

# Defer

**01**

About

**02**

Example

# Defer

- | Useful to run operations after functions complete
- | The `defer` keyword can be used to execute code **after** a function runs
  - | Clean up resources, reset data, etc

# How It Works

```
func one() {  
    fmt.Println("1")  
}  
func two() {  
    fmt.Println("2")  
}  
func three() {  
    fmt.Println("3")  
}
```

```
func sample() {  
    fmt.Println("Begin")  
  
    defer one()  
    defer two()  
    defer three()  
  
    fmt.Println("End")  
}
```

```
> go run ./lecture.go  
Begin  
End  
3  
2  
1
```

# Example

```
file, err := os.Open("sample.txt")
if err != nil {
    fmt.Println(err)
    return
}

defer file.Close()

buffer := make([]byte, 0, 30)
bytes, err := file.Read(buffer)
if err != nil {
    fmt.Println(err)
    // File will close with deferred call
    return
}
fmt.Printf("%c\n", bytes)
// File will close with deferred call
```