



GoBackwards

Go, but worse.
Much worse.

Shaquan Nelson (sdn2115)

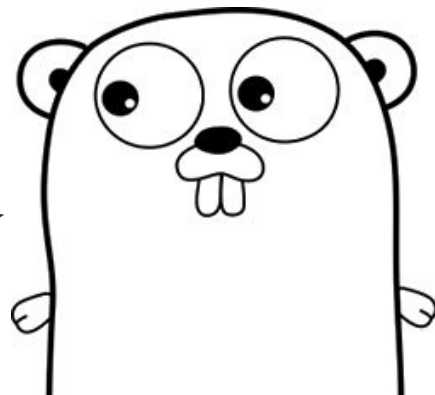
Julian Silerio (jjs2245)

Peter Richards (pfr2109)

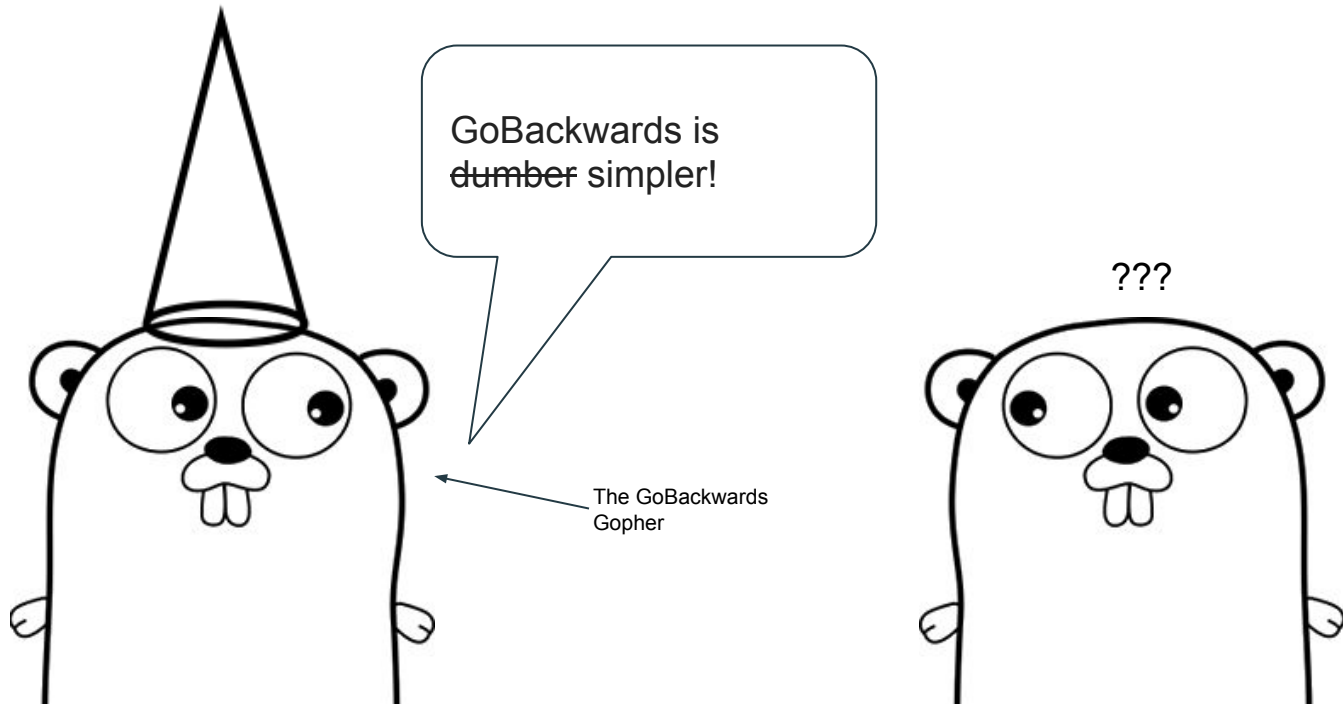
What is GoBackwards?

“Go is an open source programming language that makes it easy to build simple, reliable, and efficient software.”

The Go Gopher





What is GoBackwards?




Language Comparison

“Hello World” in Go

```
package main   
import "fmt"   
  
func main() {  
    fmt.Println("hello world")  
}
```

“Hello World” in GoBackwards

```
func main() {  
    println("hello world");   
}
```

Language Comparison

“Fibonacci Sequence” in Go

```
package main

import "fmt"

// fib returns a function that returns
// successive Fibonacci numbers.
func fib() func() int {
    a, b := 0, 1
    return func() int {
        a, b = b, a+b
        return a
    }
}

func main() {
    f := fib()
    fmt.Println(f(), f(), f(), f(), f())
}
```

“Fibonacci Sequence” in GoBackwards

```
func fib(x int) {
    if (x < 2) return 1;
    return fib(x-1) + fib(x-2);
}

func main() {
    var x int;
    for(x=0; x<6; x = x+1) {
        print(fib(x));
    }
    return 0;
}
```

