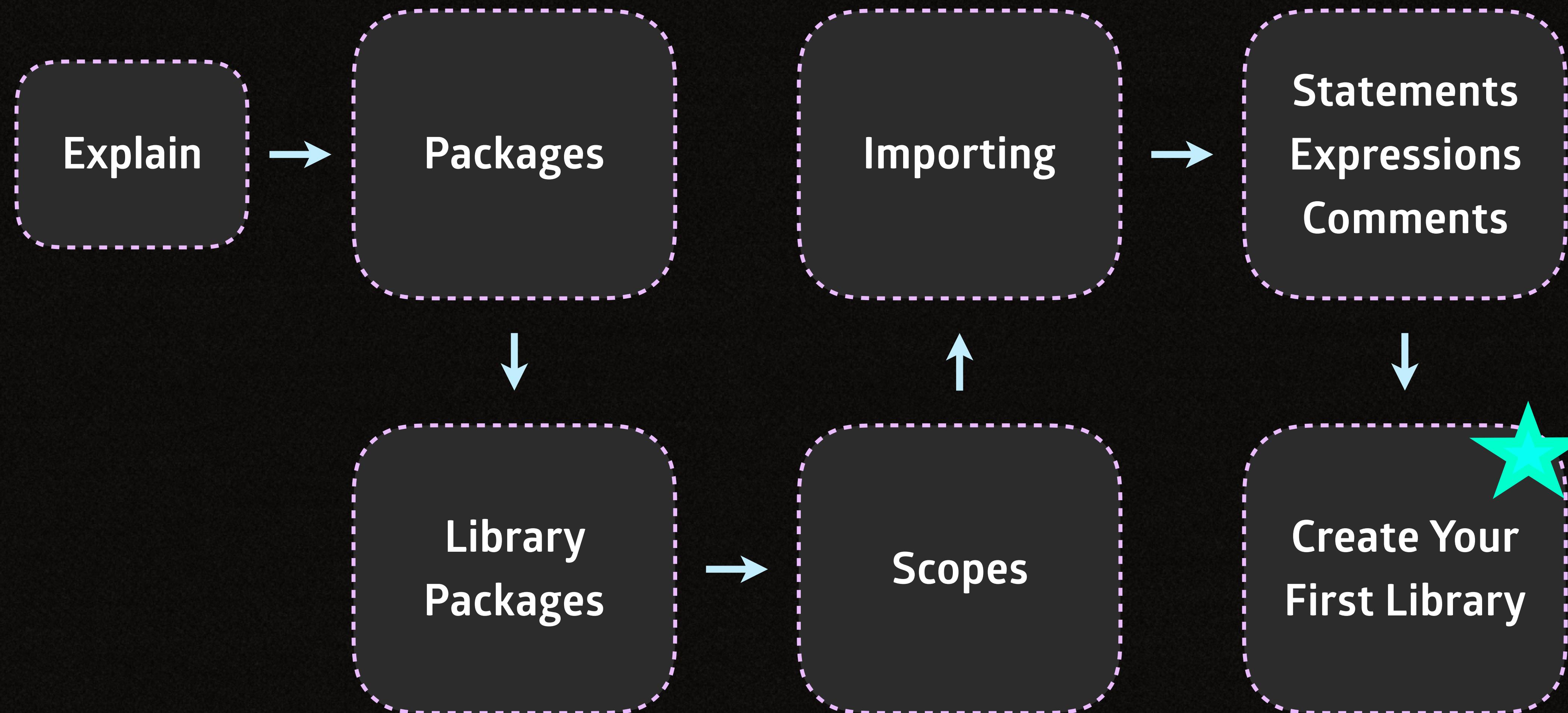


# EXPLAIN

*There are a lot of new knowledge awaits you!*



# PACKAGE

all **package files** should be in **the same directory**

example

`$GOPATH/src/github.com/username/learn.go/first/explain/packages/what`



# PACKAGE

all **files** should **belong** to **the same package**

**package main**

**package x**

**package y**

# PACKAGE CLAUSE

You can use **package clause** in a **file** to let Go know that which **package** that **file** belongs to.

