A Whirlwind Tour of Go Just the Cool Parts

Steve Willoughby

April 6, 2024



The Point

- "What is Go?"
- "What is it actually good for?"
- "Why should I care?"

- Designed by Rob Pike, Ken Thompson, and Robert Griesemer.
- Includes direct experience with C from day 1 to now.



- Designed by Rob Pike, Ken Thompson, and Robert Griesemer.
- Includes direct experience with C from day 1 to now.
- "If we were to design C today, knowing what we know now, on today's technology..."



- Designed by Rob Pike, Ken Thompson, and Robert Griesemer.
- Includes direct experience with C from day 1 to now.
- "If we were to design C today, knowing what we know now, on today's technology..."
 - ∴ Go's syntax is very much like C's
 - ... but cleaned up and streamlined a bit.



- Designed by Rob Pike, Ken Thompson, and Robert Griesemer.
- Includes direct experience with C from day 1 to now.
- "If we were to design C today, knowing what we know now, on today's technology..."
 - ∴ Go's syntax is very much like C's
 - ... but cleaned up and streamlined a bit.
- Dreamed up while waiting on a 45-minute C++ compile



- Designed by Rob Pike, Ken Thompson, and Robert Griesemer.
- Includes direct experience with C from day 1 to now.
- "If we were to design C today, knowing what we know now, on today's technology..."
 - .: Go's syntax is very much like C's
 - ... but cleaned up and streamlined a bit.
- Dreamed up while waiting on a 45-minute C++ compile
 - Fast compilation
 - Native binary compiler with low overhead
 - Strong static typing
 - Extraordinarily spartan



Go Syntax

• Type declarations follow identifier names

```
var x int
var UserName string
func AddNumbers(x, y int) int { ... }
func DivideNumbers(x, y int) (int, error) { ... }
type Shape struct {
   X
         int
        int
   Color ColorCode
```

Program Structure

- Basic unit is a package (namespace boundary).
- Multiple source files in a package, in the same directory tree.
- Every program must have a main package.
- The main package has a main function.
- Import packages into the program using the import statement.
- Always prefix identifiers from imported packages with their package name.
- Identifiers can be *public* or *private* w/r/t package boundaries.
 - Identifier names starting with an uppercase letter are public.
 - All others are private.

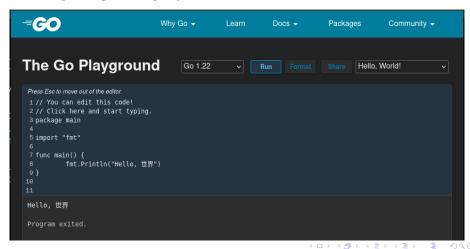


Hello, World

```
/* Standard-issue "Hello, World" program in Go */
package main
import "fmt"
func main() {
     fmt.Println("Hello, 世界")
}
```

The Playground

- Interactive playground to immediately try something in Go.
- https://go.dev/play/



7/10

Standard library package names are simple names:

```
import "fmt"
import "encoding/json"
import "flag"
import "math"
```

Standard library package names are simple names:

```
import (
    "fmt"
    "encoding/json"
    "flag"
    "math"
)
```

Standard library package names are simple names:

```
import (
    "fmt"
    "encoding/json"
    "flag"
    "math"
)
```

Getting packages from public repositories:

• Standard library package names are simple names:

```
import (
    "fmt"
    "encoding/json"
    "flag"
    "math"
)
```

Getting packages from public repositories:

```
import "github.com/MadScienceZone/go-gma/v5/dice"
```

Automatic API Documentation

• https://pkg.go.dev/repository-url

