



Go program anatomy

# PROGRAM LAYOUT

main.go

```
package main
```

```
import (  
    ...  
)
```

```
const ...
```

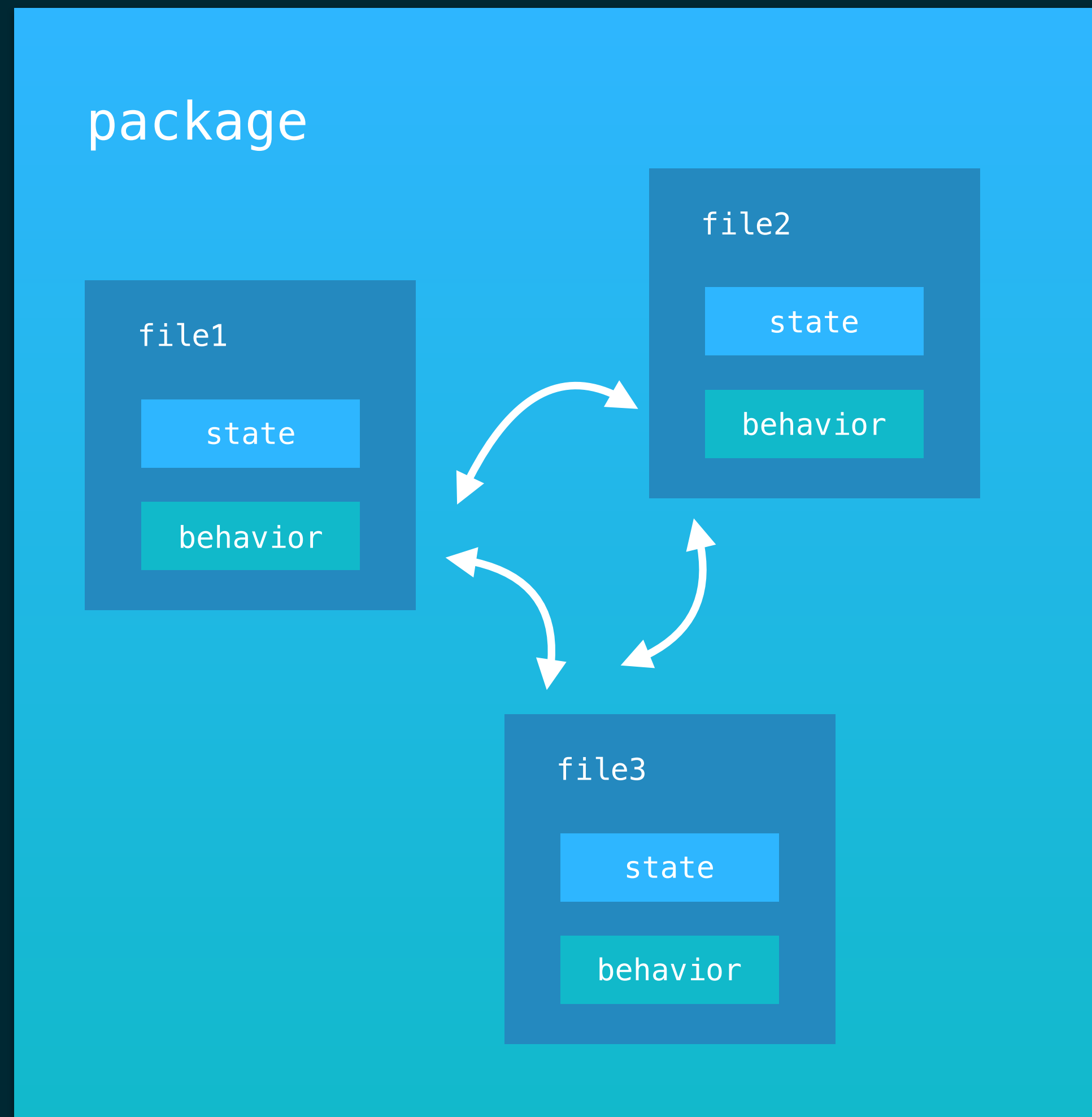
```
var ...
```

```
func init() {  
    // initialize  
}
```

```
func main() {  
    // execute  
}
```

