](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=203.2192000000001" \t "psplayer) and ignored.

[When you have a collection of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=205.9506666666667) [future objects, you may want to wait for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=209.95) [all of them to complete. The concurrent.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=211.68230769230757)[futures module has a function for this.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=213.57544444444443) [The concurrent.futures **wait** function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=216.412) [takes in an iterable of future objects and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=217.9806363636364) [blocks until the futures are completed.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=220.598) [The timeout parameter can be used to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=222.803) [control how long to wait before returning.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=225.31249999999994) [If it's set to none, then the wait time is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=227.822) [unlimited. By default, the wait method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=231.11472727272724) [waits until all futures are completed as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=233.24274999999994) [specified by the return\_when parameter,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=235.761) [but the return\_when can also be set to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=237.80549999999994) [first completed, which then means that the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=240.92075000000003) [method returns when the first future from](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=243.41916666666677) [the collection completes or is cancelled.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=246.48709090909088) [If you don't want to wait for all the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=249.312) [futures to complete but would rather](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=250.8008571428572) [process the results as they complete, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=251.97872727272727) [there is the **as\_completed** function. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=254.43381818181825) [as\_completed function takes a group of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=257.03533333333337" \t "psplayer) [future objects and returns an iterator](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=260.134) [that yields futures as they complete. Any](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=263.2885000000001) [futures that are completed before the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=266.7482) [as\_completed function was called will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=268.5794000000001" \t "psplayer) [yielded first. If set, the timeout](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=270.3299) [parameter determines how long to wait for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=271.9997) [the completion of the futures. By default,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=274.02618181818195) it's set to none.

[So that was a quick look](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=276.98033333333336) [at the executor API and the future object.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=281.49900000000014) [We started off talking about how the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=283.531) [executor API abstracts away the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=285.50791666666674) [concurrency mechanism and what that buys](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=286.92) [us. We then discussed how to use the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=290.24742857142854) [executor API with the concrete](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=292.94341666666685) [ThreadPoolExecutor and ProcessPoolExecutor](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=294.473" \t "psplayer) [classes. And lastly we looked at the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=297.14675000000005) [future object and how it enables](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=299.265) [asynchrony in Python. In the next module,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=300.492) [we'll dive deeper into asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=303.150090909091) [programming as we'll be introduced to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=304.9909) [latest member of the Python concurrency family, the asyncio module.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m4&clip=2&mode=live&start=307.2442000000002)

[Asynchronous Programming](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live)

[Introduction to Single Threaded Asynchrony](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live)

[This single-threaded asynchronous model is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=0) [a concurrency model in which a single](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=4.760545454545454) [thread achieves concurrency by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=7.165363636363635) [interleaving the execution of multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=8.512) [tasks. Typically, a single thread executes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=10.092000000000004) [tasks sequentially. If you want to execute](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=12.749) [multiple tasks simultaneously, you need an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=16.230333333333334) [array of threads or processes executing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=18.6578) [those tasks in parallel. With](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=20.807) [single-threaded asynchrony, we can achieve](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=23.364857142857144) [concurrency on a single thread by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=25.7605) [interleaving the execution of the tasks.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=28.210272727272724) [While executing a task, let's say t1, a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=30.601) [thread can pause the execution of that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=32.617615384615384) [task and then work on another task, say](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=34.452533333333335) [t4, and then come back and resume t1 at a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=36.750666666666675) [later time. **Single-threaded asynchrony** is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=40.54063636363637) [most often applied **to IO-bound tasks**,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=42.894272727272735) [because IO-bound tasks typically have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=45.24599999999999) [lot of idle time while waiting for an IO](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=47.50615384615385) [operation to complete. During that idle](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=50.91853846153848) [time, the thread can be working on another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=52.9223076923077) [task and then come back to the original](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=55.232461538461564) [task after the IO operation is completed.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=57.42676923076925)

[Implementations of single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=59.313) [asynchrony are typically **based on**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=60.537000000000006)[**event-driven architectures**. An](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=62.577) [event-driven architecture is a software](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=64.488) [design that orchestrates behavior around](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=67.537125) [the production, detection, and consumption](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=70.04275000000001) [of events, as well as reaction to those](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=71.90749999999998) [events. While event-driven architectures](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=74.20749999999994) [have existed for a long time, particularly](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=75.69233333333334) [in GUI development, two products are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=78.39280000000001) [pushed for public understanding of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=80.22760000000004) [technique as a solution for general IO](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=82.17069999999998) [multiplexing, particularly on the server](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=84.564) side, NGINX and NodeJS.

[NGINX ditched the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=87.28066666666668) [traditional model for serving web](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=93.63433333333334) [requests. In the traditional model,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=95.86155555555558) [there's a thread or process that's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=97.2502727272727) [assigned to each connection. The issue](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=98.66518181818175) [with that is that any time the thread or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=100.79561538461536) [process sends a message to the client, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=104.03857142857143) [would have to block for awhile waiting for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=105.73114285714288) [the response. This results in a lot of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=107.6565) [waste, as threads and processes are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=109.96983333333327) [heavy-weight system objects. **NGINX chose**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=112.89025000000002)[**to go with an event-driven model** where](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=114.89883333333333)[there's only a single worker process for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=116.40325) [many connections. An event on the listen](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=119.11266666666666) [socket means that a new client is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=121.73423076923073) [available, and the worker process responds](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=123.95915384615377) [by creating a new non-blocking connection](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=126.38033333333331) [socket. An event on the connection socket](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=128.9043333333333) [means that there is new data to read or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=131.38862500000005) [write. Here the worker performs the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=134.48406250000014) [appropriate response and moves on. **The**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=136.8955)[**worker process never blocks, instead it**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=139.28772727272727)[**immediately goes on to process other**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=141.2360909090909)[**events**, as a result, NGINX is able to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=143.17915384615387) [scale to support hundreds of thousands of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=145.75576923076935) [connections per worker process, and this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=148.34100000000004) [caught people's attention. **NodeJS** expanded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=151.36120000000003) [the popularity of server-side event-driven](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=154.22299999999998) [architecture by bringing it to the popular](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=156.123) [JavaScript language. With NodeJS, terms](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=159.69222222222226) [like **asynchronous IO and event loop** became](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=161.72275) [a common part of web developer lingo. As](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=164.08116666666666) [NodeJS provides the tools for developers](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=166.6335454545454" \t "psplayer) [to build highly-scalable web servers using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=168.6158888888889) [callbacks that signal the occurrence of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=171.11722222222215" \t "psplayer) events and completion of tasks.

