The Google logo is displayed in its classic multi-colored font. The 'G' is blue, the first 'o' is red, the second 'o' is yellow, the 'g' is blue, the 'l' is green, and the 'e' is red.

2007 год



Роберт
Гризмер

Роб
Пайк

Кен
Томпсон

Цели:

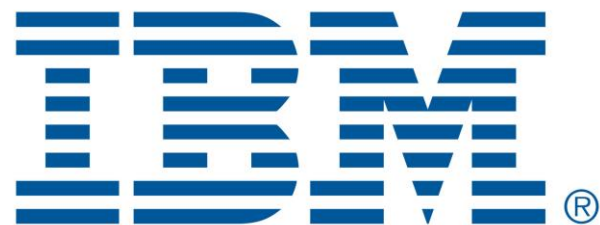
- Быстрая компиляция
- Компактный код
- Автоматическое освобождение неиспользуемой памяти
- Простота написания программ, способных одновременно выполнять несколько операций
- Качественная поддержка многоядерных процессоров

Они хотели создать быструю и удобную альтернативу популярным языкам C и C++

**2009 год – завершение разработки
языка**

**2012 год – Google открыл доступ
к языку и он быстро стал популярен
в цифровой индустрии**

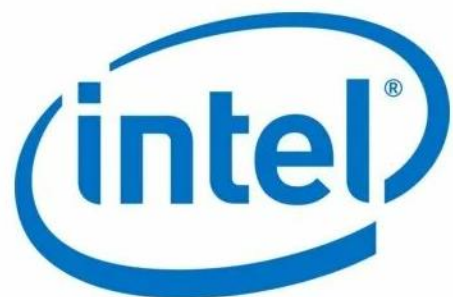
Go используют:



БАНК



SOUNDCLOUD

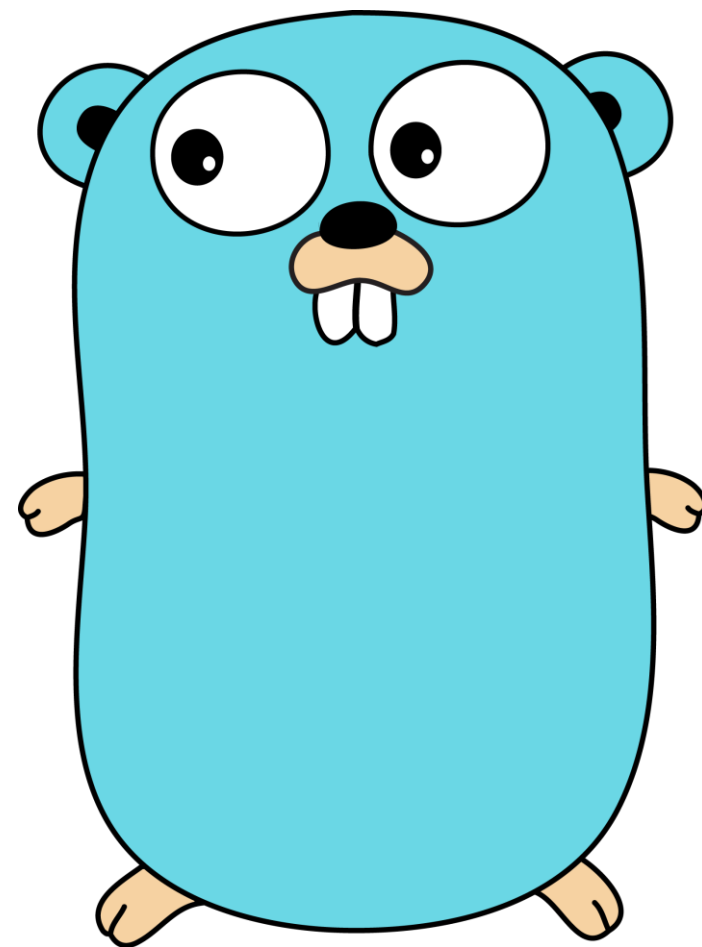


Uber



aviasales

Гофеп



Интерактивная среда:

<https://go.dev/play/>

Записать

Программа — набор инструкций для компьютера, которые позволяют решить определённую задачу


```
package main
```

```
import "fmt"
```

```
func main() {  
    fmt.Println("Hello, Go!")  
}
```

Записать

Каждый файл Go начинается с директивы **package**.

