

GoWiki

**a [Wikipedia in the console] project
written in Go**

by Stacy Black, Jake Carns, and Prince Kannah

About Go:

Background and history of Go:

Go is an open-source, imperative programming language developed by Google. It was initially designed seven years ago, and first released 5 years ago, by Robert Griesemer, Rob Pike, and Ken Thompson. Go has similarities to C, but is designed to be simpler, safer, and easier to read.

Reasons for choosing Go:

As Google's flagship language, Go is set to become a staple language for programming - already it is surging in use, establishing its importance in the programming world and ensuring it will be well-used in the industry in the years to come.

Syntactic differences from other languages:

- When declaring variables, types are inferred, rather than explicitly stated
- In function parameters and return types, the parameter variable name is listed first, and its type follows. If there are multiple parameters of the same type, they can be combined, like `foo(x, y, int)`
- Return types of a function are placed after the parameters; for example, `foo(x, y, int) string`
- Can return more than one value from a single function (for example, `foo(x, y, int) (string, error)`) - if you only want to access one of these values, use an underscore in place of the value you are ignoring (`temp, _ = foo(4, 5)`)
- Semicolons can be used, but are not necessary. New lines can be used to separate statements instead
- Null values are represented as `nil`
- Typecasting is done by: `value.(type)`
- Error handling is done with `panic`, `defer`, and `recover`
- Pointers and structs are like C
- Uses `:=` to set values of already declared variables, or to declare a variable without the prefix `var` (only inside of code blocks)
- At the top level, you may only use keyword prefixes like `func`, `var`, or `type`. The shortened version of variable declaration (for example, `i := 10`) is not allowed there
- If statements require the use of brackets

Scoping/types:

Go is both strongly and statically typed. As mentioned earlier, types are also inferred. Its scope is static, with both local (for inside a function or block) and global (outside of all functions/blocks) variables.

The grammar of Go:

The grammar that is used by the Go language is built using extended Backus-Naur form. This metasyntax is a way of making a description of a formal language. This allows for a standardized context-free grammar that in the case of Go is considered regular, allowing for easy testing and integration for tools and utilities like integrated development environments (IDEs) with auto-complete.

Things that Go is useful for:

Go Language has an extremely powerful flag system. It has built-in support to handle parsing for default values and user-input values for flags of types boolean, numeric, or string. Usage documentation is also built in, which means that by default any program that is using the flags package can run with the `-h` command, which will print out the usage for each flag and its defaults as defined by the programmer.

Go is a high-powered tool for creating network applications with relative predictability and ease. Go has native concurrency support, like goroutines, which are used for threading. Go also has HTTP support for interacting with web servers to gather information from things like RESTful APIs. The GoWiki utility relies heavily on these built-in features to run.

Go is a programming language of minimalism. This means that it is focused around sacrificing some usability and functionality for speed. This means it is a great solution for scripts and standalone command-line apps. This also means that Go has very few dependencies and runs fairly consistently regardless of platform.

Irregularities (odd/unusual/bad) of Go:

Go language does not have any native GUI packages. This limits the scope of Go as a language because it lacks a medium. Go is often missing support for some of these more-advanced capabilities as it trades the advanced operations for speed and testability. This has helped Go to transition in the direction of a CLI- and utility-building language. Another reason for the transition to this area of software is because user-created objects lack usability and verbosity.

User-created objects are second-class citizens in Go. This makes it difficult to create libraries of data structures that other people can use. The second-class citizen objects create serious design compromises for these libraries because functionality implementation is limited by the abilities of the user-created objects.

Go can be object-oriented, but not in a traditional sense. It is not similar to common object-oriented programming languages like Java. This is partially because of the stipulations of user-created objects. This leads to their “class” being limited to the different predeclared and composite types in the language. The best we can do is add methods (a function with a specified receiver to our user-created objects), but this is far from covering the range of functionality of a traditional class like in java. The limitations of what objects are confined to is part of the reason that Go can struggle to support data and emulate functionality for more loosely-typed languages like the web. This is why Go’s JSON support is more limited than other languages.

Go has poor JSON support - Go’s built-in JSON libraries are cumbersome when it comes to parsing JSON structures because JSON is loosely typed and more free-form, unlike Go, which needs to have a type specified for everything at some point. This can be done during runtime or during program design time, depending on what paradigm.

About Our Project:

What our project does:

GoWiki is a command-line utility for searching Wikipedia from your console. It provides several different features to help you narrow your search quickly without having to sift through an overwhelming amount of information.

How our project works:

Our project uses Go’s powerful flag parsing system to add different support through flag arguments. It can simply be run as a vanilla executable, and it will prompt you for the minimum amount of information needed, or granular information may be provided, such as the number of search results to display, the method of displaying an article, and the search string to be used. It then prints out the results based on the information provided by the user, at which time the user may select the index of an article in the list to be read. This article will then be printed out using the specified output method (console or file) and then the option to read a different article from the list will be presented.