[So what is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=174.07455555555552) [the benefit that the single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=177.87172727272727) [asynchronous model provides? In the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=179.7762727272727) [traditional model of concurrency, IO-bound](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=181.7728) [tasks are typically handled by multiple](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=183.90729999999994) [threads, with each thread handling a task.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=186.25599999999997) [When those threads have nothing to do](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=188.748) [because they are blocked and waiting on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=190.6759166666667) [IO, the operating system suspends the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=192.63238461538464) [thread and picks up a thread that is ready](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=194.37053846153856) [for execution. This model works well up to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=196.95838461538463) [a point, but doesn't scale well to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=199.23715384615392) [handling thousands of IO operations](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=202.0414545454546) [because of the memory overhead and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=204.141) [scheduling and switching costs associated](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=206.38680000000002) [with threads. A more efficient solution is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=208.71533333333332) [to have **one thread handle multiple**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=212.09607692307694)[**IO-bound tasks**. When the task needs to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=214.15776923076928) [wait for some IO operation to complete,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=216.22249999999997) [instead of blocking, the thread suspends](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=218.151) [the task and moves on to a task that is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=219.82130769230776) [ready to execute. When the IO gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=222.99263636363634) [completed, the thread gets notified, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=225.84545454545443) [then it can resume the task that it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=227.79478571428572) [suspended. If this task suspension and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=230.2616428571429) [resumption sounds familiar, that's because](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=231.95562500000005) [it's a similar concept to what the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=233.3459090909091) [operating system does with threads. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=235.30209090909088) [difference here is that **we don't pay the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=237.197)[**higher memory cost for multiple threads,**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=239.869)[**and we don't have the overhead of context**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=240.79463636363644)[**switching**. We can switch between tasks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=243.22560000000001) [**much quicker and more efficiently**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=246.54480000000004)

[In order to implement this asyncio model,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=249.29749999999999) [most languages and frameworks turn to an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=251.9069) [**event loop**. In the simplest of terms, an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=254.77619999999996) [event loop is responsible for taking items](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=257.14150000000006) [from an **event queue** and handling it. An](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=259.02162500000003) [event could be a change of state on a file](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=261.64213333333333) [when new data is available to read, a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=264.4234666666669) [timeout occurring, some new data arriving](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=267.3845714285714) [on a socket, etc. The thread goes into a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=270.1574285714285) [loop and checks for an event it needs to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=273.48568749999987) [response to. Its response may include](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=276.1491249999997) [executing a callback or some other code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=277.97399999999993) [that relied on the occurrence of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=280.13174999999984) [event. The code currently being executed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=282.3665454545455) [may generate more events that need to be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=284.327) [watched for. When that happens, the loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=287.16700000000014) [suspends execution of that code and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=289.3397272727273) [continues executing other code until the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=291.095) [event occurs. There are several ways of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=294.93124999999986) [implementing the event loop and mechanism](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=297.43954545454545) [for pausing execution, being notified of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=298.915) [the completion of the IO task, and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=301.41809090909084) [resuming execution. In the next section,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=304.57125) [we'll take a look at how Python implements the event loop and task execution.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=0&mode=live&start=306.70575)

[Cooperative Multitasking with Event Loops and Coroutines](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live)

[In **NodeJS** programming, the **event loop is**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=1.791) [**invisible to the programmer** and is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=4.4306363636363635) [implemented within the VM execution engine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=6.626384615384616) [and in the libev library. But **in Python**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=8.688538461538464) [**the event loop is explicit**. The event loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=11.282250000000001) [in Python is responsible for scheduling](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=13.902000000000001) [and executing tasks and callbacks in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=16.224000000000004) [response to IO events, system events, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=18.366799999999998) [application context changes. To get an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=21.125500000000002) [instance of an event loop, we call the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=24.115272727272725) **[asyncio.get\_event\_loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=26.444" \t "psplayer)** [method. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=26.444" \t "psplayer) [method returns an object of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=28.33377777777778) [abstract\_event\_loop. As the name implies,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=30.696" \t "psplayer) [abstract\_event\_loop is an abstract class](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=33.810375" \t "psplayer) [which has concrete subclasses. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=36.65566666666666) [get\_event\_loop determines the appropriate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=39.058625) [concrete implementation for the platform](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=41.30674999999999) [we're running on and returns it. However,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=43.011333333333326) [the only methods which we concern](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=45.061181818181815) [ourselves with are those exposed in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=47.69627272727271) abstract\_event\_loop class.

[After we get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=50.9092) [the event loop instance, we can **start it**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=52.80538461538461) [by calling the **abstractEventLoop.**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=54.964615384615364)[**run\_forever** method or the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=56.416000000000004) **[AbsractEventLoop.run\_until\_complete](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=58.52" \t "psplayer)** [method. If we start an event loop by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=61.57266666666667) [calling run\_forever, then we can stop it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=64.83550000000002) [by calling AbstractEventLoop.**stop**. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=66.572) [causes the event loop to exit at the next](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=69.14245454545456) [suitable opportunity. Once an event loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=73.30907692307692) [is in the stop state, then we can close it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=75.47323076923074) [by calling **close**. This method closes a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=79.69066666666666) [non-running event loop by clearing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=82.07433333333333) [queues and shutting down the executor. In](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=83.56883333333332) [the previous section of this module, I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=86.1792727272727) [mentioned that the task running in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=89.01753846153846) [loop must be suspended when it encounters](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=91.14230769230772) [an IO operation or any other long-running](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=93.39181818181817) [operation that can be offloaded to another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=95.95318181818176) [actor. While this happens implicitly in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=98.64818181818185) [some other platforms, in Python the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=101.004) [**running task itself is responsible for**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=103.29327272727276)[**suspending itself and yielding control to**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=105.54616666666666) [**the caller** so that the caller can run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=107.61716666666669) [other tasks. **When** the IO operation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=110.48510000000002) [**completes**, the call can then **restore the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=112.76930000000004) [**task back to the state it was in before it**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=114.20120000000003)[**suspended** it and resume execution. This is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=116.24040000000001) [**called cooperative multitasking, and this**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=119.47720000000002)**is where coroutines come in.**

