Golang Fundamentals

main.go

```go
package main

import "fmt"

func main() {
  fmt.Println("Hello World!")
}

```

A Programming Language For Every Situation

- For the scientific and academic community; Python, R, Haskell
- For concurrency and reliability; Erlang, Elixir, Go
- For systems programming and hardware; C, C++, Rust
- For rapid web development; Javascript, Ruby

Born with a purpose

- Multicore processors
- Distributed networked systems
- Large codebases

Out of the box goodies

- CLI
- Web Server
- Arch + OS statically linked binaries
- **go fmt**
- **go test**
- Package Management

The Language

- Compiled
- Static, strongly typed with type inference
- Multi Paradigm (Functional, OO)
- Style enforced by the compiler
- Concise declaration
- Pointer are a explicit type
- Concurrency natives: channels & goroutines
- Errors are first class types and not throwable

Structure Is Important

- Less strict in the organization of individual sources
- More strict in the organization of groups of sources
- Use functions when you perform simple tasks, use types when you want to abstract away - and place them whenever you want*

github.com/golang-standards/project-layout

Code Reading Time

Trivia: Why was go the selected Lang for k8s?

- Docker code was a motivation because of its quality.
- Go encourages code quality and was proved for systems levels programming.

Resources

A Tour of Go

Brian Kernighan on successful language design

Go Time: Creating the Go programming language

Go In Practice, Manning, Matt Butcher, Matt Farina