Code examples:

```
searchWiki(search string, limit int)
{
    fin.search = search
    inputColor("Searching for:\t", search)

    esc := url.QueryEscape(search)
```

```

body := getResults(esc, strconv.Itoa(limit))

s, err := loadSearch(body)
if err != nil {
}

printResults(s)

if len(fin.titles) == 0 {

    text := ""

    reader := bufio.NewReader(os.Stdin)
    inputColor("No results found. Try new search? (y/n) ")
    text, _ = reader.ReadString('\n')

    if text[0] == 'y' {
        searchPrompt()
    } else {
        os.Exit(0)
    }
}
readArticle()
}

```

Once a search term has been provided the searchWiki() function above is executed. The search term is prepared using Go's URL package to escape it, replacing spaces with %. This needs to be done since the term is going to be placed within a URL when we send the request to Wikipedia's servers. Once the term is escaped then request is made in getResults(). This returns a byte array containing the number of specified results.

```

loadSearch(body []byte) (*tmp, error) {

    var s = new(tmp)

    err := json.Unmarshal(body, &s)

    if err != nil {

```

```

        // un-comment bellow to see errors for json parsing
        //fmt.Println("whoops:", err)
    }
    return s, err
}

```

The byte array returned from `getResults()` is cleaned using the `loadSearch()` function using Go's JSON library, which turns the byte array into a human readable 2D string array. Once that is done, the search results are printed to the console. One of the big advantages of Go is that it is a very modern language and is very much aware that the ecosystem it lives in is web driven. So, to that end it has built-in libraries that help with dealing with common web formats like JSON.

The other crucial function of GoWiki is `readArticle()`. Another request is made to Wikipedia once an article has been chosen. The returned JSON is then unmarshaled into a query interface (similar to an object, interfaces allow for generics). This holds the data of the entire article. Once everything has been unmarshaled, we then use a for loop to go through query interface and print the first article that matches the query to the console I/O, or file if that was specified.

```

body := getArticle(check)
q, err := parseArticle(body)
qi, err := parseArticleIndex(body)

if *writeToFile {
    writeFile(qi, q)
} else {
    inputColor("reading entry:", text)
    for i, entry := range (*qi)["query"]["pageids"] {
        if i == 0 {
            // print to console
        }
    }
}

```

Input and expected output:

Input string of "go", default number of results (5):

```
mastermind@mycomputer MINGW64 /c/gowiki (master)
$ go run src/main.go
=====
GO WIKI
----- A command line utility for Wikipedia in GO Lang -----
=====
Enter a search string:
go|
```

Results:

```
go
Searching for:  go

0: Go
(https://en.wikipedia.org/wiki/Go)
=====

Go, G.O., or Go! may refer to:

1: Go (game)
(https://en.wikipedia.org/wiki/Go\_\(game\))
=====

Go (traditional Chinese: 圍棋; simplified Chinese: 围棋; pinyin: wéiqí; Japanese: 囲碁; rōmaji: igo; Korean: 바둑; romaja: baduk; literally: "encircling game") is an abstract strategy board game for two players, in which the aim is to surround more territory than the opponent.

2: Go (programming language)
(https://en.wikipedia.org/wiki/Go\_\(programming\_language\))
=====

Go (often referred to as golang) is a free and open source programming language created at Google in 2007 by Robert Griesemer, Rob Pike, and Ken Thompson.

3: GO Transit
(https://en.wikipedia.org/wiki/GO\_Transit)
=====

GO Transit is a regional public transit system in Southern Ontario, Canada, serving the Golden Horseshoe region.

4: Go Ask Alice
(https://en.wikipedia.org/wiki/Go\_Ask\_Alice)
=====

Go Ask Alice is a 1971 fiction book about a teenage girl who develops a drug habit at age 15, runs away from home, and eventually dies of a drug overdose.

Enter an index to read more:
|
```


After selecting the 2 index:

```
Enter an index to read more:
2
reading entry: 2

=====
= Go (programming language)
=====

Go (often referred to as golang) is a free and open source programming language
created at Google in 2007 by Robert Griesemer, Rob Pike, and Ken Thompson. It is
a compiled, statically typed language in the tradition of Algol and C, with gar
bage collection, limited structural typing, memory safety features and CSP-style
concurrent programming features added.

History
-----

The language was announced in November 2009. It is used in some of Google's prod
uction systems, as well as by other firms.
Two major implementations exist:
Google's Go compiler, "gc", is developed as open source software and targets var
ious platforms including Linux, OS X, Windows, various BSD and Unix versions, an
d also, since 2015, mobile devices, including smartphones.

Examples
-----

> Hello world
Here is a Hello world program in Go:

> Concurrency example
The following simple program demonstrates Go's concurrency features to implement
an asynchronous program. It launches two "goroutines" (lightweight threads): on
e waits for the user to type some text, while the other implements a timeout. Th
e select statement waits for either of these goroutines to send a message to the
main routine, and acts on the first message to arrive (example adapted from Chi
snall).

Projects using Go
-----

Some notable open-source applications in Go include:
The "Lightning Network", a Bitcoin network that allows for near-instantaneous Bi
```

External links

Official website

A Tour of Go (official)

Go Programming Language Resources (unofficial)

> Community and conferences

Gopher Academy, Gopher Academy is a group of developers working to educate and promote the Go community.

Golangprojects.com, lists programming jobs and projects where companies are looking for people that know Go

GopherCon The first Go conference. Denver, Colorado, USA April

Gopher Gala The first Go hackathon.

GopherConIndia The first Go conference in India. Bangalore Feb.

GolangUK The first Go conference in UK. London

dotGo European conference. Paris, France

GolangShow Go audio podcast GolangShow

Golang Argentina Gophers community blog in Argentina.

Return to results? (y/n)