[There are two](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=121.98675)[constructs in Python call coroutines, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=125.49690909090909) [it's important to understand which one is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=127.23472727272726) [being referred to when you hear or read](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=129.48275000000004) [the word coroutine. They are the **coroutine**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=132.1527500000001) [**function** and the **coroutine object**. For now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=135.36377777777773) [let's focus on the former. The coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=138.87533333333332) [function has a special function that can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=141.4979999999999) [give up control to its caller without](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=144.14400000000006) [losing its state. Here's an example of a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=146.80750000000015) [coroutine function. The addition of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=149.3482857142857" \t "psplayer) **[async keyword turns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=151.3593333333333" \t "psplayer)** [the say\_hello](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=151.3593333333333" \t "psplayer) **[function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=151.3593333333333" \t "psplayer)**[**into a coroutine function**, which now makes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=153.53633333333326) [it suitable for use within an event loop.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=156.72123076923077) [We create our event\_loop instance and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=160.656) [call run\_until\_complete, passing in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=164.0354444444445) [say\_hello routine. Whilst the say\_hello](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=166.20400000000004" \t "psplayer) [routine completes, the loop stops, and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=168.2090000000001) [loop.close executes. Not much is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=172.29266666666672" \t "psplayer) [happening in this example, so let's add](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=175.12519999999998) [just a little bit extra. In this example,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=177.43799999999996) [you're introduced to the **await** **keyword**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=180.374) [The await keyword tells Python to pause](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=182.942) [the execution of this coroutine at this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=184.59463636363634) [point, and return control to the event](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=186.75220000000002) [loop until an event occurs. In this case,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=189.58300000000008) [the event that we need to wait for is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=192.366) [timeout event that'll happen after one](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=194.90266666666676) [second. At this point, if there are other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=197.37466666666663) [tasks schedule in event\_loop, those tasks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=201.304) [will start running. After 1 second, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=204.01133333333328) [event\_loop will detect the timeout event,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=206.82979999999998" \t "psplayer) [and then resume the delayed\_hello](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=209.027) [coroutine where it left off, and print the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=211.11750000000004" \t "psplayer) [word world. A coroutine can wait on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=214.405) [events, return results when completed, or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=217.198) raise exceptions if needed.

[In Python **3.4**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=220.7242222222222) [where a native support for coroutines](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=224.80600000000004) [didn't yet exist, generators were used to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=227.1760000000001) [implement coroutines, and instead of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=229.3915454545454) [await statement, we would use the **yield**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=230.9373636363635) [from to transfer control back to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=233.8345714285714) [caller. The **@asyncio.coroutine** decorator](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=236.91407142857136) [was also used to indicate that a function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=239.16600000000003) [was a coroutine function, as opposed to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=242.51463636363636) [the async keyword, which was introduced](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=243.75809090909092) [from Python 3.5 on. We won't spend much](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=245.48285714285714) [time in this module on earlier](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=249.05842857142858) [implementations of asyncio, but it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=251.7267142857143) [interesting to know that many of the basic](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=253.66672727272726) [asyncio functionality from Python 3.5 on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=255.80563636363627" \t "psplayer) [are also available in Python 3.4, howbeit](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=258.400076923077) in a less-polished fashion.

[Executing a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=261.982) [coroutine function doesn't result in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=266.2887272727273" \t "psplayer) [execution of the instructions within the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=268.8049090909091) [function block, rather **executing a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=271.0143333333333)**[coroutine function returns something](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=272.5998888888888" \t "psplayer)**[**called a coroutine object. To execute the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=274.88328571428565)**[coroutine object, you need to wrap it in a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=278.3140769230769" \t "psplayer)**[**future object, and pass it to a running**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=280.7413076923077)**event\_loop.**

[In our example, we are not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=283.8258750000001) [explicitly wrapping the coroutine, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=286.9070000000001) [internally the event\_loops](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=289.217375) [run\_until\_complete method will notice that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=290.79050000000007) [it's receiving a coroutine object and not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=293.13541666666663) [a future object and will wrap it for us.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=294.8673333333332) [Now I must mention that this future object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=297.33521428571436) [is slightly different from the one we saw](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=299.7003571428573) [in the last module, but we'll discuss that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=301.69827272727275) [in more detail in a few moments. Right now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=303.9797272727273) [let's try to **visualize the state changes**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=306.8533846153846)[**and flow of control** when executing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=308.8952307692307) [coroutine object. When executing the code](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=311.81639999999993" \t "psplayer) [on the screen, the event\_loop starts](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=314.60963636363647) [running after the call to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=317.0450909090912) [run\_until\_complete. It then puts the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=318.5760909090909" \t "psplayer) [future that wraps the coroutine object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=319.9824545454545) [into the pending state and begins](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=322.41481818181813) [executing the coroutine code. When it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=325.2195454545453) [reaches the await instruction, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=328.13033333333345) [coroutine is suspended and control returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=330.48159999999996" \t "psplayer) [to the event\_loop. A second later, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=332.8611999999998) [event\_loop resumes execution of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=335.97460000000007" \t "psplayer) [coroutine. Once completed, the future is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=338.37910000000016" \t "psplayer) [set to done and the loop is stopped. So](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=340.8711538461538) [now that we've seen how a coroutine works](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=344.3790833333333) [within an event\_loop, let's talk about](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=346.6997499999998) [futures and explore other capabilities of coroutines.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=1&mode=live&start=350.57)

[More Asyncio Concepts](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live)

[The asyncio module provides a **Future**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=2.349) [**object** that like the concurrent.futures](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=4.616249999999999).[Future object can be used **to manage the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=6.3536666666666655) [**execution** of a function, and from which we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=8.317) [can **retrieve the results** once the function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=11.267000000000003) [is done executing. Some of the methods](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=13.942500000000003) [both future objects have in common are the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=16.743666666666673) [**cancel** method, which is used to cancel the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=19.02415384615384) [future, the **done** method, which returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=21.305083333333336) [true if the future is completed or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=24.629583333333343) [canceled, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=28.062799999999996" \t "psplayer) **[result](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=28.062799999999996" \t "psplayer)** [method, which returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=28.062799999999996" \t "psplayer) [the result that the future represents, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=30.051199999999987) [**exception** method, which returns any](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=33.589000000000006) [exception that happened during the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=36.464) [execution of the future, and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=38.21066666666667) **[add\_done\_callback](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=40.29583333333333" \t "psplayer)** [method, which adds a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=40.29583333333333" \t "psplayer) [callback to be run when the future becomes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=42.01541666666665" \t "psplayer) done.

