using vim at work

@danishprakash

danish prakash

software engineer @HackerRank
https://danishpraka.sh/

what led to this talk?

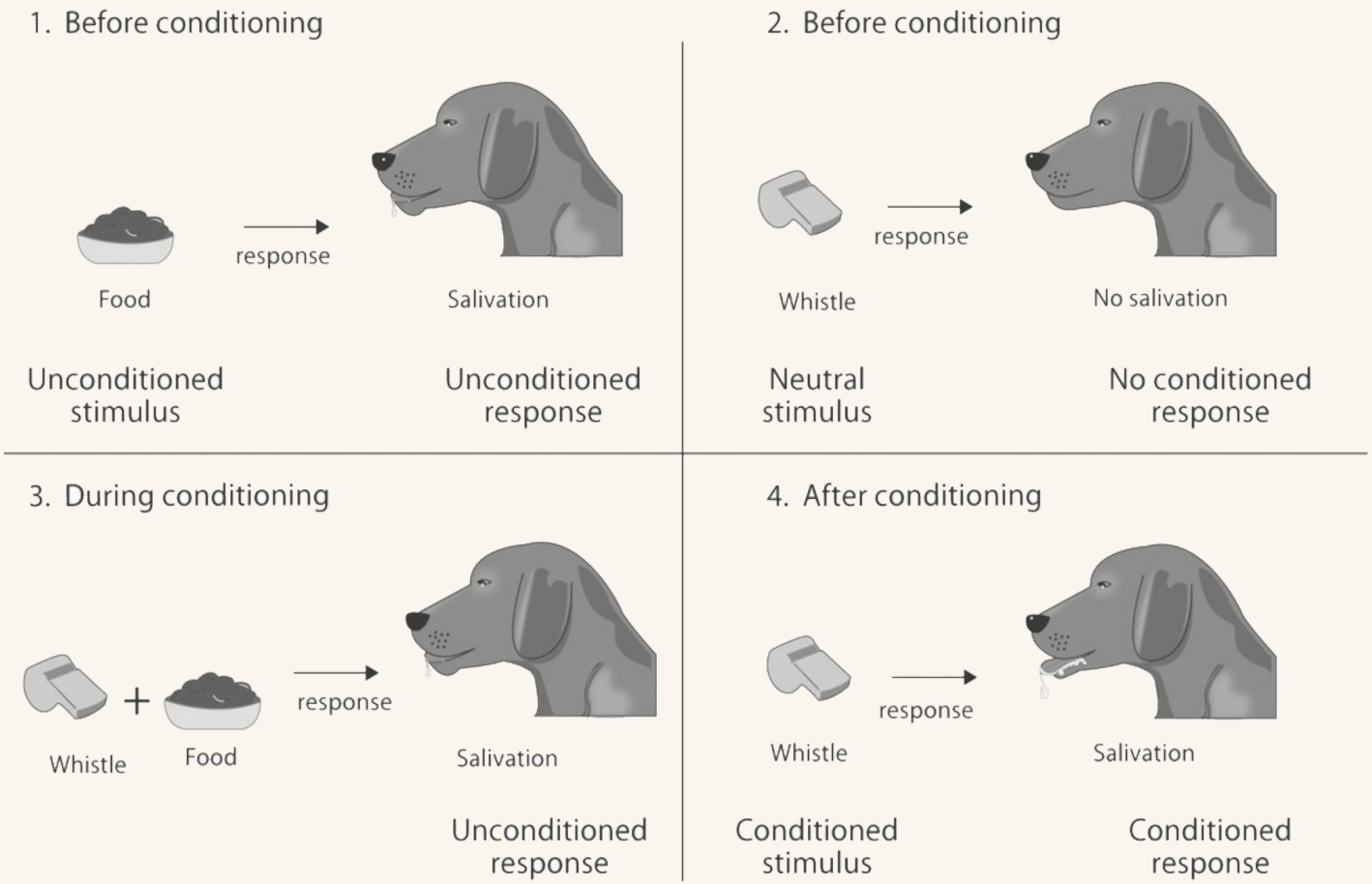
Mentor

Kindly change your editor by Monday or else it will be difficult for us to work together. See you on Monday!

psychology

classical conditioning

an associative learning process that occurs between novel and familiar stimuli



http://changecom.files.wordpress.com/2012/12/classical-conditioning.jpg

mere-exposure effect is a psychological phenomenon by which people tend to develop a preference for things merely because they are familiar with them.

from Wikipedia, the free encyclopedia

learning something new is easier when complemented with familiar elements.