**package clause** ← - - -

*this should be the first code*

*it can only appear once*

```
package main
import "fmt"

func main() {
    fmt.Println("Hello!")
}
```

# PACKAGES

*"main.go" file belongs to the main package.*

*main.go*

```
package main
import "fmt"

func main() {
    fmt.Println("Hello!")
}
```

*learn`go`/first/explain/packages/what*

# PACKAGES

"`bye.go`" file belongs to the `main` package.

`bye.go`

```
package main
import "fmt"

func bye() {
    fmt.Println("Bye!")
}
```

[learn.gofirst/explain/packages/what](https://learn.gofirst/explain/packages/what)

# PACKAGES

"`hey.go`" file belongs to the `main` package.

`hey.go`

```
package main
import "fmt"
func hey() {
    fmt.Println("Hey !")
}
```

[learn.gofirstexplain/packages/what](https://learn.gofirstexplain/packages/what)

# PACKAGES

All of the *files* belong to the *package main*. Each file has a *package clause* as "package main".

`learnigo/first/explain/packages/what`

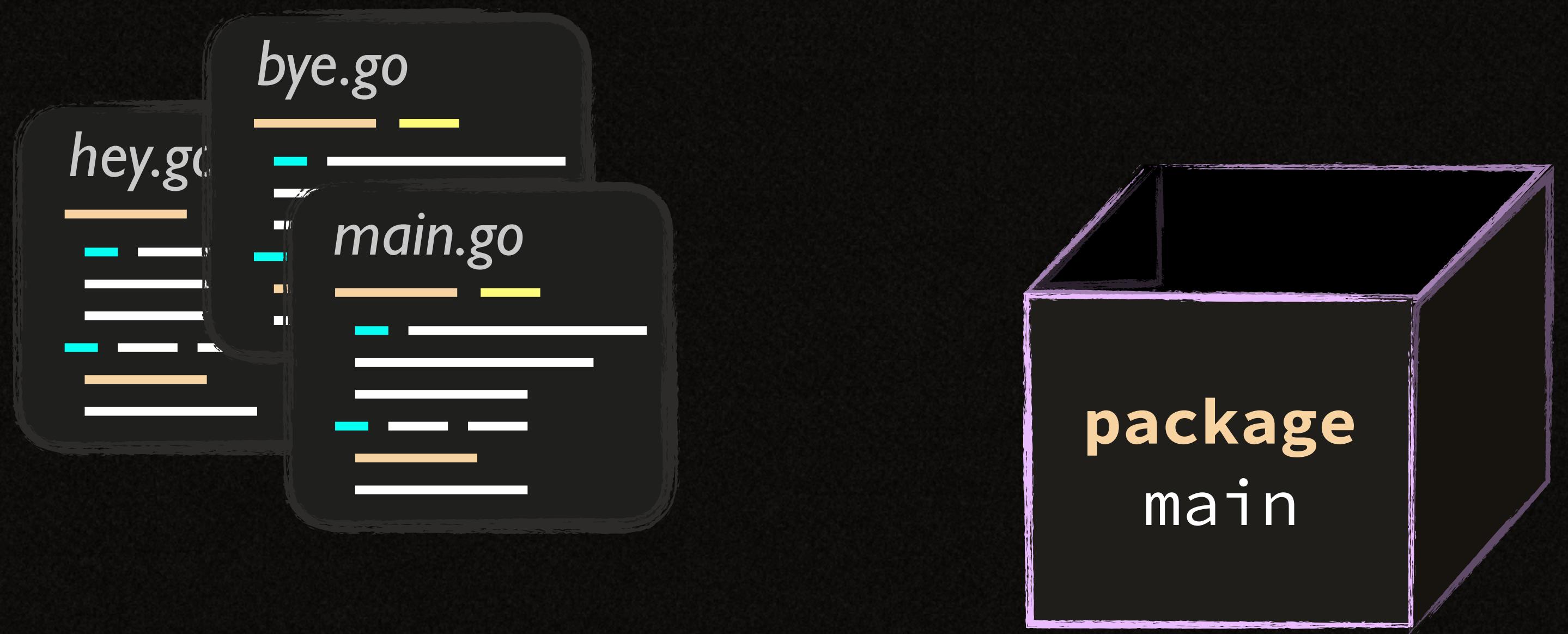


they all belong to  
**package main**

# PACKAGES

*Each Go package has its own **scope**.*

*For example, declared **funcs** are only **visible** to the files belong to the main package.*



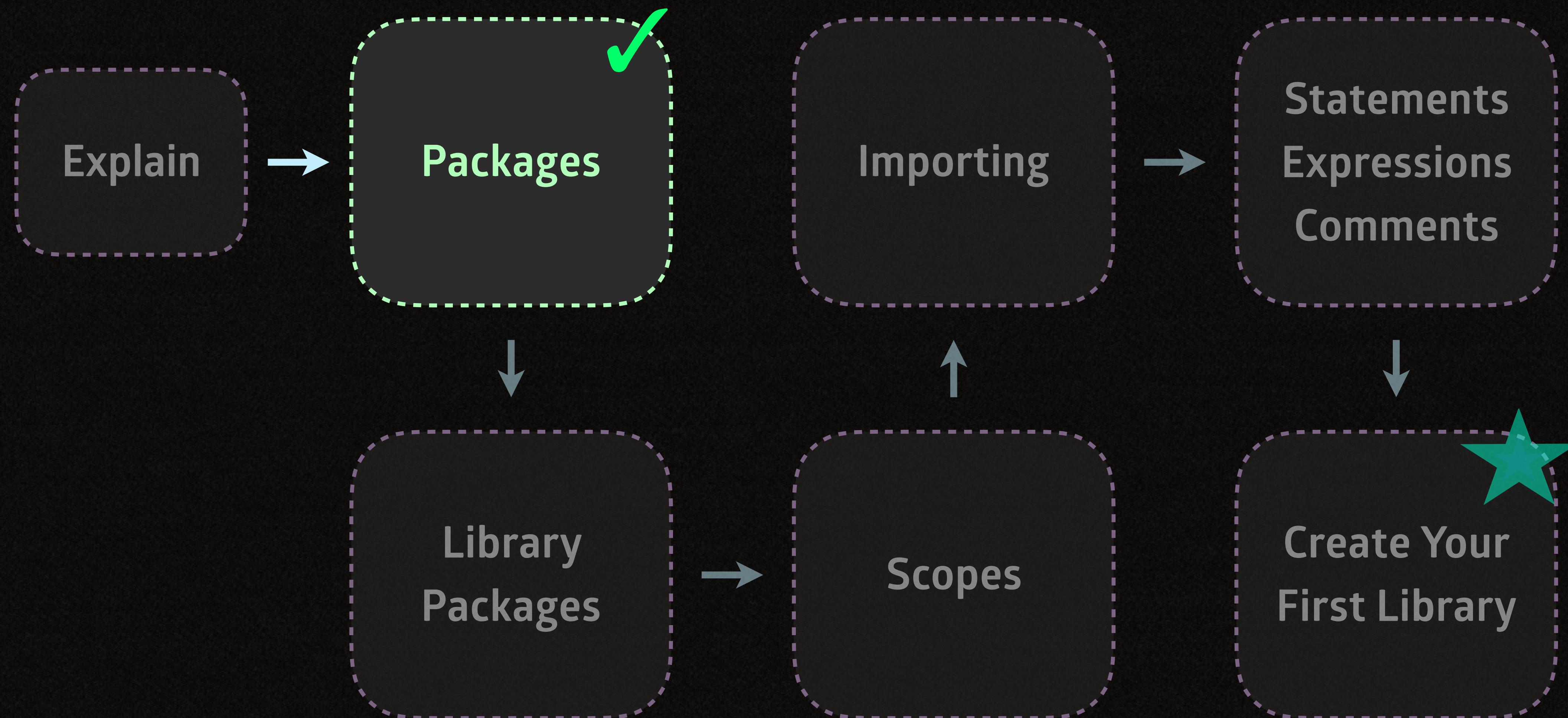
*all the files are inside the  
same directory*