[The major **difference between the two**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=46.45525000000001)[**future classes** is in the **blocking behavior**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=48.49799999999998) [of the results and exception methods. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=50.320142857142855) [concurrent.futures.Future object is meant](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=54.79536363636363" \t "psplayer) [to be used with traditional concurrency,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=56.94554545454544) [where calling result on a future object](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=59.4192) [will block the calling thread until the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=63.05989999999999) [result is available. But in the world of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=65.13799999999998) [asyncio, almost all execution is handled](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=67.5293076923077" \t "psplayer) [within a single thread, therefore blocking](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=69.46407692307693) [is highly undesirable. Therefore calling](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=73.16100000000002) [the results method on the **asyncio future**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=76.48861538461539)[**object does not block**. It gets the result](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=80.12376923076923) [and returns immediately. **If the result is**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=83.8392857142857) [**not yet available**, because the computation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=86.09846153846156) [is not yet complete, then the method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=87.64415384615391) [**raises an exception**. The **same** goes **for the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=90.54499999999999) [**exception method**, which returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=94.45554545454547) [immediately or raises an exception if the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=96.30718181818185) [future is not complete. If you're **inside a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=99.20244444444448) **[coroutine, the right way to wait for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=101.90076923076923" \t "psplayer)** [an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=101.90076923076923" \t "psplayer) [asyncio future to complete is to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=104.14630769230767" \t "psplayer) **[await](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=104.14630769230767" \t "psplayer)** [the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=104.14630769230767" \t "psplayer) [future. The event\_loop will pause](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=106.67924999999997) [execution at this point and watch for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=108.23033333333333) [future object to be set to done before](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=110.689) [resuming execution. If you're **outside a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=115.86019999999999) **[coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=118.2773076923077" \t "psplayer)**[, then you can pass the future to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=118.2773076923077" \t "psplayer) [an event\_loop using the **run\_until\_complete**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=120.52592307692309) [method. In this case, the loop will watch](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=123.5742) [for the completion of the future and then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=125.89449999999997) stop after the future gets done.

[A **task** is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=128.1802857142857)[a **subclass of future** that is used to **wrap**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=132.607) [**and manage the execution of a coroutine** in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=134.056) [an event loop. A coroutine must be wrapped](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=137.17025) [in a task before it can be run on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=140.05431250000004) [event\_loop. The term task and future are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=142.66493750000012" \t "psplayer) [often used interchangeably in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=145.19039999999995) [documentation and sometimes in this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=146.7) [course, because task is a type of future.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=148.7226923076923) [However, when the term task is mentioned](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=151.959) [with reference to a Python type, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=155.01508333333342) [specifically refers to a class of future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=157.53984615384618) [that is designed to wrap a coroutine. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=158.7363076923078) **[ensure\_future method is used to wrap a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=163.1315" \t "psplayer) [coroutine in a task](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=165.94199999999998" \t "psplayer)**[. The ensure\_future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=165.94199999999998" \t "psplayer) [method takes in a coroutine or a future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=168.08509090909092) [object, and if the object passed in is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=170.2334545454546) [already a future object, it simply returns](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=172.8527272727273) [the object. But if the passed in object is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=174.9492307692308) [a coroutine, then it creates a future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=177.20892307692324) [object by calling creating\_task on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=180.0443636363636) [passed in event\_loop instance or the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=182.7918888888889) [default\_event\_loop instance. You can also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=186.3552222222223" \t "psplayer) [directly call **create\_task** on an event\_loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=188.5681818181818) [instance to create a task. In any case,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=190.61854545454543) [creating a task using either the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=193.47100000000003) [ensure\_future or the create\_task method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=195.64" \t "psplayer) [has the important side effect of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=198.16399999999993) [scheduling the task to run on an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=201.05188888888887) event\_loop.

[In addition to being able to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=203.59911111111103) [await future objects, coroutines can also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=206.70649999999992) [await on other coroutines. This is called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=209.45) [**chaining coroutines**. Coroutine chaining](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=213.38400000000007) [makes it easier to decompose a task into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=215.19016666666667) [reusable parts. Let's take a look at an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=217.9188333333333) [example. Here we have the perform\_task](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=221.94471428571427) [coroutine that needs to call two other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=224.25908333333334" \t "psplayer) [coroutines in order to complete its task.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=226.719" \t "psplayer) [When await subtask1 is run, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=230.524) [perform\_task coroutine is suspended, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=233.15500000000003" \t "psplayer) [the subtask1 coroutine is started. When](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=235.46377777777778) [subtask1 completes, the coroutine resumes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=238.7924444444445) [and continues running until the it has to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=240.75953846153848) [do the same thing with subtask2. So we see](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=243.0463076923078) [that coroutine chaining allows us to write](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=246.60524999999998) [clean, singularly-focused functions, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=249.87875) [compose them together as part of a larger](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=252.15824999999998) [task. It also allows you to await separate](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=255.1381538461539) [sections of your code as needed, which can be really valuable.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=2&mode=live&start=258.285)

[Parallel Execution of Tasks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live)

[If you have multiple coroutines or tasks,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=0.922) [and you want to wait for them all to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=3.028461538461538) [complete, you can use the **asyncio.wait**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=5.904727272727272) [function. This function is similar to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=8.40309090909091) [concurrent.futures wait function, but](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=10.6302" \t "psplayer) [it's designed to be used for tasks or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=12.10623076923077) [coroutines that are running on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=13.940076923076923" \t "psplayer) [event\_loop. Therefore it itself is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=15.867444444444445" \t "psplayer) [coroutine. It takes a list of tasks or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=18.89877777777778" \t "psplayer) [coroutines, and notifies the event\_loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=21.97416666666666" \t "psplayer) [when they're all complete, or when the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=23.81) [optional timeout expires. Like the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=26.565199999999994) [concurrent.futures wait function, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=28.576363636363634" \t "psplayer) [**return** value is **two sets of futures, the**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=30.652272727272724)[**done set and the pending set**. Also, we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=34.21499999999999) [specify when to return, either when **any of**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=37.994307692307686) [the tasks completes or raises an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=40.46446153846152) [exception, **or** when **all the tasks complete**.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=42.64818181818183) The default is set to the latter.

[Here's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=45.389) [an **example** of the asyncio wait method in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=48.59392307692308) [action. Here we have the get\_items](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=51.91309999999999) [coroutine, which executes four get\_item](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=54.545299999999976" \t "psplayer) [coroutine calls in a loop and waits for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=56.61299999999999" \t "psplayer) [them all to complete. Because we don't](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=59.871) [have a timeout and the return\_when is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=62.40383333333331) [defaulted to all\_completed, the coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=65.02255555555556) [will only resume when all the tasks are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=67.04644444444446) [completed, and the pending set will be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=69.9273846153846) [empty, so we can just ignore it. We know](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=72.344) [that all the tasks are in the completed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=75.77000000000002) [state, and we can call the result method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=78.75200000000001) [without any exceptions being thrown. If we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=81.54400000000004) [change the code a little bit, and set our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=83.84654545454549) [timeout to 2 seconds, then we'll have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=86.55157142857144) [couple of tasks that aren't completed by](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=88.63042857142858) [then. At this point, we can wait again for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=91.37984615384616) [the remaining tasks to complete or perform](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=95.812) [some other action as dictated by your](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=97.80572727272727) [business logic. But let's say what I want](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=99.572) [to do is cancel the unfinished tasks. I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=101.24400000000003) [can do so by simply calling the cancel](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=103.71099999999997) [method on each task in the pending set.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=106.655) [This ability to set a timeout and respond](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=110.10471428571432) [to tasks that aren't completed by the time](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=113.22533333333332) [the timeout expires is a benefit of using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=115.35866666666665) [the wait function that you won't get with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=117.51384615384617) [the gather function, which we'll discuss](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=119.763) in a few moments.

