Go

Go Go “”

Go C C++ Java Go Java

Go CC++ C C++ Go Java Go Go C C++

“” C Go

Go Go C Java “”

- Go struct literal S struct

  ```
  S { x: 1, y: 2, }
  ```

  Java JavaScript

- Pascal `x : int` Go `x int` `x, y int` var

  ```
  func foo(s string, x, y, z int, c bool) {
    ...
  }
  ```

  x, y, z `x , y` Go `x y` C Java

  ```
  func foo(s string, x int, y int, z int, c bool) {
    ...
  }
  ```

  “ ” x int“ parse ”

- Go `[]string` “” * “ parse” Go `[]*Struct` `*Struct []` “”“” `*[]*Struct` `*[]*pkg.Struct` C++ `vector<struct*>` Java Typed Racket

- “” switch, for Go switch C switchScheme case Scheme cond Go switch “” Scheme “” Scheme“”

Go `{` if

Go “”“” Java

Go gofmtgodef Go Emacs VIM C C++ Emacs Go IDE

Eclipse, IntelliJ Visual Studio IDE IDEGo refactor debugger GDB

Go package package IDE

Go package import GitHub repository Go package Go godep godep godep bug

Go

C C++ Go GC Go C/C++

Go mark-and-sweep JavaOCaml Chez Scheme

GC tuning Go GC GC Go

GC Go C C++ Go

“generics”

C++ Java Go generics Java genericsGenerics Haskell parametric polymorphism Java generics

Go generics interface {} C void* generics

JavaGo “hard code” Java collections PySonar Go

generics Go Go “ generics ” Go generics Go generics Go map generics

Go Unix

Go Scheme let-valuesGo —Go Go

```
ret, err := foo(x, y, z)
if err != nil {
        return err
}
```

`foo` `err` nilGo “” `err`

```
ret, _ := foo(x, y, z)
```

foo

“” `err` “”“” foo Error1 Error2 Java exception

`ret` `err` nil“”“” `err` nil `ret` nil return “” Go

Typed Racket PySonar “union type” union type {String, FileNotFound} String FileNotFound union type Typed Racket Haskell

Go “”

Go interface

Java Go implements“” struct

Go

Go Sort `T []string`

1. `TSorter` `StringSorter`
2. `StringSorter` `Len`, `Swap`, `Less`
3. `[]string` cast `StringSorter`
4. `sort.Sort`

sort Scheme OCaml

```
(sort '(3 4 1 2) <)
```

Scheme `<` sort Go interface Sort design pattern Go “” Go generics“”“” Sort Go “”: len swap

Java

goroutine

Goroutine Go Go goroutine “”

goroutine “continuation” continuation “”“”Continuation Amr Sabry continuation

Node.js “callback hell” continuation passing style (CPS) Scheme call/cc CPS continuation Scheme Gambit-C Chez Scheme

goroutine

defer

Go defer cleanup defer cleanup defer defer

 defer feature “” defer

defer

Go Unix Java Unix

 Go Java Unicode “code point” Go string byte cast “rune” cast byte Unix

HTML template

 Go template library “”Go template Go Lisp `{ {...} }`

```
{ {define "Contents"} }
{ {if .Paragraph.Length} }
<p>{ {.Paragraph.Content} }</p>
{ {end} }
{ {end} }
```

 struct `.Paragraph.Content`

 .go “””” Go HTML

 HTML Go

 Go C C++ Java Python Python Scheme HaskellGo

 Java Go PySonar Java Go

Alan Perlis Go

Go Go