Пакет (package) представляет собой набор блоков кода, выполняющих похожие операции — например, форматирование строк или построение графических изображений.

Записать

Файлы Go почти всегда содержат одну или несколько директив **import**.

Каждый файл должен импортировать другие пакеты, чтобы использовать код, содержащийся в этих пакетах.

В программе указываем только те пакеты, которые нам действительно нужны, — то есть импортируем (import) эти пакеты.

Записать

Функция представляет собой набор строк кода, которые могут вызываться (выполняться) из других мест программы.

При запуске программа Go ищет функцию с именем `main` и выполняет ее в первую очередь; поэтому-то мы присвоили своей функции имя `main`.

Записать

Структура типичного файла Go:

Директива *package*. {package main

Директива *import*. {import "fmt"

Собственно код
программы. {func main() {
fmt.Println("Hello, Go!")
}

Эта строка сообщает, что остальной код
этого файла относится к пакету «main».

package main

Означает, что мы будем использовать код
форматирования текста из пакета «fmt».

import "fmt"

Функция «main» играет особую роль — именно
она выполняется при запуске программы.

func main() {

fmt.Println("Hello, Go!")

}

Для этого она вызывает
функцию «Println»
из пакета «fmt».

Эта строка выводит
сообщение «Hello, Go!»
в терминале (или брау-
зере, если вы использо-
уете среду Go Playground).

Записать

А если что то пойдет не так?

Программы Go должны соблюдать определенные правила, чтобы не сбивать с толку компилятор. Если вы нарушите одно из таких правил, компилятор выдаст сообщение об ошибке.

А если что то пойдет не так?

Строка

```
1 package main
2
3 import "fmt"
4
5 func main() {
6     fmt.Println "Hello, Go!"
7 }
```

Допустим,
вы забыли
поставить
здесь круглые
скобки...



А если что то пойдет не так?

Имя файла, используемое Go Playground.

Номер строки, в которой
произошла ошибка.

Описание ошибки.

```
prog.go:6:14: syntax error: unexpected literal "Hello, Go!" at end of statement
```

Позиция в строке, в которой произошла ошибка.

А если что то пойдет не так?

Если...	
Удалить директиву package...	package main
Удалить инструкцию import...	import "fmt"
Импортировать второй (неиспользуемый) пакет...	import "fmt" import "strings"
Переименовать функцию main...	func main hello
Преобразовать имя Println к нижнему регистру...	fmt.Println ("Hello, Go!")
Удалить имя пакета перед Println...	fmt. Println("Hello, Go!")

А если что то пойдет не так?

Удалите
директиву
package...



```
import "fmt"
```

```
func main() {  
    fmt.Println("Hello, Go!")  
}
```

... и получите
сообщение
об ошибке!



```
can't load package: package main:  
prog.go:1:1: expected 'package', found 'import'
```

А если что то пойдет не так?

Если...	...программа не будет работать, потому что...
Удалить директиву <code>package...</code> <code>package main</code>	Каждый файл Go должен начинаться с директивы <code>package</code>
Удалить инструкцию <code>import...</code> <code>import "fmt"</code>	Каждый файл Go должен импортировать все пакеты, которые в нем используются
Импортировать второй (неиспользуемый) пакет... <code>import "fmt"</code> <code>import "strings"</code>	Файлы Go должны импортировать <i>только</i> те пакеты, которые в них используются. (Это ускоряет компиляцию кода!)
Переименовать функцию <code>main...</code> <code>func mainhello</code>	Компилятор Go ищет функцию с именем <code>main</code> , чтобы выполнить ее при запуске программы
Преобразовать имя <code>Println</code> к нижнему регистру... <code>fmt.Pprintln("Hello, Go!")</code>	В Go учитывается регистр символов. <code>fmt.Println</code> — действительное имя, но имени <code>fmt.pprintln</code> не существует
Удалить имя пакета перед <code>Println...</code> <code>fmt.Println("Hello, Go!")</code>	Функция <code>Println</code> не принадлежит пакету <code>main</code> , поэтому перед вызовом функции необходимо указать имя пакета

