درس ۴

Go

if err != nil

main.go

```go
v, err := f()
if err != nil {
  return err
}
```

یک کورس رایگان از MyWebUniversity

برنامه‌نویسی Go

درس ۴

تابع‌ها و خطا به شکل مقدار



برگشت چندگانه، نتیجه‌های نام‌دار، تابع‌های variadic، closure ها و if err != nil

تعریف تابع‌ها

func، نام، پارامترها با نوع آن‌ها بعد از نام‌ها، و نوع برگشتی بعد از پرانتز را بنویسید.

نام‌هایی که با حرف بزرگ شروع می‌شوند export می‌شوند، یعنی package های دیگر می‌توانند از ...

funcs.go

```go
func add(a, b int) int {
    return a + b
}

func Average(scores []float64) float64 {    // exported
    sum := 0.0
    for _, s := range scores { sum += s }
    return sum / float64(len(scores))
}
```

01

چند مقدار برگشتی

یک تابع Go می‌تواند چند مقدار برگرداند. الگوی استاندارد یک نتیجه و یک error را به عنوان ...

صدا کننده هر دو را می‌گیرد و فوراً error را بررسی می‌کند.

divide.go

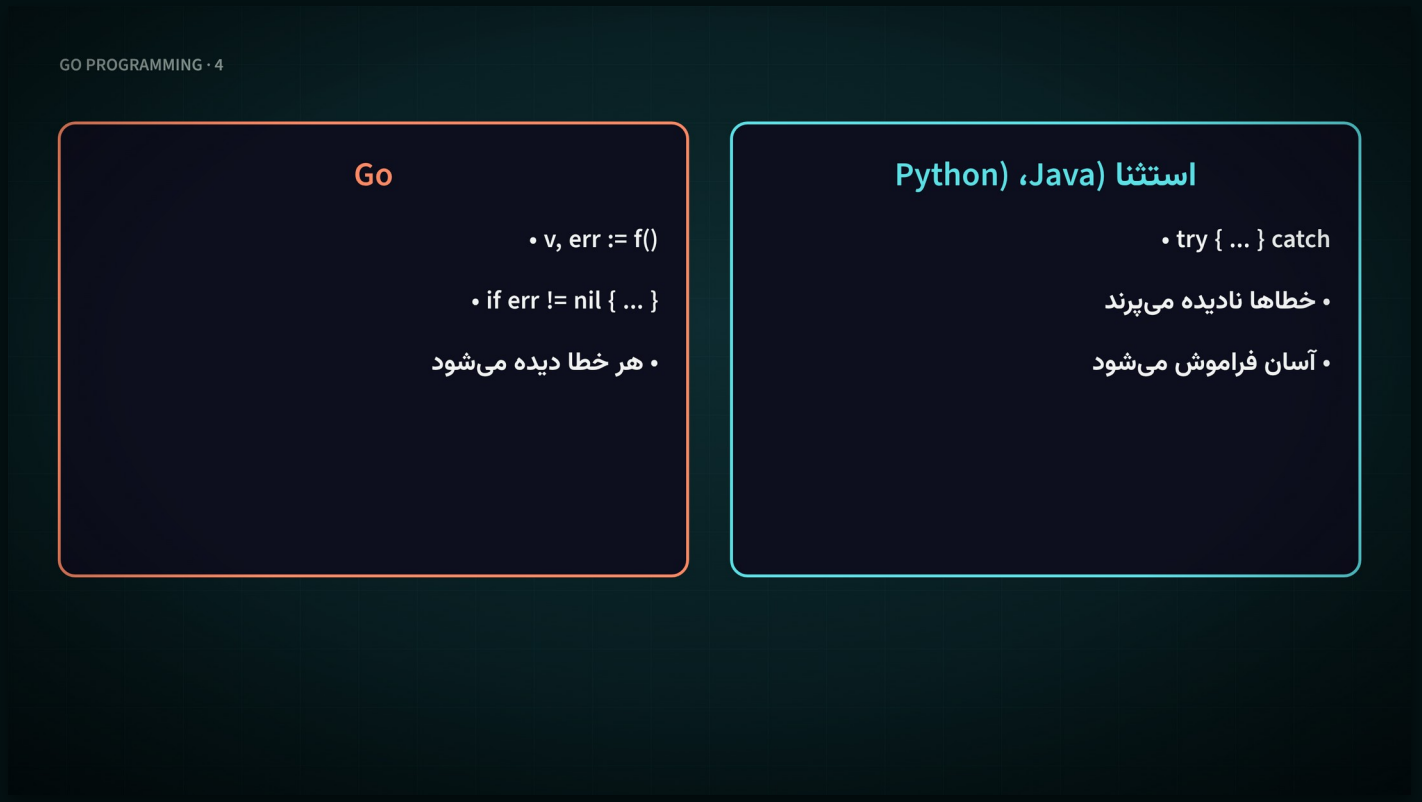
```go
func divide(a, b float64) (float64, error) {
    if b == 0 {
        return 0, errors.New("division by zero")
    }
    return a / b, nil
}

q, err := divide(10, 0)
if err != nil { fmt.Println("error:", err); return }
```

