

GO AWAY!!!!

Not really. Actually, Go right in :)

<https://github.com/nmpiazza/SPARSA-Go>

Info

- Google
- 2007 inception
- 2009 release
- Current version 1.5 (2015/08/19)
- 1.6 in development

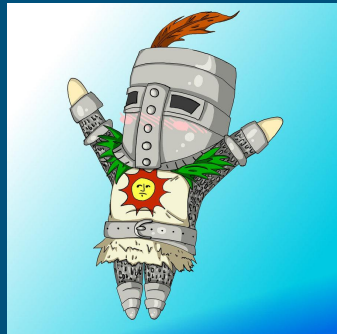
Thanks Wikipedia ([https://en.wikipedia.org/wiki/Go_\(programming_language\)](https://en.wikipedia.org/wiki/Go_(programming_language)))

Language Paradigms

- Compiled
- Concurrent
- Imperative
- Structured

[Thanks again Wikipedia](#)

Interesting things



- You can use it on Linux, BSD, OS X, Winderrs, Solaris (Praise the Sun!)
- C cross compatibility and compiling
 - gc
 - gccgo
 - cgo
- It is kind of object oriented, kind of
- There are Structs and these things called Interfaces
- The defer statement
- Not threads, Goroutines
- In Go, garbage collection is fast (<https://blog.golang.org/go15gc>)

Hello World!

hello.go

```
1 package main
2 import "fmt"
3
4 func main() {
5     fmt.Printf("Hello, world.\n")
6 }
```

Hello Again, World! :)

```
~/P/I/L/G/hello >>> go build hello.go
~/P/I/L/G/hello >>> ls
hello    hello.go
~/P/I/L/G/hello >>> ./hello
Hello, world.
~/P/I/L/G/hello >>> trash hello
~/P/I/L/G/hello >>> go run hello.go
Hello, world.
~/P/I/L/G/hello >>> 
```

Hello World Comparison!

hello.go

```
1 package main
2
3
4 import "fmt"
5
6 func main() {
7     fmt.Printf("Hello, world.\n")
8 }
```

hello.c

```
1 #include <stdio.h>
2
3 int main() {
4     printf("Hello World\n");
5     return 0;
6 }
```

Assign some variables

rand.go

```
1 package main
2 import "fmt"
3
4 func main() {
5     var my_int int = 44
6     randAssign := 44
7     fmt.Printf("my_int: %T", my_int)
8     fmt.Printf("randAssign: %T", randAssign)
9
10 }
```


Demos, Demos, Demos

Funcs.go



Maps.go



Server and Client Stuff

Resources that I am using currently

- https://golang.org/doc/effective_go.html
- <https://www.golang-book.com/>