\* REQUIRED

# PACKAGE ANATOMY

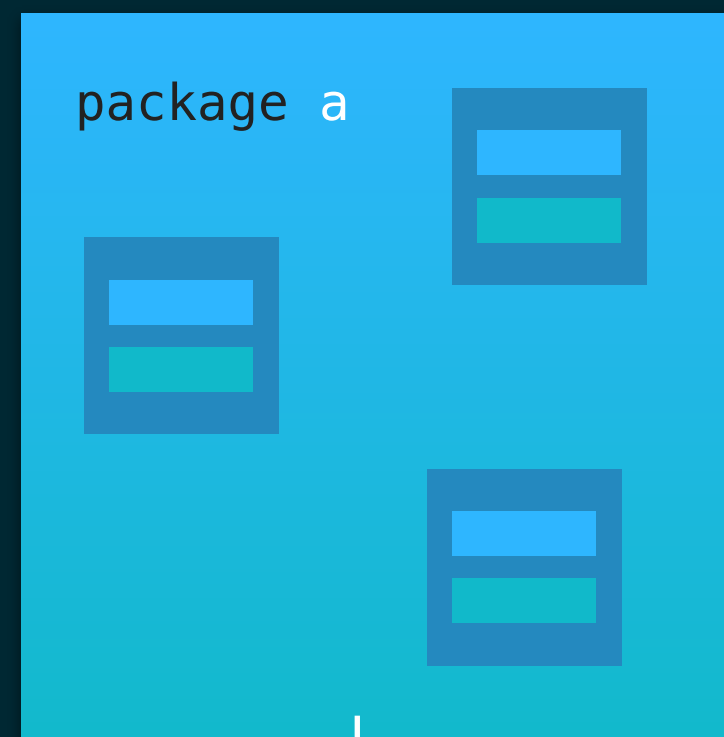


state

behavior

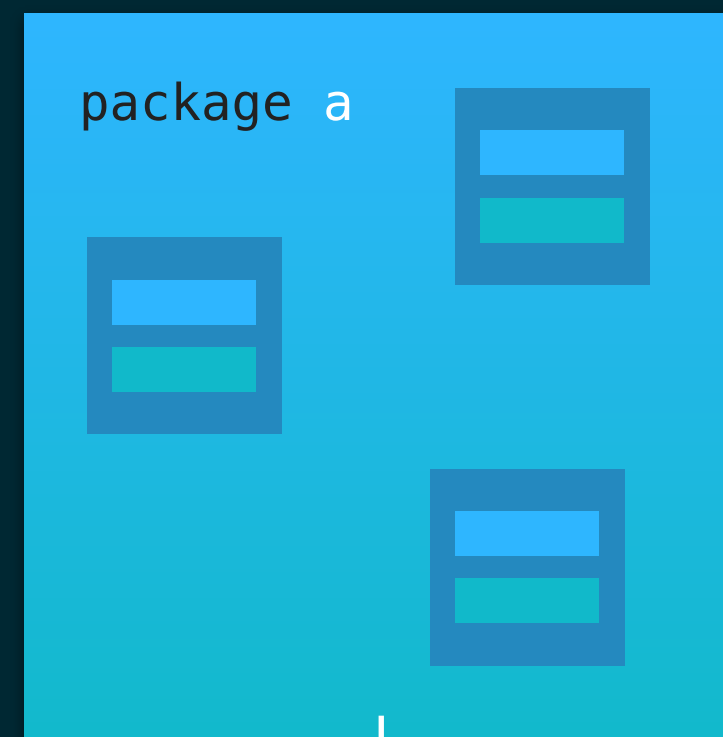
# FILE TYPES

controller.go



go build  
go run

controller\_test.go



go test

# GO PROGRAM



# PACKAGE TYPES

## executable (main)

```
go build -o exec
```



package main



exec