[If you want to have that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=121.905) [wait, but with a timeout functionality](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=125.18330769230766) [when you have only coroutine or task, then](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=127.03361538461533) [you can use the **wait\_for** function. You can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=130.88884615384615) [pass the function the task or coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=134.43885714285713) [that you want to await and the timeout.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=136.22335714285708) [You could set the timeout to none in order](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=138.71814285714285) [to simply wait indefinitely until the task](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=142.37857142857143) [completes, but then you might as well just](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=144.46450000000004) [directly await the task. With this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=146.856) [function, **if the timeout expires** before](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=149.9961818181818) [the task is completed, the **task gets**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=152.09672727272726) [**cancelled**, and an **asyncio.Timeout.Error** is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=155.4886666666667) [**raised**. If instead of waiting for all the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=159.5865833333334) [tasks to complete before reacting, you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=162.1629166666666) [want to yield the tasks as they complete,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=163.8907142857143) [you can use the **asyncio.as\_completed**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=166.3364285714286) [method. The as\_completed method also lets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=168.3749) [you specify an optional timeout value. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=171.43429999999992) [a timeout occurs before all the tasks are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=176.40223076923078) [done, then a timeout error is raised. With](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=178.33207692307704) [both the wait and the as\_completed](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=181.3904545454545) [methods, the return value is a collection](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=184.53063636363626) of futures.

[The **asyncio.gather** method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=188.177) [differs in this regard. Instead of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=191.3615) [returning a collection of futures, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=194.16500000000002) [gather method returns a single future that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=196.1450000000001) [aggregates the result from the past and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=198.89423076923072) [futures. The **results are aggregated** **in**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=201.8191538461537) [**order of the original sequence**, not](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=204.18799999999996) [necessarily in order of the results or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=206.45700000000002) [rival, which is highly beneficial in cases](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=208.6830000000001) [where a business logic requires it. When](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=211.73133333333334) [the ***return\_exceptions*** parameter is set to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=215.61512499999998) [False, the gather function immediately](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=217.6018749999999) [propagates the first raised exception to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=220.0338888888889) [the returned future. Otherwise exceptions](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=222.9340909090909) [in the tasks are treated the same as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=224.3045454545455) [successful results and gathered in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=227.42236363636363) [result list. It should be noted that the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=230.91709090909086) [gather function **has no timeout** parameter,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=233.83449999999993) [and because it returns a single future](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=235.587) [representing the collection of results,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=237.4827272727273) [you **cannot cancel tasks individually**. But](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=239.30883333333333) [if you don't care about these features,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=243.21892307692306) [then the gather function is a simpler](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=244.59038461538452) replacement for the wait function.

[We can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=246.8617142857143) [take our previous example and replace the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=250.57116666666667) [wait function with the gather function.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=252.25175) [And awaiting the future returned by the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=254.2519090909091) [gather function directly returns the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=257.23836363636366) [results, removing the need to call a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=258.8308181818182) [result on individual futures. Note that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=260.871) [I'm using the **splat operator** (\*) on the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=264.1557272727274) [item\_coros list to unpack the list into](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=267.7411818181818" \t "psplayer) [positional arguments for the gather](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=270.70981818181826) [function. Now that we've covered the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=272.76399999999995) [concepts involved in asyncio, let's use it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=275.1759999999999) [to do something useful, but first we'll need some library help.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=3&mode=live&start=277.82)

[Async IO Libraries](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live)

[When running coroutines, we want to ensure](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=1.503) [that we don't block. Instead of blocking,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=3.986727272727272) [we should yield control back to the event](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=6.176) [loop, so that it can do other tasks. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=8.229333333333335) [poses a challenge for most traditional](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=10.725999999999996) [Python libraries that perform IO. Most of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=12.8885) [these libraries are not designed to be run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=17.232230769230767) [an event\_loop and yield control, instead](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=19.357153846153835) [they block an IO. As a result, in order to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=21.27807692307692) [perform the functions that these libraries](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=25.025) [provide, we need to find alternative](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=26.98700000000001) [libraries that support single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=29.255857142857142) [asynchronous behavior. Unfortunately](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=31.79128571428571) [everyone's favorite HTTP client requests](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=33.478249999999996) [doesn't have support asyncio, however,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=35.574499999999986) [there's an alternative library that does,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=38.64233333333333) the aiohttp library.

[The **aiohttp** library](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=43.101) [is an HTTP server and client library that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=46.475500000000004) [was designed for asyncio. To install](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=50.09550000000001) [aiohttp, you need to run the command](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=52.60254545454547" \t "psplayer) *[pip](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=52.60254545454547" \t "psplayer)* [*install aiohttp*. Once you have the library](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=56.3724) [installed, creating a server is as easy as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=60.9378) [creating an instance of web.Application,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=63.904) [creating a coroutine that responds to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=66.45579999999997) [get requests, and then registering that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=69.57059999999998) [handler coroutine to be run whenever a get](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=72.39779999999996) [request is made to a particular path or](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=75.3367692307692) [paths. There is way more that can be done](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=77.98614285714285) [with the aiohttp library to **implement an**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=80.3897857142857) [**HTTP server**. In fact, the **aiohttp.web**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=83.79799999999997)[**module provides an entire web framework**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=88.36024999999992)[**that rivals Flask, NodeJS**, or any other](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=90.93588888888885) [web framework. But exploring this further](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=93.57149999999999) [is beyond the scope of this course. If](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=95.76449999999997) [you're interested in exploring this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=99.23918181818185) [further, the aiohttp **documentation has a**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=101.61736363636369) [**really good tutorial** that will help you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=103.62481818181821) [get going. What we're more interested in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=106.15599999999999) [is the client-side user of the aiohttp](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=109.7343076923077) [library. We can use aiohttp to **make HTTP**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=112.80353846153847) [**requests in a non-blocking**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=116.32900000000002)**[asyncio-compatible manner](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=118.093" \t "psplayer)** [like in this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=118.093" \t "psplayer) [example. In the coroutine called main, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=122.31133333333332) [created an instance of ClientSession using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=125.85445454545453) [an asynchronous context manager. A](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=128.74477777777778) [**asynchronous context manager** is a context](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=131.94866666666664) [manager that is able to yield control](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=134.6932307692308) [within its enter and exit methods. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=136.61176923076934) [client session object is the main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=139.15733333333336) [interface for making HTTP requests, and it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=141.7423333333334) [has coroutine methods like get, post, put,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=145.4926363636363) [and delete that correspond to the standard](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=149.5524) [HTTP methods. After creating our session](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=153.70620000000002) [object, we called the fetch coroutine,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=156.33019999999996) [passing in the session object and the URL](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=158.486) [to get. In our fetch coroutine, we invoke](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=162.18359999999996) [session.get, passing in the URL and use](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=165.34323076923073) [another asynchronous context manager to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=168.5905) [manage the response object. The response](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=171.78799999999998) [to text method is also a coroutine. So we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=175.1686666666667) [await its completion, and then return the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=178.65061538461538) [results to the main coroutine, which](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=180.7597692307692) [prints it. We'll put this into practice](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=183.1512) [soon, so that we get to truly understand](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=187.30289999999994) [what goes on, but first let's get a look](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=188.91986666666662) [at another library that supports asyncio,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=190.57185714285714) aiofiles.