[learn.go/fir.../explain/packages/what](https://learn.go/fir.../explain/packages/what)

*all the files belong to the  
same package*

# EXPLAIN

*Congrats! You've completed the first step.*



# EXECUTABLE

vs

# LIBRARY

# PACKAGE KINDS

*There are two kinds of packages in Go: **Executable** and **Library**.*

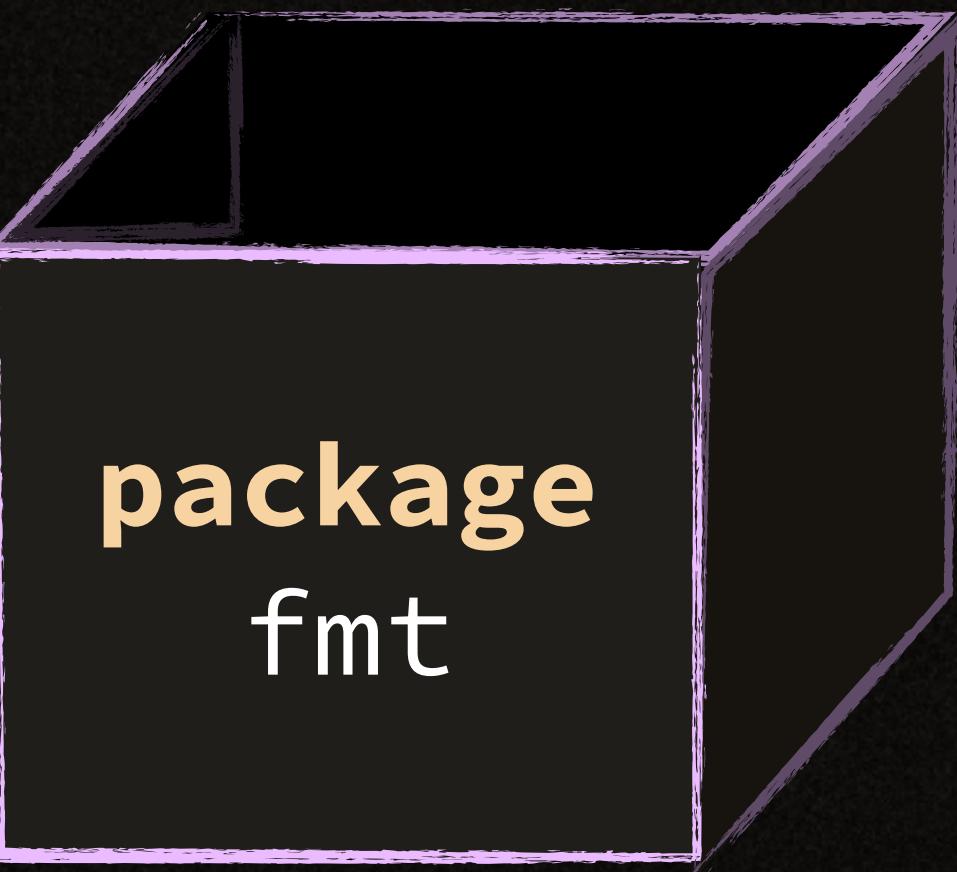
**Executable  
Package**



+

**func main()**

**Library  
Package**



# EXECUTABLE PACKAGE

an **executable go program** should belong to **package main**

