The Pool Class

The Pool class is used to represent a pool of worker processes. It has methods which can allow you to offload tasks to the worker processes. Let’s look at a really simple example:

```
from multiprocessing import Pool


def doubler(number):
    return number * 2

if __name__ == '__main__':
    numbers = [5, 10, 20]
    pool = Pool(processes=3)
    print(pool.map(doubler, numbers))
```

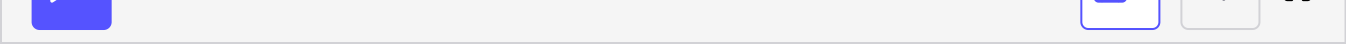
Basically what’s happening here is that we create an instance of Pool and tell it to create three worker processes. Then we use the **map** method to map a function and an iterable to each process. Finally we print the result, which in this case is actually a list: **[10, 20, 40]**.

You can also get the result of your process in a pool by using the **apply_async** method:

```
from multiprocessing import Pool


def doubler(number):
    return number * 2

if __name__ == '__main__':
    pool = Pool(processes=3)
    result = pool.apply_async(doubler, (25,))
    print(result.get(timeout=1))
```



What this allows us to do is actually ask for the result of the process. That is what the **get** function is all about. It tries to get our result. You will note that we also have a timeout set just in case something happened to the function we were calling. We don’t want it to block indefinitely after all.