Reder et al
Carnegie Mellon University

mouse & scroll

“but vim promotes a mouse-less workflow!”

mouse is ubiquitous

navigating websites

highlighting text

creating presentations (like this one)

managing emails

modifying system settings (non GNU/Linux systems)

designing graphics

exploring image galleries

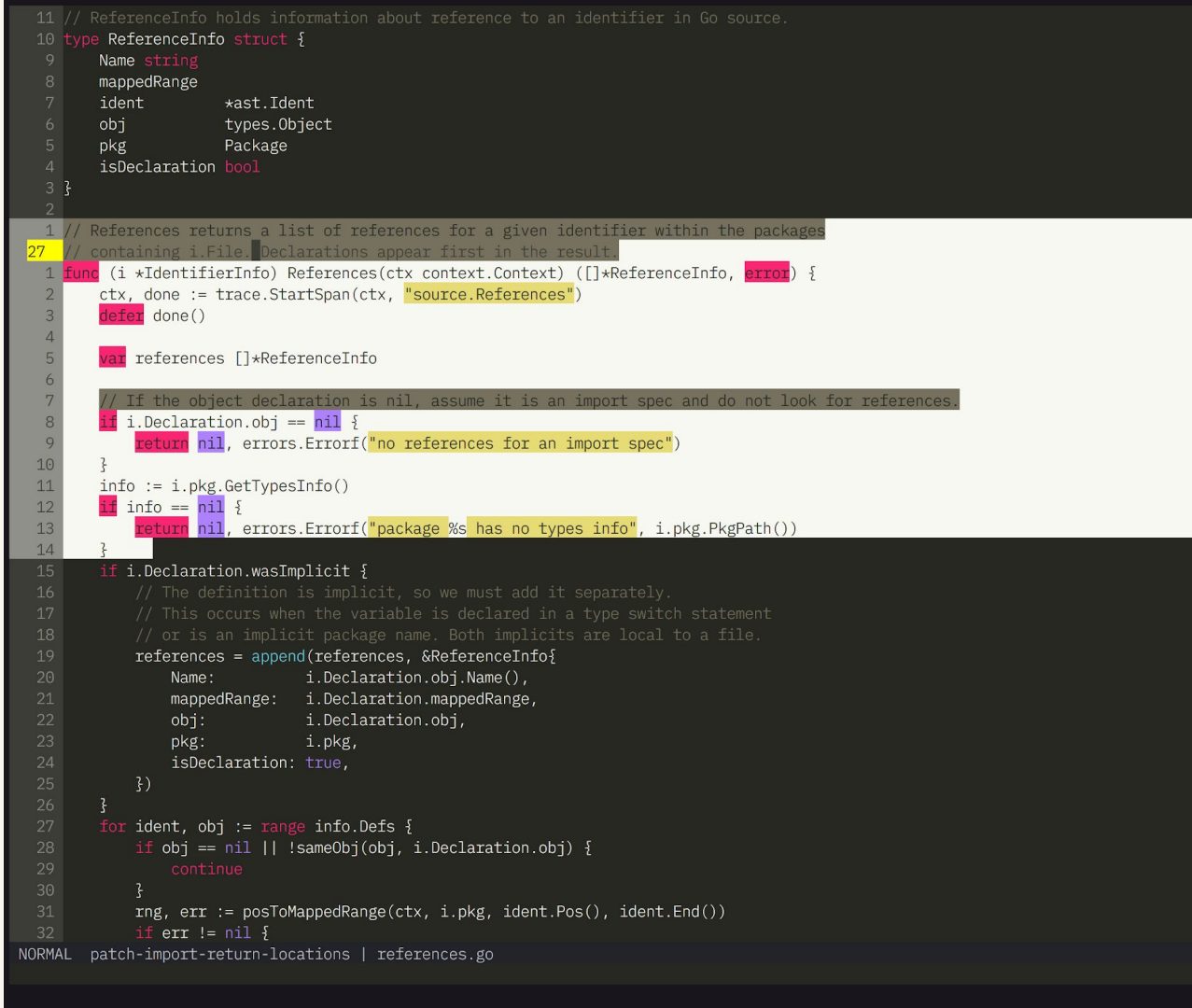
keyboard vs mouse

test subjects report that keyboarding is faster than mousing, whereas stopwatch proves mousing is faster than keyboarding.