**go build**

**go run**

# LIBRARY PACKAGE

created for reusability

# LIBRARY

created for **reusability**

non-executable

**importable**

can have **any name**

no **func main**

# EXECUTABLE

created for **running**

executable

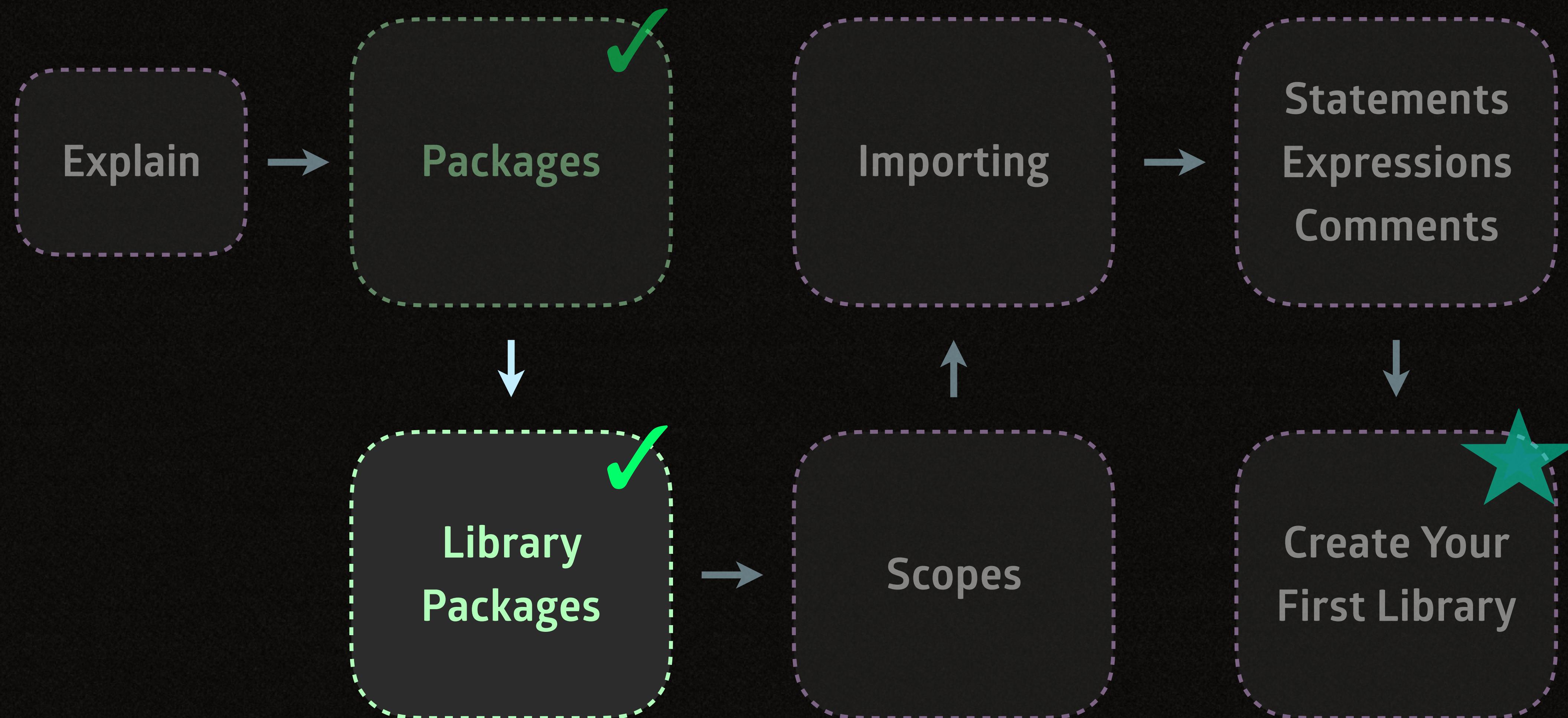
**non-importable**

name should be **main**

**func main**

# EXPLAIN

*Congrats! You've completed the second step.*



# SCOPE

**who can see what**

there are **package**, **file**, **func** and **block** scopes  
*there are a few more...*