Language Tutorial

- Declarations

- Variables and Types

- Integers `var x int;`
- Booleans `var x bool;`
- Strings `var x string;`

- Arrays

- `var x[5] int;`
- `x[3] = 3;`

- Functions

- Built in Main Function `func main(){}`
- Helper functions `func id(id type, id type,...){ }`

- Expressions

- Literals

- String `"Hello World!"`
- Number `42`
- Boolean `True | false`

- Assignment

- Declaration

`var x int; x= 10;`

- Arithmetic Operators

- `+, -, *, / print(30 / 15);`

- Built in Function Calls

- Call Ascii `ascii("star.png");`
- Call Print `print(34);`

Language Tutorial

- Declarations

- Variables and Types

- Integers `var x int;`
 - Booleans `var x bool;`
 - Strings `var x string;`

- Arrays

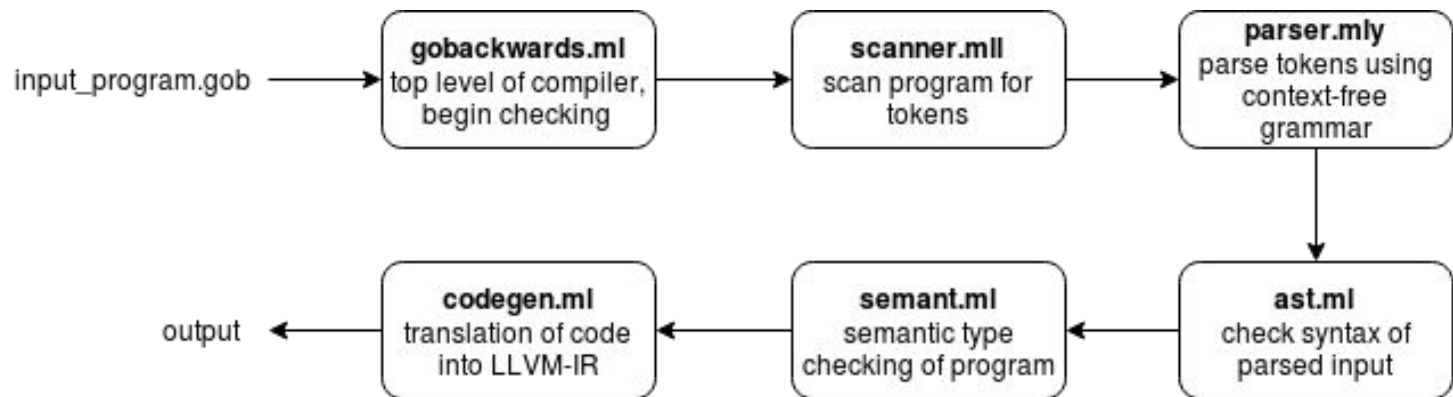
- `var x[5] int;`
 - `x[3] = 3;`

- Functions

- Built in Main Function `func main() {}`
 - Helper functions `func id(id type, id type, ...) { }`

```
39 block:
40     LBRACE locals_list stmt_list RBRACE
41     { { locals = List.rev $2;
42         stmts = List.rev $3 } }
43
44 fdecl:
45     FUNC ID signature block
46     { { fname = $2;
47         signature = $3;
48         body = $4 } }
49
50 signature:
51     LPAREN formals_opt RPAREN typ_opt
52     { { formals = List.rev $2;
53         ret_typ = $4 } }
54
```

Compiler pipeline



Example program

```
1
2 /* explicit types of parameters and return value */
3
4 func helloworld_helper(x string) string{
5     return(x);
6 }
7
8 }
9
10
11 /* multiple parameters and expressions in returns */
12 func add(x int, y int) int {
13     return x + y;
14 }
15
16 }
17
18
19 /*main method is needed to run any gobackwards function*/
20 func main(){
21     /* implicit declarations */
22     var s1 string;
23     var i1 int;
24     var b1 bool;
25
26     /* explicit declarations */
27     var s2 string;
28     var i2 int;
29     var b2 bool;
30
31
32
33
34
```

```
34
35     /*like in c, we need to allocate storage before
36     delcaring variables*/
37     s2 = "hi";
38     i2 = 10;
39     b2 = false;
40
41     println(s1);
42     println(s2);
43     print(i1);
44     print(i2);
45     printb(b1);
46     printb(b2);
47
48
49     /* calling helper functions */
50     print(add(3,4));
51     println(helloworld_helper("Hello, world!"));
52
53
54     /* calling built in ascii function */
55     ascii("star.png");
56
57
```

Testing

Test-Driven
Development

Edge Case
Testing:

making sure the user
knew what was and
was not allowed

Automation:
`./test.sh`
`./exe.sh -filepath`

Conclusion

- Major goals
 1. Make a Language Similar to Go
 2. Remove Some capabilities of Go to Make A More C-like Language
 3. Add picture-to-ASCII conversion capabilities to Go
- Success?
 - Successful implementation of basic Go syntax with C syntactic sugar
 - ASCII function is dependent on external C library
 - Overall successful group project despite tough road with many roadbumps