

# Go release 1.22

Speaker's choice from “What's new”



“for” loop gotcha

```
done := make(chan bool)

values := []string{"a", "b", "c"}
for _, v := range values {
    go func() {
        fmt.Println(v)
        done <- true
    }()
}

// wait for all goroutines to complete before exiting
for _ = range values {
    <-done
}
```

# Go 1.21 or older

```
./prog.go:13:16: loop variable v captured by func literal
```

```
Go vet failed.
```

```
C
```

```
C
```

```
C
```

```
Program exited.
```

```
12         go func() {  
13             fmt.Println(v)  
14             done <- true  
15         }()
```

## Go 1.21 or older FIX

```
for _, v := range values {  
    go func(s string) {  
        fmt.Println(s)  
        done <- true  
    }(v)  
}
```

## Go 1.22+

```
for _, v := range values {  
    go func() {  
        fmt.Println(v)  
        done <- true  
    }()  
}
```



# HTTP Routing improved



GET /book/{id}



```
package main
```

```
import (
```

```
→   "log/slog"
```

```
→   "net/http"
```

```
)
```

```
// curl http://localhost:8080/books/golang
```

```
func main() {
```

```
→   mux := new(http.ServeMux)
```

```
→   mux.HandleFunc("GET /books/{id}", handleBook)
```

```
→   http.ListenAndServe(":8080", mux)
```

```
}
```

```
func handleBook(w http.ResponseWriter, r *http.Request) {
```

```
→   slog.Info("handle", "id", r.PathValue("id"))
```

```
}
```

2024/02/26 20:50:58 INFO handle id=golang

Trailing-slash redirection

`http.Handle("/images/",`

`then`

`/images` **redirects to** `/images/`

Null[T]

type NullBool

type NullByte

type NullFloat64

type NullInt16

type NullInt32

type NullInt64

type NullString

type NullTime

## Implements sql.Scanner

```
type Null[T any] struct {  
    V      T  
    Valid bool  
}
```

```
func (n *Null[T]) Scan(value any) error
```

```
var s Null[string]
err := db.QueryRow("SELECT name FROM foo WHERE id=?", id).Scan(&s)
...
if s.Valid {
    // use s.V
} else {
    // NULL value
}
```



Go have fun