GO on the toilet

episode #2

<u>There is no *finally*. In go you *defer* stuff.</u>

If you want to make sure that something will happen, in go you can use defer. Statement marked with defer will be called at the end of enclosing function.

Examples from Docker project:

```
defer os.RemoveAll(tmp) // inside test as clean-up step

defer ls.mountL.Unlock() // make sure to release acquired lock

defer f.Close() // close file after reading content

defer body.Close() // close HTTP response body after reading
content

defer d.Stop() // stop a daemon
```

Very often defer is used at the beginning of the function, which is very different from usage of finally in other languages (again example from Docker):

```
func (ps *Store) GetV2Plugin(refOrID string) (*v2.Plugin,
error) {
  ps.RLock()
  defer ps.RUnlock()
  …
}
```

Since `Runlcok()` is deferred, it will for sure happen at the end.