NAME

gg – grep Go-language source code

SYNOPSIS

gg [*options*] *acdiknoprstvg regexp* [*file ...*]

DESCRIPTION

gg is classic grep (g/RE/p) with flag-directed Go token focus to search in package names, numbers, identifiers, comments, keywords, and more. Token flags are "acdiknoprstvg" in any order or combination:

- a search in All of the following
- c search in Comments (//... or /*...*/)
- d search in Defined non-types (iota, nil, new, true,...)
- i search in Identifiers ([alphabetic][alphabetic | numeric]*)
- k search in Keywords (if, for, func, go, ...)
- n search in Numbers ("255" matches 255, 0.255, 1e255)
- o search in Operators (, + - * / [ ] { } ( ) >>...)
- p search in Package names
- r search in Rune literals ('a', '\U00101234')
- s search in Strings (quoted or raw)
- t search in Types (bool, int, float64, map, ...)
- v search in Values (255 is 0b11111111, 0377, 255, 0xff)
- g search as grep, perform simple line-by-line matches in file

gg combines lexical analysis and Go-native pattern matching to extend **grep**(1) for Go developers. The search is restricted, seeking matches only in chosen token classes. A search in number literals can match *values*, "v 255" matches the numeric value 255 in source code as 0b1111_1111, 0377, 0o377, 255, 0xff, etc. Go's linear-time regular expression engine is Unicode-aware and supports many Perl extensions: numbers in identifiers are found with "gg i [0-9]" or "gg i [\d]", comments with math symbols by "gg c \p{Sm}", and Greek in strings via "gg s \p{Greek}" each with appropriate shell escaping.

gg searches files names listed on the command line or in a file of filenames provided the "-list" argument. If neither of these is present, gg reads file names from the standard input which is useful in shell pipelines such as "find . -name "*.go" | gg k fallthrough"

Files are Go source code files or directories. Source files include typical ".go" files; compressed ".go" files named ".go.bz2", ".go.gz", or ".go.zst" for Bzip2, Gzip, and ZStandard compression formats; archives of any such files in the formats "a.cpio", "a.tar", or "a.zip"; or, finally, compressed archives as in "a.cpio.bz2" and "a.tar.gz". If a named file is a directory then all Go source files in that directory are scanned without visiting subdirectories. With the "-r" flag enabled, named directories are processed recursively, scanning each Go source file or archive in that directory's hierarchy.

OPTIONS

–cpu=*n* Set the number of CPUs to use. Negative n means "all but n." Default is all.

–go=*bool*
Limit search to ".go" files. Default is true.

–h=*bool*
Display file names ("headers") on matches. Default is false for single-file searches and true otherwise.

–list=*file*
Search files listed one per line in the named file.

–log=*file*
Write a log of execution details to a named file. The special file names "[stdout]" and "[stderr]" refer to the stdout and stderr streams. (Last line of log details efficiency.)

–n=*bool*
Display line numbers following each match. Numbers count from one per file. Default is false.

–output=*file*
gg output is normally to stdout but may be directed to a named file. The special names "[stdout]" and "[stderr]" refer to the stdout and stderr streams.

–r=*bool* Search directories recursively. Default is false.

–visible=*bool*
Restrict search to visible files, those with names that do not start with "." (in the shell tradition). Default is true.

acdiknoprstvCDIKNOPRSTVg
The Go token class flags have an upper case negative form to disable the indicated class. Used with "a" for "all", "aCS" means "search All tokens except Comments and Strings." Flag "g" means search as if the grep command, ignore Go lexical analysis and match lines.

EXAMPLES

To search for comments containing "case" (ignoring switch statements) in every ".go" file in the current working directory, use the command:

```
gg c case .
```

To find number literals containing the digits 42 in ".go" files located anywhere in the current directory's hierarchy, use the command:

```
gg -r n 42 .
```

Find numbers with values of 255 (0b1111_1111, 0377, 0o377, 255, 0xff) in ".go" files in the gzipped **tar**(1) archive omega with the command:

```
gg v 255 omega.tar.gz
```

AUTHOR

Michael T. Jones (https://github.com/MichaelTJones)

SEE ALSO

https://golang.org/pkg/regexp/syntax/
https://en.wikipedia.org/wiki/Unicode_character_property