```
go run main.go
```



package main



SUCCESS

## un executable (non main)

```
go build -o exec
```



package other



exec

```
go run other.go
```



package other



FAIL

# PROGRAM LIFECYCLE

```
package main
```

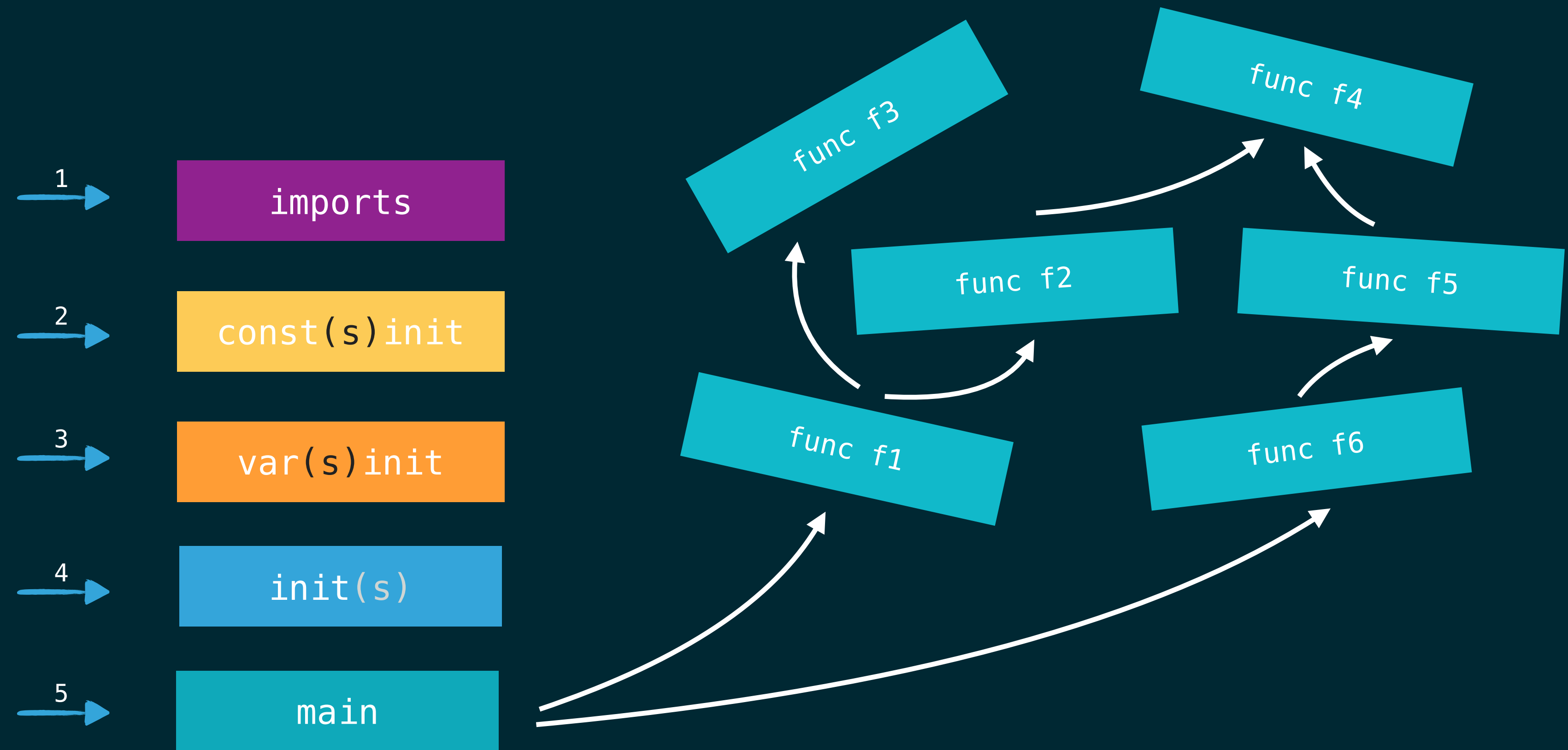
```
import (  
    ...  
)
```

```
const ...
```

