**Golang**

*[一.import](#_Toc12321)* [1](#_Toc12321)

[Golang 环境设置 1](#_Toc14569)

[操作符 2](#_Toc16439)

*[二．内建函数或GO基础包](#_Toc20554)* [3](#_Toc20554)

[保留函数 3](#_Toc19784)

*[三 golang GUI包/扩展库](#_Toc12573)* [3](#_Toc12573)

[1. walk Windows Application Library Kit 3](#_Toc3554)

[2. nwui 6](#_Toc21686)

[3. mahonia gbk->utf-8 编码工具 7](#_Toc18179)

*[四.数据库](#_Toc16685)* [7](#_Toc16685)

[1.mysql数据库链接包 7](#_Toc12079)

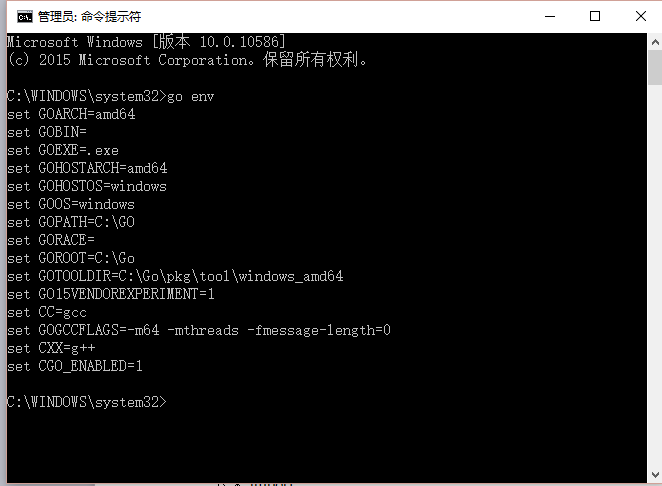
# *一.import*

## Golang 环境设置

通过>go env 可以查看go安装的配置信息

Goroot: go的安装路径,官方包路径根据这个设置自动匹配。如同golang的安装一致。

Gopath: 工作路径，如果设置错误可能是系统运行有问题。尤其是在这种情况下安装的第三方包，在引用是也会出错。我的经验是这种情况下，应该卸载掉当前的语言环境，重新安装。



绝对路径/相对路径 加载文件/包

1. 相对路径

import “./model” //当前文件同一目录的model目录，但是不建议这种方式来import

2. 绝对路径

import “shorturl/model”//加载gopath/src/shorturl/model模块

面展示了一些import常用的几种方式，但是还有一些特殊的import，让很多新手很费解，下面我们来一一讲解一下到底是怎么一回事

## 操作符

1. 点操作

我们有时候会看到如下的方式导入包

import(

. "fmt"

)

这个点操作的含义就是这个包导入之后在你调用这个包的函数时，你可以省略前缀的包名，也就是前面你调用的fmt.Println("helloworld")可以省略的写成Println("helloworld")

2. 别名操作

别名操作顾名思义我们可以把包命名成另一个我们用起来容易记忆的名字

import(

f "fmt"

)

别名操作的话调用包函数时前缀变成了我们的前缀，即f.Println("helloworld")

3. \_ 操作

这个操作经常是让很多人费解的一个操作符，请看下面这个import

import (

"database/sql"

\_ "github.com/ziutek/mymysql/godrv"

)

\_操作其实是引入该包，而不直接使用包里面的函数，而是调用了该包里面的init函数

注意：

同一个目录下不能定义不同的package

应该分为两个不同的目录才可以。

go，模块中要导出的函数，必须首字母大写。

# *二．内建函数或GO基础包*

go/src/builtin/ builtin.go 内建函数文档包，可直接使用而不用引入包

go调用C/C++，或dll文件

## 保留函数

无参数、返回值

main 只能应用于package main

init 能够应用于所有的package 可选

C/C++中接收命令行参数是通过~main函数传入，而go可以使用 os ,flag 库。 os.Args 命令行

# *三 golang GUI包/扩展库*

## 1. walk Windows Application Library Kit

地址： https://github.com/lxn/walk

原始go编写

安装出错：**go get github.com/lxn/walk** 。

其安装依赖于git包，而非node.js那样不依赖git。之前命令安装是一直报错时没有设置GOPATH包，也即是goroot、path等。下载[git](https://www.git-scm.com/)(https://www.git-scm.com/)软件

安装golang 时需要设置GO对应的环境变量，

$GOPATH- 设置错误时会出现问题/ 比如： import cycle not allowed

Golang 语言安装环境，设置错误时，可能无法正常工作

***说明>>***

a.安装walk

b.安装 rsrc 编译manifest

**go get github.com/akavel/rsrc** 安装rsrc命令

rsrc -manifest test.manifest -o rsrc.syso 编译

rsrc -manifest main.manifest –ico icon.ico -o rsrc.syso

exe文件加上自己喜欢的图标(网上)，未成功

rsrc -manifest gogui.manifest –ico ./img/gogui.ico -o rsrc.syso

test.manifest配置 注意在当前的目录下执行

c.build app

go build //可做调试编译

go build -ldflags="-H windowsgui" //编译后原来的cmd窗口将不存在

d.点击执行文件并可以

***引入包>>***

import (

"github.com/lxn/walk"