# DECLARATIONS

**declares a unique name bound to a scope**

**same name** *cannot be declared again inside the same scope*

# SCOPE

*Every line of code can have **different scope** depending on their **position** in a Go file.*

**file scoped** ←---  
only visible in this file

**package scoped** ←-  
visible to all the files  
belong to the package

*other packages can't  
see them*

**package main**

**import "fmt"**

**const ok = true**

**func main() {**

**var hello = "Hello!"**  
**fmt.Println(hello, ok)**

**}**

→ **block scoped**  
declaration  
only visible  
after its declaration  
until " } "

# SCOPE

*Every line of code can have **different scope** depending on their **position** in a Go file.*

```
package main
import "fmt"

func nope() {
    const ok = true
    var hello = "Hello!"
    _ = hello
}

func main() {
    fmt.Println(hello, ok)
}
```

→ **block scoped declaration**  
only visible  
in "nope" func

# PACKAGE SCOPE

**names are visible throughout the package**

# PACKAGE SCOPE

*Each Go package has its own **scope**. For example, declared **funcs** are only **visible** to the files belonging to the same package.*

```
package main
import "fmt"

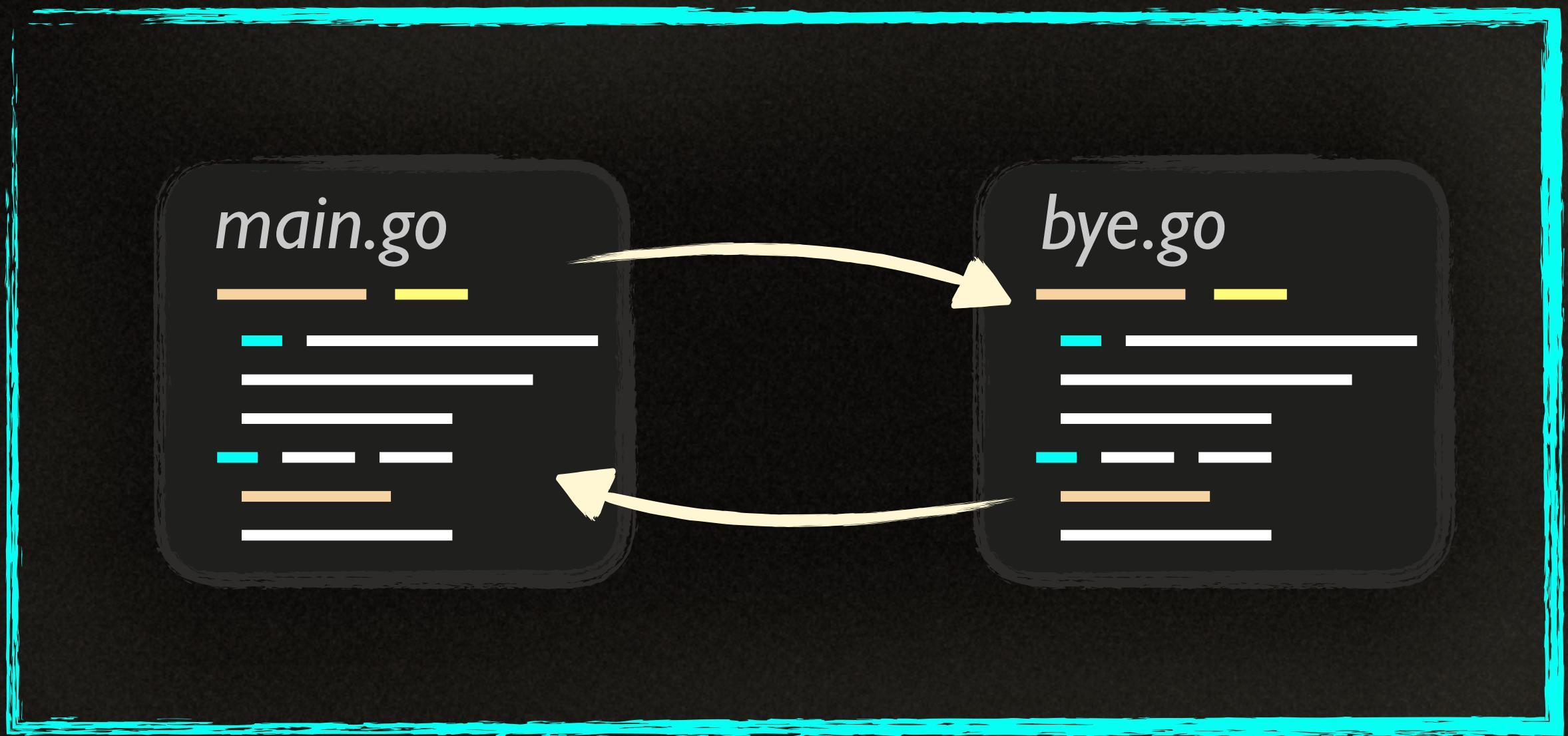
func main() {
    fmt.Println("Hello!")
}
```