Bruce Tognazzini, Apple Interface Design

```
:set mouse-=a
```

mouse selection



term buffer scroll

```
+github.com/kisielk/gotool v1.0.0/go.mod h1:XhKaO+MFFWcvkIS/tQcRk01m1F5IRFswLeQ+oQHNcck=
+github.com/kr/pretty v0.1.0/go.mod h1:dAy3ld7l9f0ibDNOQOHHMYYIIbhfbHSm3C4ZsoJORNo=
+github.com/kr/pty v1.1.1/go.mod h1:pFQYn66WHrOpPYNljwOMqo10TkYh1fy3cYio2l3bCsQ=
+github.com/kr/text v0.1.0/go.mod h1:4Jbv+DJW3UT/LiOwJeYQe1efqtUx/iVham/4vfdArNI=
+github.com/pmezard/go-difflib v1.0.0 h1:4DBwDE0NGyQoBHbLQYPwSUPoCMWR5BEzIk/f1lZbAQM=
+github.com/pmezard/go-difflib v1.0.0/go.mod h1:iKH77koFhYxTK1pcRnkKkqfTogsbg7gZNVY4sRDYZ/4=
+github.com/rogpeppe/go-internal v1.3.0/go.mod h1:M8bDsm7K2OlrFYOpmOWEs/qY81heoFRclV5y23lUDJ4=
+github.com/sergi/go-diff v1.0.0 h1:Kpca3qRNrduNnOQeazBd0ysaKrUJiIuISHxogkT9RPQ=
+github.com/sergi/go-diff v1.0.0/go.mod h1:0CfEIISq7TuYL3j771MWULgwwjU+GofnZX9QAmXWZgo=
+github.com/stretchr/objx v0.1.0/go.mod h1:HFkY916IF+rwdDfMAkV7OtwuqBVzrE8GR6GFx+wExME=
+github.com/stretchr/testify v1.4.0 h1:2E4SXV/wtOkTonXsotYi4li6zVWxYlZuYNCXe9XRJyk=
+github.com/stretchr/testify v1.4.0/go.mod h1:j7eGeouHqKxXV5pUuKE4zz7dFj8WfuZ+81PSLYec5m4=
 golang.org/x/crypto v0.0.0-20190308221718-c2843e01d9a2/go.mod h1:djNgcEr1/C05ACkg1iLfiJU5Ep61QUkGW8qpdssI0+w=
+golang.org/x/crypto v0.0.0-20190510104115-cbcb75029529/go.mod h1:yigFU9vqHzYiE8UmvKecakEJjdnWj3jj499lnFckfCI=
+golang.org/x/mod v0.0.0-20190513183733-4bf6d317e70e/go.mod h1:mXi4GBBbnImb6dmsKGUJ2LatrhH/nqhxcFungHvyanc=
+golang.org/x/net v0.0.0-20190311183353-d8887717615a/go.mod h1:t9HGtf8HONx5eT2rtn7q6eTqICYqUVnKs3thJo3Qplg=
+golang.org/x/net v0.0.0-20190404232315-eb5bcb51f2a3/go.mod h1:t9HGtf8HONx5eT2rtn7q6eTqICYqUVnKs3thJo3Qplg=
 golang.org/x/net v0.0.0-20190620200207-3b0461eec859 h1:R/3boaszxrf1GEUWTVDzSKVwLmSJpwZ1yqXm8j0v2QI=
 golang.org/x/net v0.0.0-20190620200207-3b0461eec859/go.mod h1:z5CRVTTTmAJ677TzLLGU+0bjPO0LkuOLi4/5GtJWs/s=
 golang.org/x/sync v0.0.0-20190423024810-112230192c58 h1:8gQV6CLnAEikrhgkHFbMAEhagSSnXWGV915qUMm9mrU=
 golang.org/x/sync v0.0.0-20190423024810-112230192c58/go.mod h1:RxMgew5VJxzue5/jJTE5uejpjVlOe/izrB70Jof72aM=
(patch-import-return-locations) tools/gopls $
```

```
    "golang.org/x/tools/go/ast/astutil"
    "golang.org/x/tools/internal/lsp/protocol"
    "golang.org/x/tools/internal/span"
    "golang.org/x/tools/internal/telemetry/trace"
    errors "golang.org/x/xerrors"
)

// IdentifierInfo holds information about an identifier in Go source.
type IdentifierInfo struct {
    Name string
    File ParseGoHandle
    mappedRange

    Type struct {
        mappedRange
        Object types.Object
    }

    Declaration Declaration

    pkg             Package
    ident           *ast.Ident
    wasEmbeddedField bool
    qf              types.Qualifier
```

```
    f, err := view.GetFile(ctx, uri)
    if err != nil {
        return nil, err
    }
    ident, err := source.Identifier(c
    if err != nil {
        return nil, err
    }
    decRange, err := ident.Declaratio
    if err != nil {
        return nil, err
    }
    return []protocol.Location{
        {
            URI:   protocol.NewURI(id
            Range: decRange,
        },
    }, nil
}

func (s *Server) typeDefinition(ctx c
TypeDefinitionParams) ([]protocol.Loc
    uri := span.NewURI(params.TextDoc
    view := s.session.ViewOf(uri)
```

