Python `multiprocessing` Cheat Sheet

---

1. Process

- **Create process:**

```
from multiprocessing import Process
p = Process(target=function, args=(args,))
p.start()
p.join()
```

- **Properties & Methods:**
- `start()`, `join(timeout=None)`, `terminate()`, `is_alive()`
- `daemon` (True/False), `pid`, `name`

2. Pool

- **Create pool:**

```
from multiprocessing import Pool
with Pool(processes=4) as pool:
    results = pool.map(func, iterable)
```

- **Key Methods:**
- `map()`, `map_async()`, `apply()`, `apply_async()`, `starmap()`, `starmap_async()`
- `close()`, `terminate()`, `join()`

3. Queue

- **Create queue:**

```
from multiprocessing import Queue
q = Queue()
q.put(obj)
q.get()
```

- **Properties:** `empty()`, `full()`, `qsize()`

4. Pipe

- **Create pipe:**

```
from multiprocessing import Pipe
parent_conn, child_conn = Pipe()
parent_conn.send(obj)
child_conn.recv()
```

5. Shared memory

- **Value:** `val = Value('i', 0)`
- **Array:** `arr = Array('i', [1,2,3])`
- **Access:** `val.value`, `arr[:]`

6. Lock

- **Prevent race conditions:**

```
from multiprocessing import Lock
lock = Lock()
with lock:
    # critical section
```

- Methods: `acquire()`, `release()`

7. Semaphore

- **Limit concurrent access:**

```
from multiprocessing import Semaphore
sem = Semaphore(2)
sem.acquire()
sem.release()
```

8. Manager

- **Share complex objects:**

```
from multiprocessing import Manager
manager = Manager()
shared_list = manager.list()
shared_dict = manager.dict()
```

9. Current process info

```python
from multiprocessing import current_process
current_process().name
current_process().pid
```

10. Best Practices

1. Use `if __name__ == '__main__':` for Windows/macOS.
2. CPU-bound → `multiprocessing`, IO-bound → `threading`.
3. Shared memory (`Value`, `Array`) vs message passing (`Queue`, `Pipe`).
4. Avoid global mutable state.
5. Use `daemon=True` for background processes.
6. Context options: `get_context('spawn'|'fork'|'forkserver')`

Summary Table: | Feature | Key Methods | Notes | |---------|------------|------| | Process | start(), join(), terminate(), is_alive() | Each process has own memory | | Pool | map(), apply(), starmap(), close(), terminate() | Manage multiple worker processes | | Queue | put(), get(), empty(), full() | Process-safe message passing | | Pipe | send(), recv() | Bidirectional communication | | Value/Array | value, array[:] | Shared memory primitives | | Lock | acquire(), release() | Avoid race conditions | | Semaphore | acquire(), release() | Limit access to resources | | Manager | list(), dict(), Namespace() | Shared complex objects |

This cheat sheet contains all essential `multiprocessing` methods, properties, and best practices for practical use.