[The **aiofiles** module is an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=193.463) [library that provides an asyncio-enabled](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=197.11459999999997) [alternative to Python's standard file API.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=200.3787142857143) [The API is very similar to the standard](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=205.357) [file API, but with support for async and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=207.23928571428567) [await keywords and constructs. Here's an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=210.197) [example of a standard blocking file read](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=213.70190909090906) [versus a non-blocking asynchronous file](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=215.1470909090908) [read. As we can see, the aiofiles version](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=217.61837500000001) [allows you to use it as an asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=221.9915) [context manager, and the read method is a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=224.50483333333332) [coroutine that is non-blocking and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=227.81661538461535" \t "psplayer) [awaitable. But aside from that, the API](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=229.95776923076917" \t "psplayer) [still looks very familiar because of its](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=232.1555384615385) [similarity to its blocking counterpart.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=234.22) [Just to illustrate that further, here's an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=236.91999999999996) [example of a blocking file write using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=240.16200000000003) [standard file API versus the asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=243.04733333333343) [file API. Installing the aiofiles library](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=246.4501428571428) can also be done using pip.

[And now with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=248.957) [these two libraries, aiohttp and aiofiles,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=252.9529999999999) [we can implement an asyncio version of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=255.96854545454542) [download image in our thumbnail\_maker](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=259.8968181818181) [example (**script5\_asyncio**). Let's switch over to the command](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=264.1739999999999) [prompt or a Git GUI if that's what you](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=266.11828571428583) [prefer, and then we can check out a new](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=267.9325333333333) [branch, which I'll call asyncio. This way](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=272.0773333333333) [we can safely experiment with asyncio and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=275.13969230769226) [return to our main branch if we need to.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=277.7986153846152) [For this project, I'm going to install the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=280.5365714285715) [two libraries I mentioned earlier, so I'll](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=282.48628571428594) [add them to the requirements.txt file,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=284.6608) [and then run pip or pip3 install -r](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=289.10215384615384) [requirements.txt. Now I'm ready to code.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=297.5132307692308)

[First off, I'll import asyncio, aiohttp,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=304.0225384615385) [and aiofiles. Now I need to turn my](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=309.3103846153847) **[download\_image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=312.98900000000015" \t "psplayer)** [method into a coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=312.98900000000015" \t "psplayer) [that takes in an aiohttp.ClientSession](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=314.58566666666695) [object and a URL, and downloads the image](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=318.1189285714285) [asynchronously. So I'll add the async](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=323.2612142857142) [keyword and post-fix the method name with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=326.05207692307687) [coro just to make it easier to identify.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=329.0593846153845" \t "psplayer) [I'll also change the passed-in inputs. Now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=333.72292307692305) [we still need the img\_filename and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=337.952) [img\_filepath definitions, as well as the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=341.55200000000013" \t "psplayer) [putting of the img\_filename into the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=344.18923076923085) [img\_queue once the download is completed,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=346.3450769230771" \t "psplayer) [but most of this code can go. For the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=348.6242) [dl\_size calculation, because we're now in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=352.2175" \t "psplayer) [a position where we never modify the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=354.5889999999998) [dl\_size variable from another thread, I](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=357.2305" \t "psplayer) [don't need to have a lock around it. As an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=359.3515) [aside, if I was accessing a shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=364.3415000000001) [resource that I needed to lock, I would](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=368.271) [want to use an asyncio lock instead of a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=370.6069999999998) [threading lock. This is because a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=373.47925000000004) [threading lock blocks the running thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=375.71975000000015) [when it can't acquire the lock, whereas](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=377.4122499999999) [the asyncio lock let's you yield from it.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=378.7474999999998) [To do the download, we need to call](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=382.2048571428571) [session.get using an asynchronous context](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=386.306" \t "psplayer) [manager to manage the response. Now when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=388.9034545454546) [the response arrives, we want to write it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=391.5928333333334) [to the img\_filepath, and to do so](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=393.69216666666694) [asynchronously, we need to use the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=396.0374166666666) [aiofiles API in place of the standard file](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=398.2219166666666" \t "psplayer) [api. So I'm going to open the file in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=400.6570666666666" \t "psplayer) [binary write mode using another async with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=403.73355555555554) [block. And then I'll wait for the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=408.7004444444443) [response. contents and write it to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=412.47520000000014) [file asynchronously. We now have our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=416.05126666666695) [download file coroutine that downloads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=419.067909090909) each image.

[Let's create another](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=421.56154545454524) [coroutine. I'll call it the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=424.09000000000003" \t "psplayer) **[download\_images\_coro](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=426.6416666666667" \t "psplayer)** [that takes in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=426.6416666666667" \t "psplayer) [img\_url list and calls the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=429.9840000000001" \t "psplayer) [download\_image\_coroutine. All we're doing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=433.5240000000002" \t "psplayer) [in here is creating the ClientSession](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=435.04725) [object with an asynchronous context](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=436.67775) [manager, and then we loop through the URL](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=439.1128181818182) [list, and call the downloads\_image\_coro](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=445.59209090909116) [coroutine, passing in the session we just](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=447.67479999999995" \t "psplayer) [created and the URL. We'll use the await](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=450.30978571428574) [keyword to chain the coroutine. Finally,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=456.1160714285716) [in our **download\_images** method, in place of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=460.1586) [looping through the images and doing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=463.0537999999998) [download, we'll create an event\_loop and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=466.75933333333336) [pass the download\_images coroutine to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=470.2133333333334) [run\_until\_complete method. We should wrap](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=475.31542857142847" \t "psplayer) [this in a try final block, and close the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=478.6268) [event loop afterwards. This will execute](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=483.1292000000001) [the coroutine function, wrap the coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=485.73154545454537) [object in a task, and wait for the task to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=487.9346363636361) complete.

