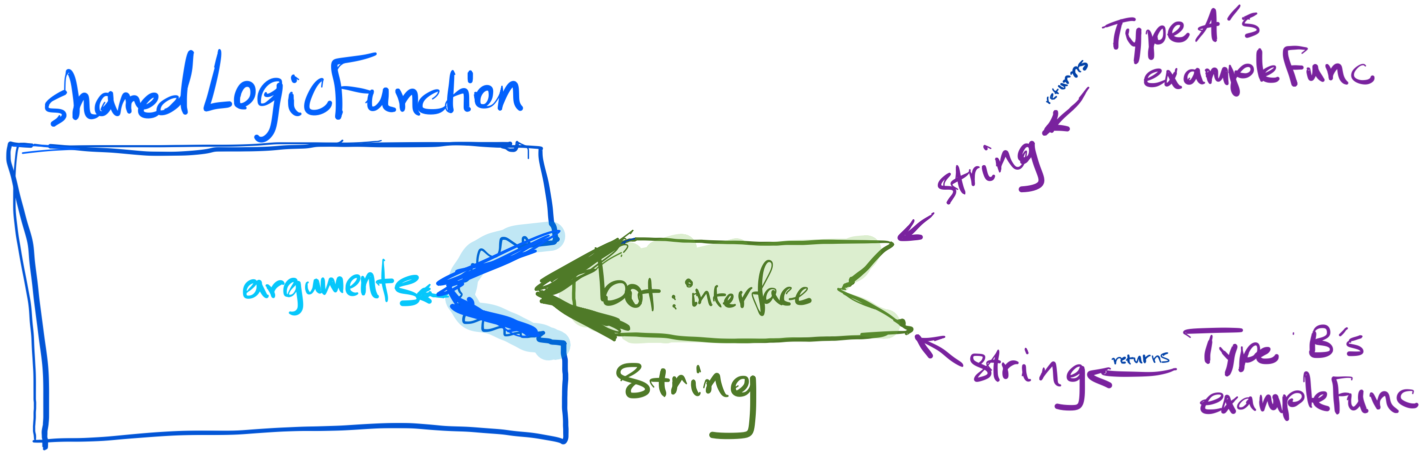

```
type bot interface {
    exampleFunction string
}
```

exampleFunction تابعی هست که اونا دو type struct
از انواع مختلف (به عنوان reciever) تعریف شده

exampleFunc تابعی هست که logic متفاوت رو پیاده می کنه
ولی از اون طرف از خروجی این تابع یه تابع دیگه استفاده خواهد کرد که اما logic
اون تابع یکسانه کاملا برای هر دو نوع struct ها

```
type A struct{}
type B struct{}
```

```
func (a A) exampleFunction() string {
        return "A"
}
```

هر دو تابع exampleFunc فرض شده است که:
1. logic متفاوتی دارند ولی نه به جای دوتاشون یه تابع پیاده می شه،
2. خروجی (return) هردو یکسانه که نسبت به یه interface باید گسترش کرد.
3. اسم های یکسانی دارند که باز نسبت به یه interface باید گسترش کرد.
4. هرکدام روی نوع خاص خودشون (به عنوان reciever) تعریف می شن.

```
func (b B) exampleFunction() String {
        return "B"
}
```

---

interface مشترکی باید یچی باشه

```
func shareLogicFunc(b bot) {
        fmt.println( b)
}
```

```
func main(){
        a := A{}
        b := B{}

        shareLogicFunc (a)
        shareLogicFunc (b)
}
```