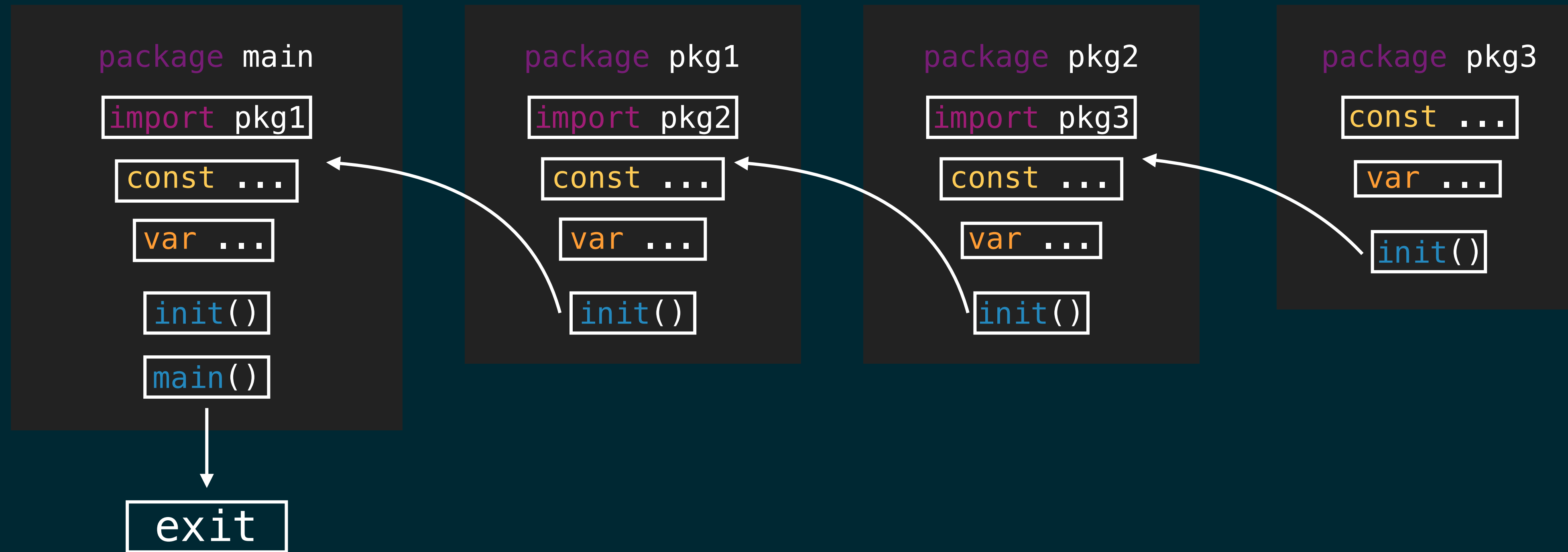
```
var ...
```

```
func init() {  
    // initialize  
}
```

```
func main() {  
    // execute  
}
```



# FLOW EXAMPLE





# GO BUILD / TAGS

2

Explicit \$GOOS & \$GOARCH

```
env GOOS=linux  
GOARCH=amd64
```

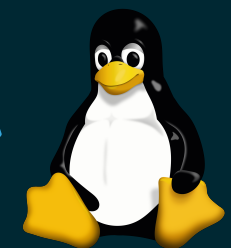
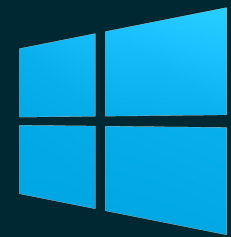
\$GOOS \$GOARCH

go build

platform-binary

1

Default \$GOOS & \$GOARCH



SUPPORTED ARCH

File name suffix

FILE\_GOOS.go

FILE\_GOARCH.go

FILE\_GOOS\_GOARCH.go

3

File names or tags

File tags

```
// +build os1[,arch1] [os2[,arch2]]
```

```
// +build !linux,!darwin !cgo
```

```
// +build ignore
```

```
// +build custom-tag
```

