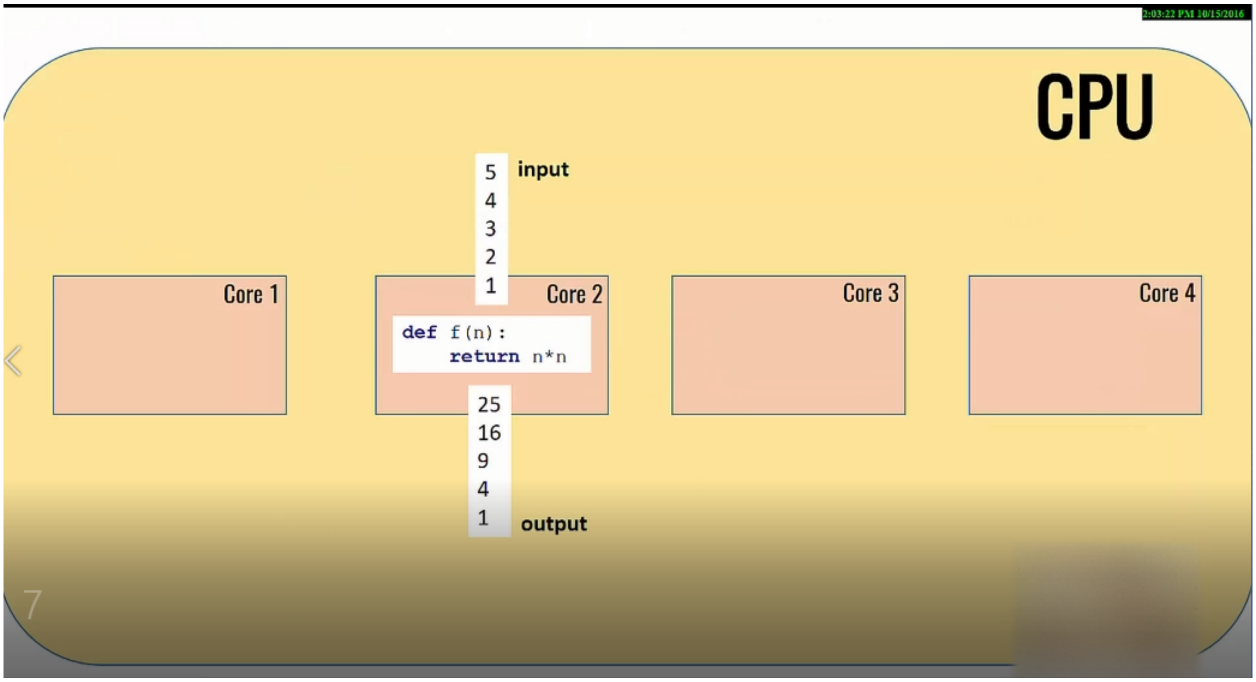
# Video 7

### Multiprocessing Pool

so in bellow I have a sample code

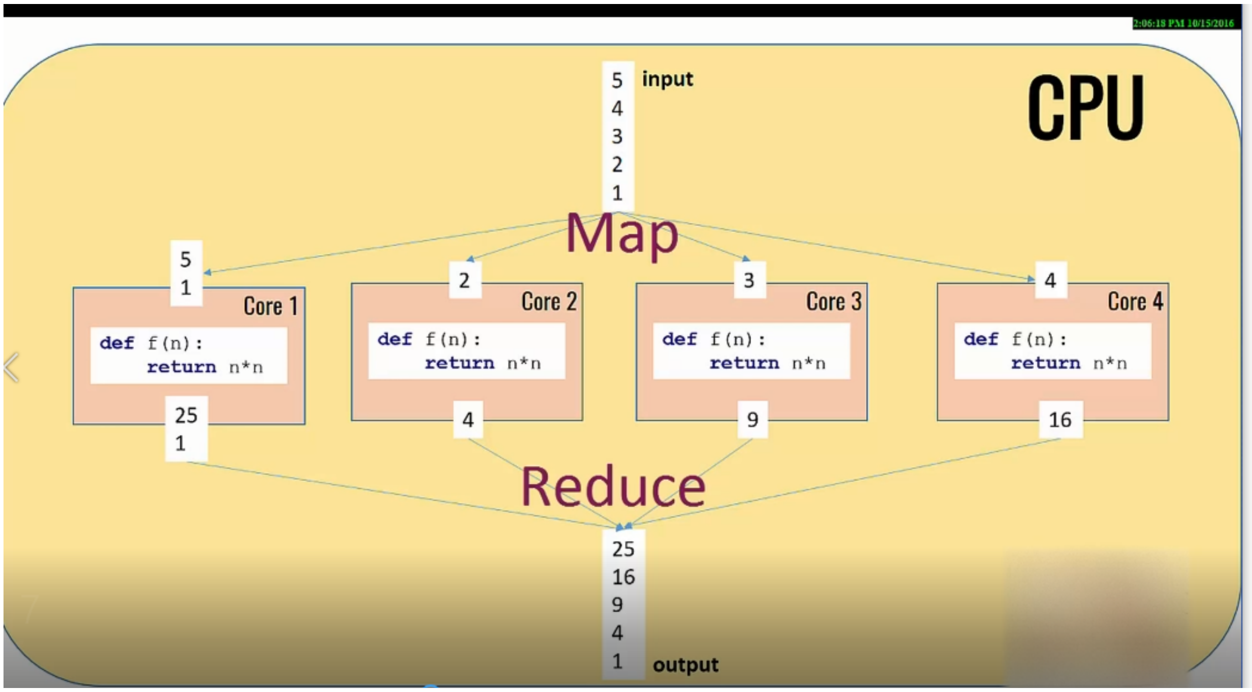
```
# Multiprocessing Pool
def f(n):
    return n*n
if __name__ == "__main__":
    array = [1,2,3,4,5]
    result = []
    for n in array:
        result.append(f(n))
    print(result)
```

so let’s look at details of how Python execute this script



```
def f(n):
    return n*n
```

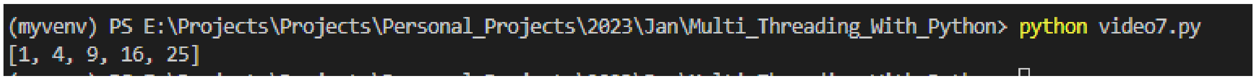
here the big yellow is your CPU and the rectangles are the cores, and when you execute the code your OS will run the code on one of the cores and the other cores will sit either,



```
def f(n):
    return n*n
```

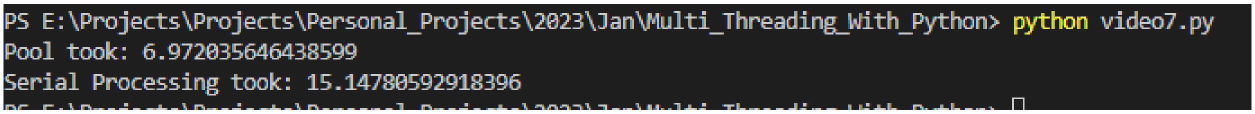
so the above method is called parallel processing, the process of dividing input between multiple cores is called **map**, and the process of aggregating the results together is called **Reduce**

```
# Multiprocessing Pool
from multiprocessing import Pool
def f(n):
    return n*n
if __name__ == "__main__":
    array = [1,2,3,4,5]
    p = Pool()
    result = p.map(f,array)
    print(result)
```



```
(myvenv) PS E:\Projects\Projects\Personal_Projects\2023\Jan\Multi_Threading_With_Python> python video7.py
[1, 4, 9, 16, 25]
```

```
# Multiprocessing Pool
from multiprocessing import Pool
import time
def f(n):
    sum = 0
    for x in range(1000):
        sum += x*x
    return sum
if __name__ == "__main__":
    t1 = time.time()
    p = Pool(processes=10) # processes limits number of processes
    result = p.map(f,range(100000))
    p.close()
    p.join()
    print("Pool took:",time.time()-t1)
    result = []
    t2 = time.time()
    for x in range(100000):
        result.append(f(x))
    print("Serial Processing took:",time.time()-t2)
```



```
PS E:\Projects\Projects\Personal_Projects\2023\Jan\Multi_Threading_With_Python> python video7.py
Pool took: 6.972035646438599
Serial Processing took: 15.14780592918396
```