clipboard ambiguity

```
+github.com/kisielk/gotool v1.0.0/go.mod h1:XhKaO+MFFWcvkIS/tQcRk01m1F5IRFswLeQ+oQHNcck=
+github.com/kr/pretty v0.1.0/go.mod h1:dAy3ld7l9f0ibDNOQOHHMYYIIbhfbHSm3C4ZsoJORNo=
+github.com/kr/pty v1.1.1/go.mod h1:pFQYn66WHrOpPYNljwOMqo10TkYh1fy3cYio2l3bCsQ=
+github.com/kr/text v0.1.0/go.mod h1:4Jbv+DJW3UT/LiOwJeYQe1efqtUx/iVham/4vfdArNI=
+github.com/pmezard/go-difflib v1.0.0 h1:4DBwDE0NGyQoBHbLQYPwSUPoCMWR5BEzIk/f1lZbAQM=
+github.com/pmezard/go-difflib v1.0.0/go.mod h1:iKH77koFhYxTK1pcRnkKkqfTogsbg7gZNVY4sRDYZ/4=
+github.com/rogpeppe/go-internal v1.3.0/go.mod h1:M8bDsm7K2OlrFYOpmOWEs/qY81heoFRclV5y23lUDJ4=
+github.com/sergi/go-diff v1.0.0 h1:Kpca3qRNrduNnOQeazBd0ysaKrUJiIuISHxogkT9RPQ=
+github.com/sergi/go-diff v1.0.0/go.mod h1:0CfEIISq7TuYL3j771MWULgwwjU+GofnZX9QAmXWZgo=
+github.com/stretchr/objx v0.1.0/go.mod h1:HFkY916IF+rwdDfMAkV7OtwuqBVzrE8GR6GFx+wExME=
+github.com/stretchr/testify v1.4.0 h1:2E4SXV/wtOkTonXsotYi4li6zVWxYlZuYNCXe9XRJyk=
+github.com/stretchr/testify v1.4.0/go.mod h1:j7eGeouHqKxXV5pUuKE4zz7dFj8WfuZ+81PSLYec5m4=
 golang.org/x/crypto v0.0.0-20190308221718-c2843e01d9a2/go.mod h1:djNgcEr1/C05ACkg1iLfiJU5Ep61QUkGW8qpdssI0+w=
+golang.org/x/crypto v0.0.0-20190510104115-cbcb75029529/go.mod h1:yigFU9vqHzYiE8UmvKecakEJjdnWj3jj499lnFckfCI=
+golang.org/x/mod v0.0.0-20190513183733-4bf6d317e70e/go.mod h1:mXi4GBBbnImb6dmsKGUJ2LatrhH/nqhxcFungHvyanc=
+golang.org/x/net v0.0.0-20190311183353-d8887717615a/go.mod h1:t9HGtf8HONx5eT2rtn7q6eTqICYqUVnKs3thJo3Qplg=
+golang.org/x/net v0.0.0-20190404232315-eb5bcb51f2a3/go.mod h1:t9HGtf8HONx5eT2rtn7q6eTqICYqUVnKs3thJo3Qplg=
 golang.org/x/net v0.0.0-20190620200207-3b0461eec859 h1:R/3boaszxrf1GEUWTVDzSKVwLmSJpwZ1yqXm8j0v2QI=
 golang.org/x/net v0.0.0-20190620200207-3b0461eec859/go.mod h1:z5CRVTTTmAJ677TzLLGU+0bjPO0LkuOLi4/5GtJWs/s=
 golang.org/x/sync v0.0.0-20190423024810-112230192c58 h1:8gQV6CLnAEikrhgkHFbMAEhagSSnXWGV915qUMm9mrU=
 golang.org/x/sync v0.0.0-20190423024810-112230192c58/go.mod h1:RxMgew5VJxzue5/jJTE5uejpjVlOe/izrB70Jof72aM=
(patch-import-return-locations) tools/gopls $
```

```
    "golang.org/x/tools/go/ast/astutil"
    "golang.org/x/tools/internal/lsp/protocol"
    "golang.org/x/tools/internal/span"
    "golang.org/x/tools/internal/telemetry/trace"
    errors "golang.org/x/xerrors"
)

// IdentifierInfo holds information about an identifier in Go source.
type IdentifierInfo struct {
    Name string
    File ParseGoHandle
    mappedRange

    Type struct {
        mappedRange
        Object types.Object
    }

    Declaration Declaration

    pkg              Package
    ident            *ast.Ident
    wasEmbeddedField bool
    qf               types.Qualifier
```

```
    f, err := view.GetFile(ctx, uri)
    if err != nil {
        return nil, err
    }
    ident, err := source.Identifier(c
    if err != nil {
        return nil, err
    }
    decRange, err := ident.Declaratio
    if err != nil {
        return nil, err
    }
    return []protocol.Location{
        {
            URI:   protocol.NewURI(id
            Range: decRange,
        },
    }, nil
}

func (s *Server) typeDefinition(ctx c
TypeDefinitionParams) ([]protocol.Loc
    uri := span.NewURI(params.TextDoc
    view := s.session.ViewOf(uri)
```