[Now let's go do some cleanup in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=493.051222222222) [our make\_thumbnails method. We can remove](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=497.15779999999995) [all the queuing code, and simply call the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=499.8104545454546) [download\_images method. The problem here](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=502.817" \t "psplayer) [now is that with the old code, we had](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=505.7001818181817) [processes reading items off the queue as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=510.0865) [they were produced, but here the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=512.2483333333333) [download\_images method completes the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=515.1837777777778" \t "psplayer) [downloads before the img\_resizing starts.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=517.5015555555556) [So to fix this, I'll move the invocation](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=519.5799090909089) [of the download\_images method until after](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=522.5626666666667) [the processes have been kicked off. This](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=524.7206666666671) [way, as soon as an image is downloaded,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=527.4169230769231) [one of the processes can pick it up off](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=530.58) [the queue and begin resizing. I'll also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=532.9495714285719) delete this queue.join here.

[Now we're](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=535.5294666666667) [ready to run this using pytest. So now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=539.9516000000004) [what we have is that we're performing the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=544.2350769230768) [IO-bound task of downloading the files](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=546.2073846153842) [asynchronously using a single thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=548.2172) [instead of a pool of threads running in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=550.3342000000001) [parallel. After each download is done, it](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=553.6604999999997) [gets written to a queue, what is a pool of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=556.3558571428571) [processes waiting to retrieve the items](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=558.5144285714291) [and perform the CPU-bound task of resizing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=560.6225000000002) [the images. Even though our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=564.0303333333338) [single-threaded download in this instance](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=565.6739) [is much slower than our pool of threads,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=566.7904) [it still out performs synchronously](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=569.8906363636366) [downloading the image by a factor of 2x on](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=572.102) [this machine. More importantly, **it's less**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=575.4622727272723)[**resource intensive than thread pooling and**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=577.495)[**can better to scale to large numbers of**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=579.424)[**downloads**, like downloads in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=582.7011428571432) thousands.

[If you want to speed this up](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=584.7493571428577) [even further, it's possible to run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=586.5853846153847) [multiple threads, each with its own event](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=587.808) [loop, performing single-threaded asyncio,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=590.5371249999997) [but that's a more advanced topic that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=592.3693333333334) [won't be covered in this course. So now](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=595.8764285714286) [we've seen how aiohttp and aiofiles](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=600.5566923076923) [enables us to make HTTP requests and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=603.3937692307692) [perform file IO in an asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=606.7906666666668) [fashion. But what about other IO-heavy](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=609.888666666667) [functions. If we need to, for instance,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=612.2583333333334) [perform a MySQL query, are we forced to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=614.853) [use a blocking MySQL library? Well, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=617.9453076923081) [don't have to. There's an asyncio-enabled](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=621.015) [MySQL library called **aiomysql**, as well as](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=624.4259999999998) [an **aiopg** for postgres, an **aiocouchdb** for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=628.9755) [Couch DB, and an **aiocassandra** for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=633.753) [Cassandra, and it's not just databases.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=637.4405000000003) [There's a growing list of libraries and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=641.128) [frameworks that support asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=644.110583333333) [operation. The wiki page of the asyncio](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=645.8149166666661) [GitHub titled ThirdParty has a pretty](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=649.3562857142856) [comprehensive list (page 128). You should check this](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=652.4715714285712) [out if you need an alternative library for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=654.3616923076922) [a task, but if there isn't an alternative](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=656.0650769230767) [library available for what you're trying](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=658.6121818181816) [to achieve, all is not lost. It is still](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=660.93425) [possible to use those libraries in an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=664.8453846153848) [application based on asyncio by using an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=667.0536153846159) [executor from concurrent.futures to run](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=669.4646666666667) [the code in either a separate thread or process.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=4&mode=live&start=671.6616666666669)

[Combining Coroutines with Threads and Processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live)

[Blocking in a coroutine function can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=1.735) [very harmful to the system. So **if we need**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=4.32675)[**to call a blocking function** or library,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=6.949272727272727)[instead of calling it directly, we should](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=8.299) [delegate that call to an executor. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=11.304153846153842) [AbstractEventLoop class provides a method](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=14.224454545454547" \t "psplayer) [for doing so in an asyncio-friendly way](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=15.946727272727278) [called **run\_in\_executor**. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=18.082) [run\_in\_executor method is a coroutine,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=20.313249999999996" \t "psplayer) [therefore it can run on the event\_loop](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=24.032) [without blocking because it can be](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=26.46700000000001) [awaited. The run\_in\_executor method runs](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=28.554142857142875) [the passed-in blocking function in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=30.809600000000003) [passed\_in\_executor. If the executor is set](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=33.671" \t "psplayer) [to none, then by default a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=36.214500000000015) [ThreadPoolExecutor is used with the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=38.758" \t "psplayer) [default number of threads, which varies](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=40.70244444444443) [based on the number on the number of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=42.65066666666667) [processor cores available on the machine.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=44.221333333333355) [The args parameter is a sequence of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=46.67981818181817) [positional arguments that gets passed in a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=50.295) [function. If you need to use keyword args](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=52.873625000000004) [instead, you can use the functools.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=55.45615384615387)[partial function to create a callable to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=57.354) pass it a run\_in\_executor.

[Here's an](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=60.55363636363636) [example of run\_in\_executor in action using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=63.4854) [a ThreadPoolExecutor. Here we have a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=66.79559999999998) [blocking function aptly named](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=69.55136363636365) [blocking\_function that we would like to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=71.20372727272729" \t "psplayer) [call from within our coroutine. In the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=73.78025000000001) [\_main block, we create our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=76.94000000000001) [ThreadPoolExecutor instance with three](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=79.26750000000004" \t "psplayer) [worker threads and our event\_loop, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=80.97990909090909) [then run the \_main coroutine in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=82.87536363636362) [event\_loop, passing in the loop object and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=85.82084615384616" \t "psplayer) [the executor instance. In the \_main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=89.74515384615388) [coroutine, we create six tasks that call](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=92.56675000000001" \t "psplayer) [the blocking function, and then we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=95.22616666666673) [await the completion of those tasks.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=97.87372727272728) [Because we are calling the function using](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=100.139) [run\_in\_executor, the event\_loop thread](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=102.63187500000004" \t "psplayer) [never gets blocked, as the execution gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=104.09859999999999) [delegated to the worker threads. And we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=106.69) [can have an asyncio future that gets](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=109.49408333333331) [returned, which the event\_loop can watch](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=112.12663636363636) [for the completion of the tasks. Once the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=114.01554545454546) [tasks are complete, we print the results,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=117.5770833333333) and now the loop can shut down.

