**Multiprocessing**

1. runs in difference memory space
2. CPU bound operations
3. Takes advantage of multiple CPUs & cores
4. Avoids GIL limitations
5. Child processes are interruptible/killable

**Multithreading**

1. same memory space
2. cannot work in python due to GIL
3. I/O bound tasks such as waiting for API RESPONSES
4. Not interruptible/killable

* GIL only ensures that only one cpython bytecode instruction will run at any given time. At a time only one thread controls the interpretor. When the interpreter is executing a bytecode, it serializes execution by acquiring the Global Interpreter Lock. This means that two threads cannot be executing bytecode concurrently on two different cores.
* In Multiprocessing each process has its own interpretor no shared memory