comma

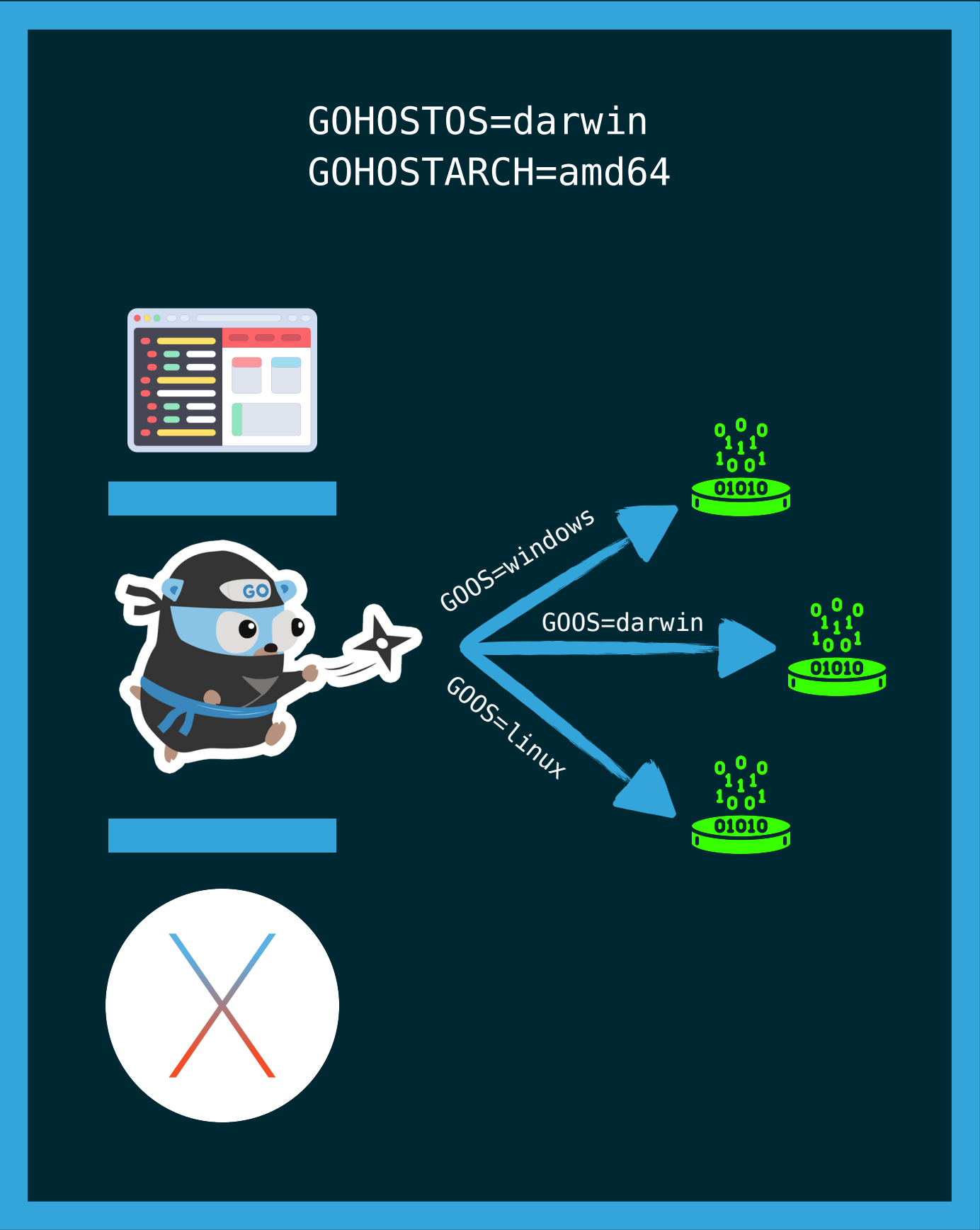
AND

space

OR

# BUILD/COMPILE TIME VS RUNTIME

## BUILD TIME



## RUN TIME

