



1

Почему Golang

Прост в использовании

```
func main() {  
    a := 10  
    fmt.Println(a)  
}
```

← Функция

← Переменная

← Вызов функции

Функциональная std библиотека

http

crypto

os

fmt

testing

io

log

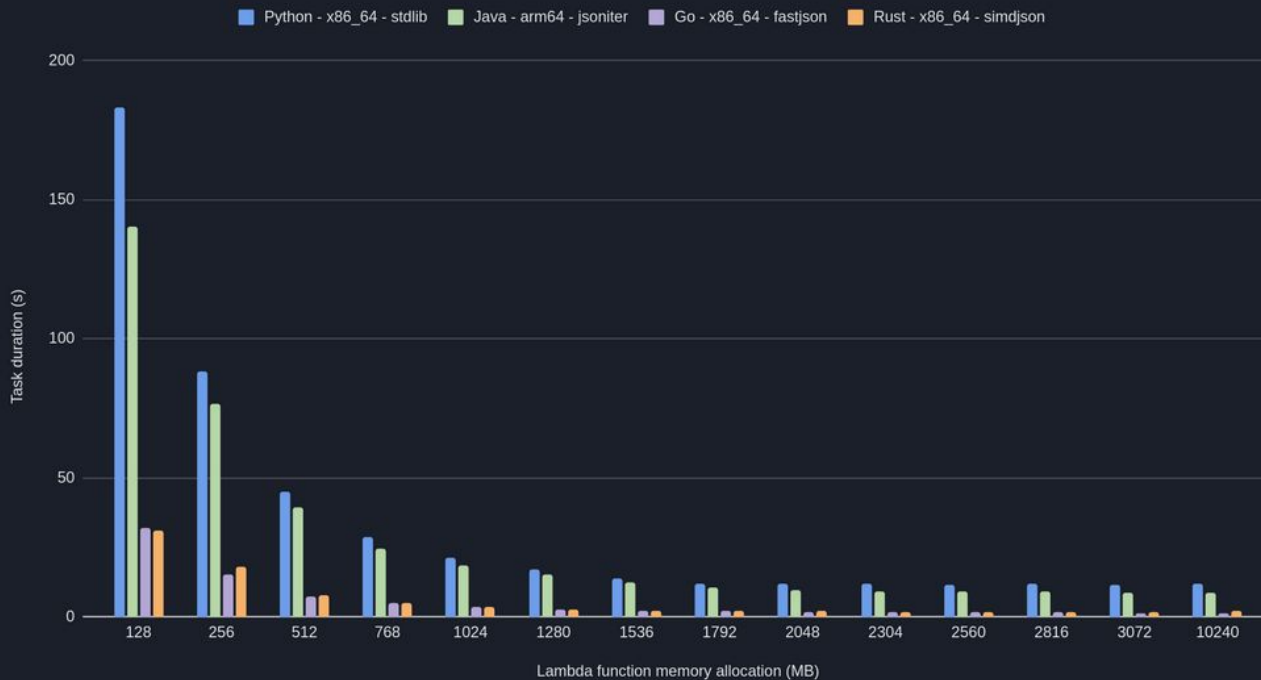
time

...

Высокая производительность

All languages. Task durations. Lower is better.

Task duration (s) for Lambda function memory allocations (MB)



Большая экосистема

921 660 github.com

13 172 gitlab.com

12 904 gopher.live

9 035 gitee.com

5 658 gopkg.in

~ Число внешних модулей на 2023 год

Компилируемый язык

```
- .
- |
- | .git
- | .vscode
- | geo
31 | go.mod
7.5M | main
496 | main.go
- | weather
```

Итоговое приложение со всем
необходимым runtime

Строгая статическая типизация

```
22 func GetMyLocation(city string) (*GeoData, error) {  
23     if city != 1 {
```

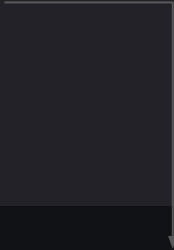
⊗ geo.go Проблемы: 1 из 1

invalid operation: city != 1 (mismatched types string and untyped int)

Прозрачная работа с ошибками

Явный возврат ошибки без
конструкций типа try / catch

```
func GetWeather(geo geo.GeoData, format int) (string, error) {  
    if format < 1 || format > 4 {  
        return "", ErrWrongFormat  
    }  
}
```



Встроенные инструменты

```
10 func TestGetWeather(t *testing.T) {
11     expected := "Moscow"
12     geoData := geo.GeoData{
13         City: expected,
14     }
15     format := 3
16     result, err := weather.GetWeather(geoData, format)
17     if err != nil {
18         t.Errorf("Пришла ошибка %v", err)
19     }
20     if !strings.Contains(result, expected) {
21         t.Errorf("Ожидалось %v, получение %v", expected, result)
22     }
}
```

ПРОБЛЕМЫ ВЫХОДНЫЕ ДАННЫЕ КОНСОЛЬ ОТЛАДКИ РЕЗУЛЬТАТЫ ТЕСТА ТЕРМИНАЛ ПОРТЫ КОММЕНТАРИИ

Running tool: /usr/local/bin/go test -timeout 30s -run ^TestGetWeather
\$ demo/weather/weather

Тестовый запуск в 12.06.2024, 13:47:07

TestGetWeather

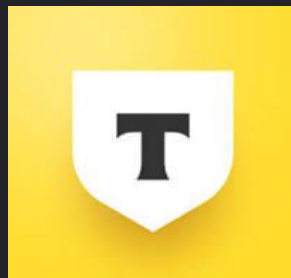
Кто использует?

Google



Microsoft

NETFLIX



Цель курса

- ✓ Разобраться в основах Golang: функции, переменные, управление потоком, slice, array, map, struct, interfaces
- ✓ Научиться работать с ОС и стандартной библиотекой
- ✓ Научиться писать приложения, использовать packages, env переменные, делать http запросы
- ✓ Собирать приложение, писать тесты, делать DI

Кому подойдет курс?

- ✓ Начинающим разработчикам без опыта
- ✓ Frontend разработчикам, которые хотят научиться писать backend
- ✓ Backend разработчикам, которые хотят освоить **НОВЫЙ ЯЗЫК**

Требования: нет