

Dealing with nutt

why Kotlin doesn't need Optimal

```
fun censorToEmpty(s: String?, censor: (String) -> String?) : String =  
    Optional.ofNullable(s).flatMap { Optional.ofNullable(censor(it)) }.orElse("")
```

```
fun censorToEmpty2(s: String?, censor: (String) -> String?): String =  
    s?.let(censor) ?: ""
```



scope function

Dealing with null

why Kotlin doesn't need Optional

```
fun censorToEmpty(s: String?, censor: (String) -> String?): String =  
    Optional.ofNullable(s).flatMap { Optional.ofNullable(censor(it)) }.orElse("")
```

```
fun censorToEmpty2(s: String?, censor: (String) -> String?): String =  
    s?.let(censor) ?: ""
```



scope function

