

What Is Concurrency?

Tasks Execute Sequentially

```
fmt.Println("Welcome!")
```

```
file, err := os.Create("file.txt")
```

```
file.WriteString("Test")
```

```
fmt.Println("Done!")
```

Tasks Execute Concurrently

```
fmt.Println("Welcome!")
```

```
os.Create("file.txt")
```

```
file.WriteString("Test")
```

```
fmt.Println("Done!")
```

Goroutine 1

Goroutine 2

Goroutines are functions that can execute in parallel (Go does not wait for their completion)

What Is Concurrency?

Code Execution – Sequentially

`fmt.Println(...)`

`os.Create(...)`

`file.WriteString(...)`

...

Code Execution – Concurrently (In Parallel)

`fmt.Println(...)`

`os.Create(...)`

`file.WriteString(...)`

...