**"main()** is visible  
throughout ← - -  
the *main* package

# PACKAGE SCOPE

*Declarations which are outside of functions are visible to the files belong to the same package.*

**main package**



# PACKAGE SCOPE

*Each Go package has its own scope*

*For example, declared **funcs** are **visible** in the same package*

**main.go**

```
package main
import "fmt"

func main() {
    fmt.Println("Hello!")
    bye()
}
```

**bye.go**

```
package main
import "fmt"

func bye() {
    fmt.Println("Bye!")
}
```

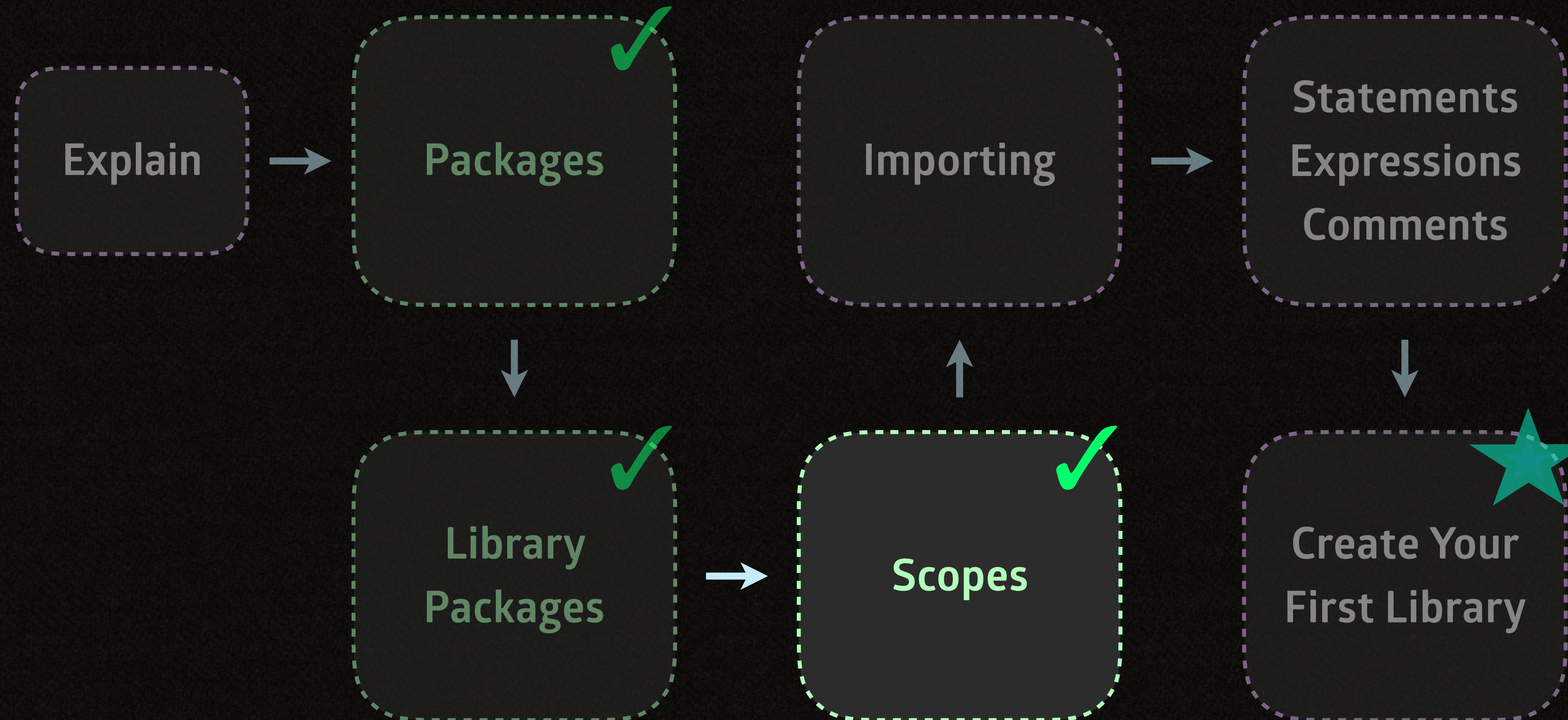


# PACKAGE SCOPE #2

**declaring the same names in the same scope**

# EXPLAIN

*Congrats! You've completed the third step.*



# IMPORTING #1

allows a **file** to use **functionalities** from a **library package**

# IMPORTING

*Importing is like as if you've declared what's inside the imported package's files in your own file.*

*myfile.go*

```
import "fmt"  