bottomline

:set mouse=a

even if you are a power user
or keyboarder

pair programming

easy on the person next to you

unexpected surprises

no more of them

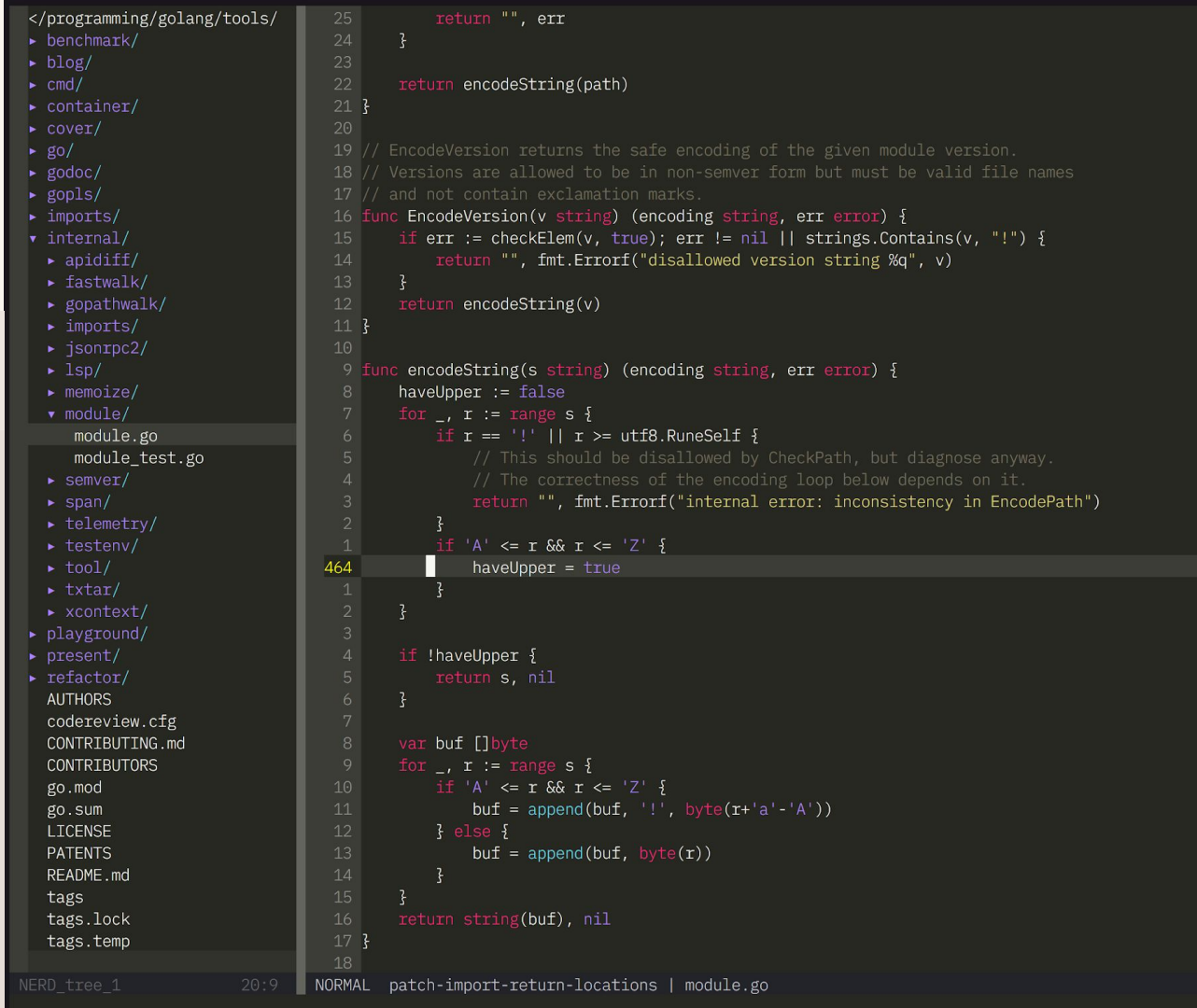
visual cues

“no thanks, i’d rather search my way through”

directory tree

netrw (:Explore)

