Goroutines

- A goroutine is a lightweight thread of execution; **goroutines are key ingredients to achieve concurrency in Go.**
- A goroutine is a function that is capable of running concurrently with other functions. To create a goroutine we use the keyword go followed by a function invocation;
- Goroutines are far smaller that threads, they typically take around 2kB of stack space to initialize compared to a thread which takes a fixed size of 1-2Mb.
- An OS Thread Stack is fixed size but a goroutine stack size shrinks and grows as needed.
- **Scheduling a goroutine is much cheaper than scheduling a thread.**
- OS threads are scheduled by the OS kernel, but goroutines are scheduled by its own Go Scheduler using a technique called m:n scheduling, because it multiplexes (or schedules) **m goroutines** on **n OS threads**.
- Goroutines have no identity. There is no notion of identity that is accessible to the programmer.