02

خطاها مقدار هستند

Go برای خطاهای عادی استثنا ندارد. یک error فقط یک مقدار است که برمی‌گردانید و با if err != ... اول تکراری به نظر می‌رسد، اما هر نقطهٔ ناکامی در کود دیده می‌شود، و هیچ چیز غیرمنتظره از یک ...



03

تابع‌های variadic

سه نقطه پیش از یک نوع هر تعداد آرگومان را می‌پذیرد، که به شکل یک slice می‌رسند. خود ...

یک slice موجود را با سه نقطه بعد از آن بفرستید.

variadic.go

```go
func sum(nums ...int) int {
    total := 0
    for _, n := range nums { total += n }
    return total
}
sum(1, 2, 3)               // 6
scores := []int{90, 85, 70}
sum(scores...)             // 245
```

04

تابع به شکل مقدار و closure ها

تابع‌ها مقدار هستند: آن‌ها را در متغیرها نگه دارید و به تابع‌های دیگر بفرستید. یک تابع ...

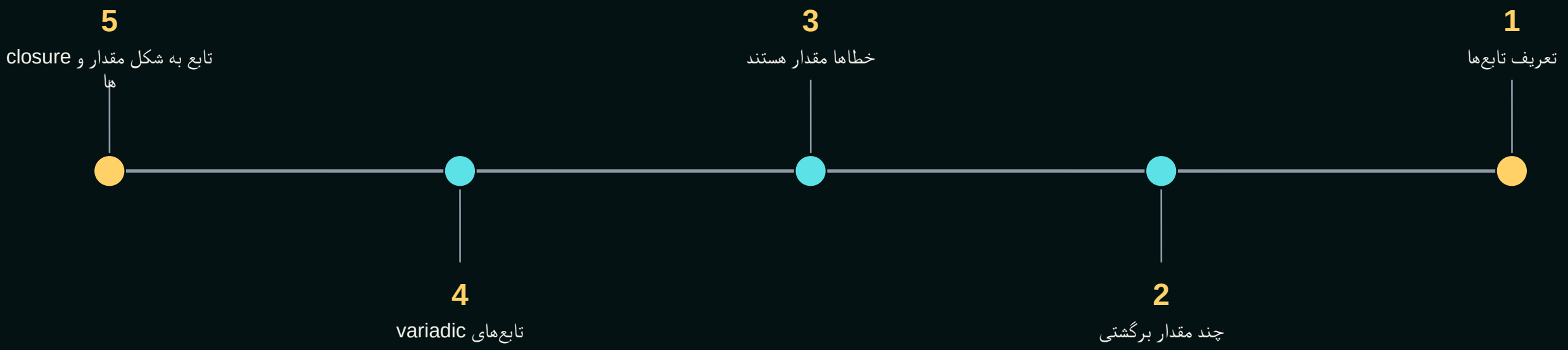
اینجا یک تابع شمارنده یک تابع نو برمی‌گرداند که شمارش خودش را به خاطر دارد.

closure.go

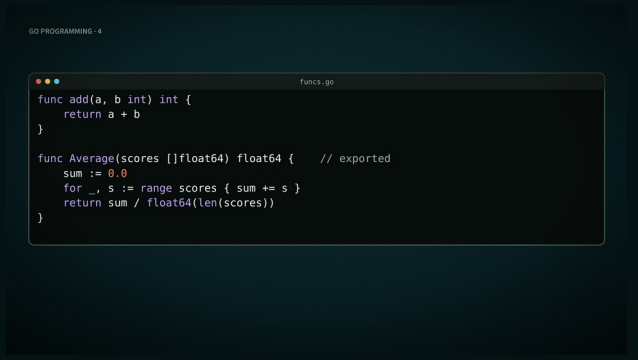
```go
func counter() func() int {
    count := 0
    return func() int {
        count++
        return count
    }
}
next := counter()
next(); next()
fmt.Println(next())       // 3
```

05

نقشهٔ درس


1
تعریف تابع‌ها
2
چند مقدار برگشتی
3
خطاها مقدار هستند
4
تابع‌های variadic
5
تابع به شکل مقدار و closure ها

منابع و مطالعهٔ بیشتر





تشکر از تماشای شما

MyWebUniversity.com

آموزش رایگان تکنالوژی از سال ۲۰۰۰ — انگلیسی و دری

@MyWebUniversity-Dari · @MyWebUniversity