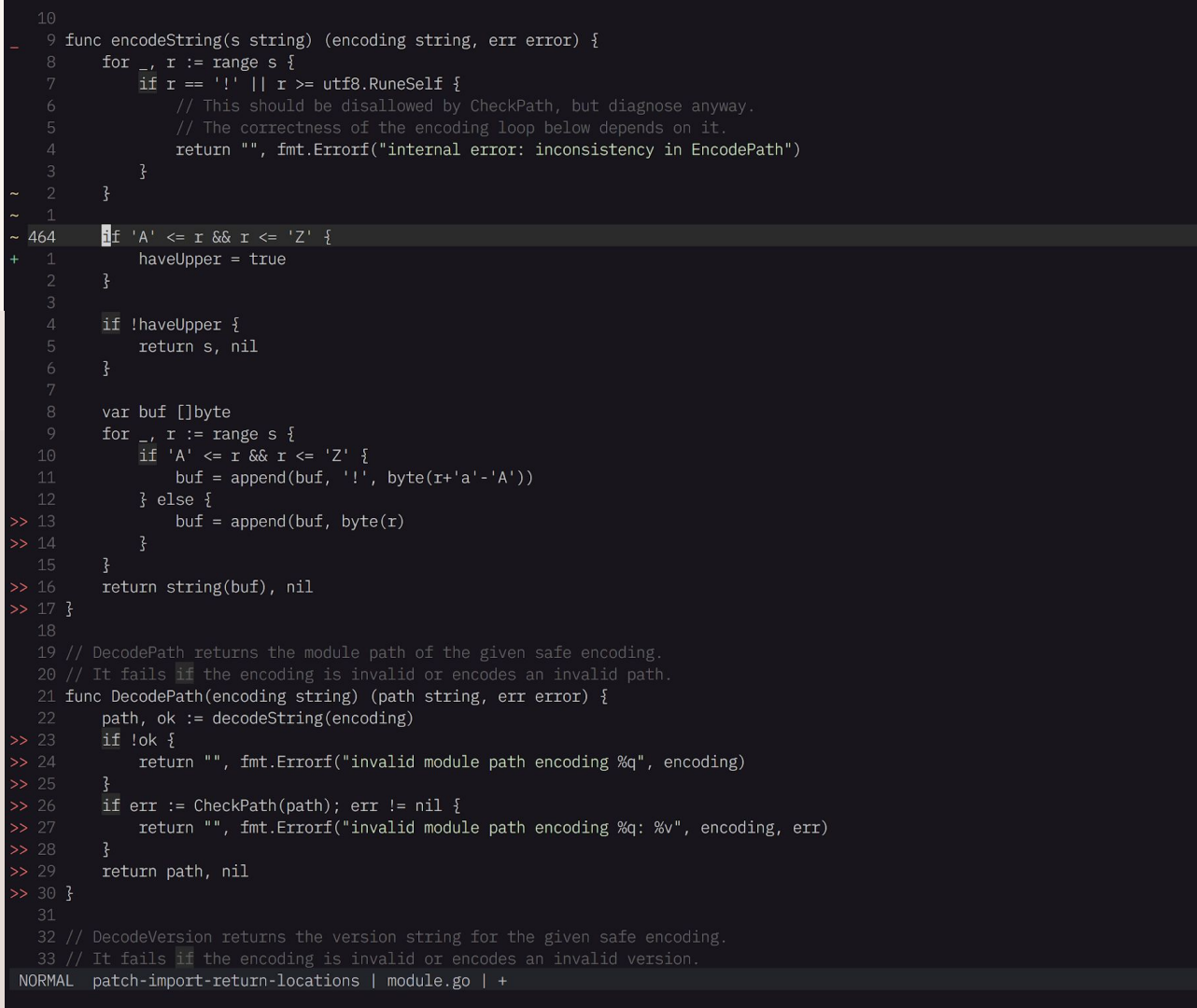
NerdTree



sign column

linter/formatter

git integration

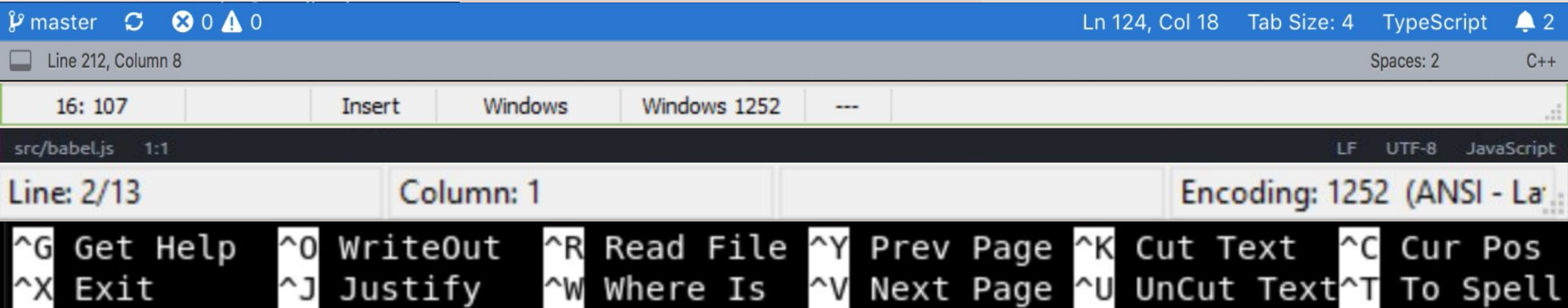


status line

branch info | filename | cursor row/column

language encoding | tabs/spaces | editing mode

linting errors | buffer identifier | filetype



status line

custom

lightline

airline

```
26
25 // EncodePath returns the safe encoding of the given module path.
24 // It fails if the module path is invalid.
23 func EncodePath(path string) (encoding string, err error) {
22     if err := CheckPath(path); err != nil {
21         return "", err
20     }
19
18     return encodeString(path)
17 }
16
15 // EncodeVersion returns the safe encoding of the given module version.
14 // Versions are allowed to be in non-semver form but must be valid file names
13 // and not contain exclamation marks.
12 func EncodeVersion(v string) (encoding string, err error) {
11     if err := checkElem(v, true); err != nil || strings.Contains(v, "!") {
10         return "", fmt.Errorf("disallowed version string %q", v)
 9     }
 8     return encodeString(v)
 7 }
 6
 5 func encodeString(s string) (encoding string, err error) {
 4     haveUpper := false
 3     for _, r := range s {
 2         if r == '!' || r >= utf8.RuneSelf {
 1             // This should be disallowed by CheckPath, but diagnose anyway.
460             // The correctness of the encoding loop below depends on it.
 1             return "", fmt.Errorf("internal error: inconsistency in EncodePath")
 2         }
 3         if 'A' <= r && r <= 'Z' {
 4             haveUpper = true
 5         }
 6     }
 7
 8     if !haveUpper {
 9         return s, nil
10     }
```

NORMAL patch-import-return-locations | module.go | + go 85% 460:8

additional

“but i’m good with fuzzy search”

code navigation

tags

ctags, exuberant ctags,
universal ctags

language server protocol

ale, vim-lsp, coc.nvim

search

fzf, ctrl-p

thank you
ありがとう

references

a. reference 1
b. reference 2