[The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=121.241) [ProcessPoolExecutor works the same way.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=123.69533333333334" \t "psplayer) [Here's an example where we use a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=125.947) [ProcessPoolExecutor to run a single](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=128.84900000000005" \t "psplayer) [blocking function. Here the blocking](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=130.92185714285722) [function we're trying to call is the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=132.81425000000002) [factorial function. We create our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=135.29516666666677) [ProcessPoolExecutor instance, our](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=137.78042857142853" \t "psplayer) [event\_loop, and we specify the number](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=139.55685714285707" \t "psplayer) [whose factorial we want to calculate. Like](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=142.38763636363632) [before, we pass all this to the \_main](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=145.3372142857143) [coroutine, and in the \_main coroutine, we](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=147.4429285714287" \t "psplayer) [perform the computation using the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=149.91255555555554) [run\_in\_executor method. This delegates the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=152.79644444444443" \t "psplayer) [function call to the executor instance,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=155.25466666666662) [and gives us a future object that we can](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=157.864) [await. When the future is done, execution](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=160.62430000000006) [of the \_main coroutine resumes. We print](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=162.97263636363633) [the message and exit the coroutine,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=165.7263076923077) [allowing the loop to close. SO we see that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=168.88323076923078) [we can use the run\_in\_executor method to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=172.65600000000006) [allow us to run normally-blocking code,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=174.65450000000016) [which can include not just our own code,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=176.70036363636368) [but also external libraries inside of an asyncio event\_loop.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=5&mode=live&start=179.872)

[Concurrency in Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live)

[That concludes the asyncio topics we want](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=2.503) [to cover in this course. While there are a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=4.593666666666667) [few **things we didn't examine like asyncio**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=7.285545454545453)[**synchronization primitives and the stream**](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=9.383)[**reader and writer APIs**, what we did](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=11.519666666666666) [accomplish is that we learned what](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=14.2025) [single-threaded asynchrony is and why it's](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=16.149500000000003) [beneficial for doing highly-scalable IO](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=18.70672727272727) [work. We also learned about how event](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=20.93945454545454) [loops and coroutines interacts, with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=23.702599999999993) [coroutines yielding control back to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=25.568846153846152" \t "psplayer) [event loop when it has to wait for a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=27.499923076923068) [long-running operation to complete, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=30.278090909090913) [event loop knowing to resume the coroutine](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=31.6899090909091) [after the operation is completed. We also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=34.51611111111111) [talked about the terms future and task in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=37.72092307692308) [the context of asyncio and examined](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=40.644615384615385) [coroutine chaining. We then looked at](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=42.6959" \t "psplayer) [three different ways to execute IO tasks](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=44.605700000000006) [in parallel on the event loop, and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=47.49599999999999) [aggregate the results of the tasks. We](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=50.41225) [worked with the aiohttp and aiofiles](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=52.82200000000002) [libraries, two of the many libraries that](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=54.945800000000006) [provide awaitable APIs for performing](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=57.55960000000003) [IO-bound tasks. There's growing support](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=61.060333333333325) [for the asyncio constructs amongst Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=63.934999999999995) [libraries. But in cases of blocking a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=66.55699999999999) [library or some other blocking function](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=68.553) [that we need to call, we looked at the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=70.112) [run\_in\_executor function for delegating](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=72.83824999999995" \t "psplayer) the call to an executor instance.

[Python](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=74.29785714285714) [is often thought of as a single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=77.8581111111111) [language, but there are several avenues](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=79.64388888888884) [for executing tasks concurrently. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=82.89788888888887) [threading module allows us to spin up](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=85.57263636363636) [native operating system threads to execute](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=87.50209090909092) [multiple tasks concurrently. The threading](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=90.46549999999999) [API allows us to create thread objects,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=93.09636363636363) [start them, and join on them. It also](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=94.82663636363634) [provides several synchronization and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=98.27899999999997) [inter-thread communication mechanisms for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=100.251) [when threads need to communicate and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=101.65099999999998) [coordinate with each other, or for when](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=103.9834) [multiple threads are mutating the same](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=107.24680000000001) [area of memory. But the current](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=109.79612499999998) [implementation of CPyton has a GIL in](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=112.1243846153846) [order to make Python easier to implement](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=114.64061538461532) [and faster to run for single-threaded](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=117.28081818181818) [programs. And as a result of the GIL,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=119.62354545454545) [threading is not suitable for CPU-bound](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=122.62049999999998) [tasks. So instead, we have the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=124.85464285714282) [multiprocessing package. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=127.88128571428572) [multiprocessing package uses processes](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=129.3529) [instead of threads as the actors of](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=130.99249999999998) [parallel execution. And the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=134.05640000000002) [multiprocessing API tries to mimic the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=136.08520000000007) [threading API as much as possible in order](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=138.64830769230767) [to reduce the amount of distance between](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=140.78553846153832) [the two, and to make switching easier.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=142.98299999999995) [This is on display right down to the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=145.426) [implementation of equivalent](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=148.69145454545452) [synchronization primitives. One of the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=149.916) [major areas where there is a difference](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=151.701) [between the two actors is in the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=154.11908333333332) [implementation of shared state. Threads](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=156.05166666666665) [automatically share memory with each](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=157.6748) [other, but processes don't. So special](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=160.2688) [accommodations must be made to allow for](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=162.9203333333333) [processes to share state, namely OS-shared](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=164.9374999999999) [memory areas and manager processes. The](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=168.059125) [concurrent.futures module provides a](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=172.2207" \t "psplayer) [layer of abstraction over both concurrency](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=174.21919999999997) [mechanisms and also introduces us to](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=176.59088888888888) [futures, which represent pending results](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=178.9095555555555) [and also allows us to manage the execution](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=180.95963636363632) [of computations. And lastly, the asyncio](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=184.458) [module brings single-threaded asynchronous](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=187.84449999999998) [programming into Python as a native](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=190.057) [first-class citizen of the language with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=192.08499999999995) [new constructs such as the async and await](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=194.31936363636362) [keywords, asynchronous context managers,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=197.84881818181805) and much more.

[If you would like to go](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=200.2295) [learn more about Python concurrency, the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=204.18663636363638) [official Python documentation and the](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=206.14463636363635) [Python Module of the Week blob, which is](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=207.52781818181813) [available in book form as a book called](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=209.96121428571428) [The Python 3 Standard Library by Example,](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=212.76178571428568) [both have sections on concurrency and are](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=215.58327272727274) [great resources on the available APIs and](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=218.683) [how to use them. Aside from that, there](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=221.6815454545454) [are many books, youtube videos and blog](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=224.5780769230769) [posts that you can dig into with](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=226.9442307692307) [confidence now that you've gotten started](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=228.5) [with Python concurrency. My name's Tim Ojo, and thanks for watching.](https://app.pluralsight.com/player?course=python-concurrency-getting-started&author=tim-ojo&name=python-concurrency-getting-started-m5&clip=6&mode=live&start=231.60666666666668)