. "github.com/lxn/walk/declarative" //win 常用UI部件

)

***declarative>>声明***

* Action win动作 MenuItem 的子对象 action.go
* ActionList win动作列表 actionlist.go
* Application 应用 ，继承 Settings appliction.go
* Bitmap 位图文件 bitmap.go
* Icon 图标，Bitmap类，icon.go
* Image 图片，image.go
* ImageList 图标表，imagelist.go
* ImageView图像视图，imageview.go
* MainWindow 主视窗 mainwindow.go
* mapTableModel 表编辑模式，maptablemodel.go
* BoxLayout 盒子图层,boxlayout.go
* Menu 菜单，menu.go
* Form 表单，form.go表单
* Brush 刷新，brush.go
* Button 按钮，button.go
* Label 标签，label.go
* ListBox 列表框，listbox.go
* ComboBox win组合框，combobox.go
* GroupBox 单选框，groupbox.go
* NumberEdit 数字编辑器，numberedit.go
* ProgressBar 进度条，progressbar.go
* ProgressIndicator 进度条指示，progressindicator.go
* MsgBox 信息提示窗口，messagebox.go
* Composite 组合，composite.go
* FileDialog 文件对话框，commondialogs.go
* Dialog 对话框， dialog.go
* Layout(图层)→Container(容器)，container.go(容器)
* GridLayout(栅格化图层),gridlayout.go
* Condition 条件，使用在事件处理上，condition.go条件
* CancelEvent 取消事件，cancelevent.go
* CloseEvent 关闭事件，closeevent.go
* ErrorEvent 错误事件， errorevent.go
* IntEvent ，intevent.go
* KeyEvent 键盘事件，keyevent.go
* MouseEvent鼠标事件，mouseevent.go
* Event 事件，event.go
* Error 错误信息，error.go
* IniFileSettings ini文件设置，inifilesettings.go
* keyboard.go 键盘
* DropFilesEvent 文件拖动事件，dropfilesevent.go
* Canvas 画布，canvas.go
* CheckBox勾选框，checkbox.go
* CustomWidget 自定义部件，customwidget.go
* clipboard剪切板，clipboard.go
* Cursor 光标，cursor.go
* NotifyIcon 系统托盘，notifyicon.go
* Color win颜色，color.go
* Font 字体，font.go
* DataBinder 数据绑定器，databinding.go
* DateEdit 日期编辑，dateedit.go
* LineEdit线文件，lineedit.go
* LineErrorPresenter lineerrorpresenter.go
* Metafile 原文件/原图文件metafile.go
* ListModel 列表模式，models.go
* path.go 路径
* Pen 笔，pen.go
* Point 点，point.go
* property 性能/资源，property.go
* PushButton 按钮，pushbutton.go
* RadioButton 单选框按钮，radiobutton.go
* Rectangle 矩形，rectangle.go
* reflectModel 反射模式，reflectmodels.go
* RegistryKey 注册键，registry.go
* ScrollView 滚动条，scrollview.go
* Alignment 准线，simpletypes.go
* Size 尺寸，size.go
* Slider 滑块，slider.go
* Spacer 空格，spacer.go
* Splitter 语音分离器splitter.go
* splitterHandle 分离器句柄，splitterhandle.go
* splitterLayout 分离器图层，splitterlayout.go
* StatusBar 状态栏，statusbar.go
* TableView 表视图tableview.go
* TableViewColumn 表视图列，tableviewcolumn.go
* TableViewColumnList 表示图列列表tableviewcolumnlist.go
* TabPage 切换页，tabpage.go
* TabPageList 切换页列表，tabpagelist.go
* TabWidget 切换小部件tabwidget.go
* TextEdit 文字编辑框，textedit.go
* ToolBar 工具栏，toolbar.go
* ToolButton 工具按钮，toolbutton.go
* ToolTip 提示工具，tooltip.go
* TreeItemEvent 树项目事件，treeitemevent.go
* TreeView 树图，treeview.go
* 集合包util.go
* Validator 验证器，validators.go
* walk公用函数walk.go
* WebView 网页内核，webview.go
* webview\_dwebbrowserevents2.go
* webview\_idochostuihandler.go
* webview\_ioleclientsite.go
* webview\_ioleinplaceframe.go
* webview\_ioleinplacesite.go
* Widget 部件，widget.go
* WidgetList 部件列，widgetlist.go
* Window 对象，window.go

>> https://gowalker.org/github.com/lxn/walk

## 2. nwui

国产：https://github.com/go-nwui/nwui

采用HTML/css等基于node.js

## 3. mahonia gbk->utf-8 编码工具

安装命令 **go get github.com/axgle/mahonia**

github: axgle/mahonia

文档 <https://gowalker.org/github.com/axgle/mahonia>

# *四.数据库*

## 1.mysql数据库链接包

安装： **go get github.com/go-sql-driver/mysql**

官网： https://github.com/go-sql-driver/mysql