GO 1.13 - it's finally here

Disclaimer

This is my personal list of the interesting parts of the new 1.13 release.
For more information visit: https://golang.org/doc/go1.13

Number literals

Digit separators

Digits of any number literal may now be "grouped" with underscores
3_342_000 = 3342000

Number literals

Digit separators

Digits of any number literal may now be "grouped" with underscores
3_342_000 = 3342000

Binary literals

The prefix `0b` or `0B` can now be used to indicate a binary integer literal
0b011101 / 0b_011_101

Number literals

Digit separators

Digits of any number literal may now be "grouped" with underscores
3_342_000 = 3342000

Binary literals

The prefix `0b` or `0B` can now be used to indicate a binary integer literal
0b011101 / 0b_011_101

Imaginary literals

By using the suffix `i` any (binary,decimal,hexadecimal) integer or float can be represented as an imaginary part of a complex constant. #Math
2.71828i

Signed shift counts

The shifting operators `<<` and `>>` are now working with signed counts to and therefore eliminate a lot of uint conversions in the future

Runtime changes

Out of range panic

Out of range panic messages now include the requested index and the length of the slice
using `s[3]` on a slice with 1 length will now result in:

```
runtime error: index out of range [3] with length 1
```

Runtime changes

Out of range panic

Out of range panic messages now include the requested index and the length of the slice
using `s[3]` on a slice with 1 length will now result in:

```
runtime error: index out of range [3] with length 1
```

Faster defer

Performance improvement of most uses of defer by 30%

Core library

TLS 1.3

Is now enabled by default and can be disabled with `GODEBUG=tls13=0`
The opt-out possibility will be removed with version 1.14

Core library

TLS 1.3

Is now enabled by default and can be disabled with `GODEBUG=tls13=0`
The opt-out possibility will be removed with version 1.14

ED25519

Is now part of the core library.

Core library

TLS 1.3

Is now enabled by default and can be disabled with `GODEBUG=tls13=0`
The opt-out possibility will be removed with version 1.14

ED25519

Is now part of the core library.

SSL 3.0

Support is now deprecated and will be removed with GO 1.14

Core library

os.UserConfigDir()

Returns (as the name suggests) the user-specific config directory

- Linux: `$XDG_CONFIG_HOME` if not empty or `$HOME/.config` as fallback
- Darwin: `$HOME/Library/Application Support`
- Windows: `%AppData%`

Godoc

The godoc webserver is no longer included in the main binary distribution.
But it is available as an installable package with:

```
go get golang.org/x/tools/cmd/godoc
```

Now to the real cool stuff

Modules 1/2

GO111MODULE

The `GO111MODULE` environment variable is still on the value `auto` but the behaviour of the value changed to activate the module-aware mode of the `go` command — even if the current directory is within `GOPATH/src`

Modules 1/2

GO111MODULE

The `GO111MODULE` environment variable is still on the value `auto` but the behaviour of the value changed to activate the module-aware mode of the `go` command — even if the current directory is within `GOPATH/src`

GOPRIVATE

A new environment variable to indicate packages which are not publicly available — this will disable the go proxy and checksum database `GOPRIVATE=*.corp.example.com, rsc.io/private`

Modules 2/2

GOPROXY

The `GOPROXY` environment variable may now be set to:

- `comma-separated list of proxy URLs`
- and or the special token `direct`

The default value is now https://proxy.golang.org,direct.
When resolving a package path to its containing module, the go command will try all candidate module paths on each proxy in the list in succession. An unreachable proxy or HTTP status code other than 404 or 410 terminates the search without consulting the remaining proxies.

Error wrapping 1/2

As

Finds the first error in a given error’s chain (sequence of wrapped errors)

Error wrapping 1/2

As

Finds the first error in a given error’s chain (sequence of wrapped errors)

Is

Reports whether a given error value matches an error in another’s chain

Error wrapping 1/2

As

Finds the first error in a given error’s chain (sequence of wrapped errors)

Is

Reports whether a given error value matches an error in another’s chain

Unwrap

Returns the result of calling Unwrap on a given error, if one exists

Error wrapping 1/2

As

Finds the first error in a given error's chain (sequence of wrapped errors)

Is

Reports whether a given error value matches an error in another's chain

Unwrap

Returns the result of calling Unwrap on a given error, if one exists

fmt.Errorf

Has a new symbol `%w` whose operand must be an error

Error wrapping 2/2

Simple wrapping: https://play.golang.org/p/gqdReVEA-sG

Custom errors: https://play.golang.org/p/_qqy_rZhBsc

Private proxies

- Goproxy - https://github.com/goproxy/goproxy
- Athens - https://github.com/gomods/athens
- Thumbai - https://thumbai.app/
- Artifactory - https://jfrog.com/artifactory/