import "errors"  
import "time"
```

*format.go  
print.go*

**package**  
fmt

*errors.go*

**package**  
errors

*bye.go  
hey.go*

**package**  
your

*time.go*

**package**  
time

# FILE SCOPE

**names are visible throughout the file**

# FILE SCOPE

*Each Go file has its own scope*

*Imported packages are only visible to the importing file*

**main.go**

"**fmt**" is visible  
throughout  
the file

```
package main
import "fmt"

func main() {
    fmt.Println("Hello!")
}
```

# FILE SCOPE

*Each Go file has its own scope*

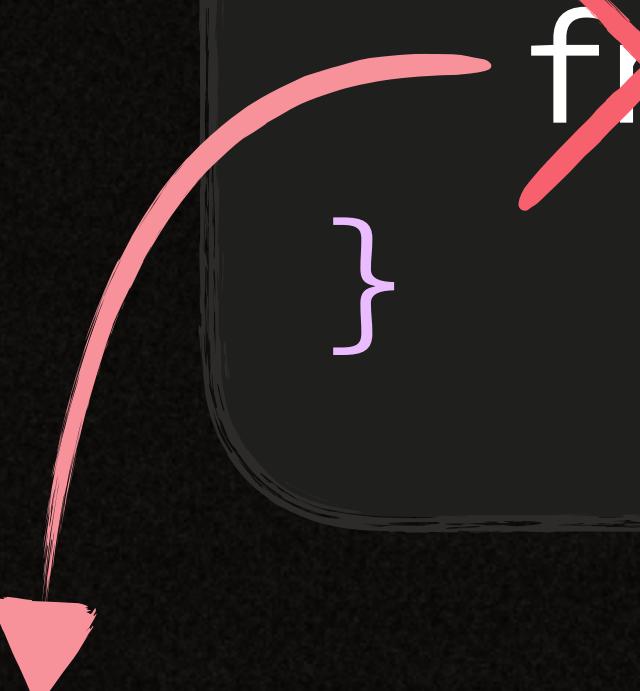
*Imported packages are only visible to the importing file*

**bye.go**

```
package main
```

```
func bye() {  
    fmt.Println("Bye!")
```

```
}
```



bye.go can't use a package that it didn't import

# FILE SCOPE

*Each Go file has its own scope*

*Imported packages are only visible to the importing file*

```
package main
import "fmt"

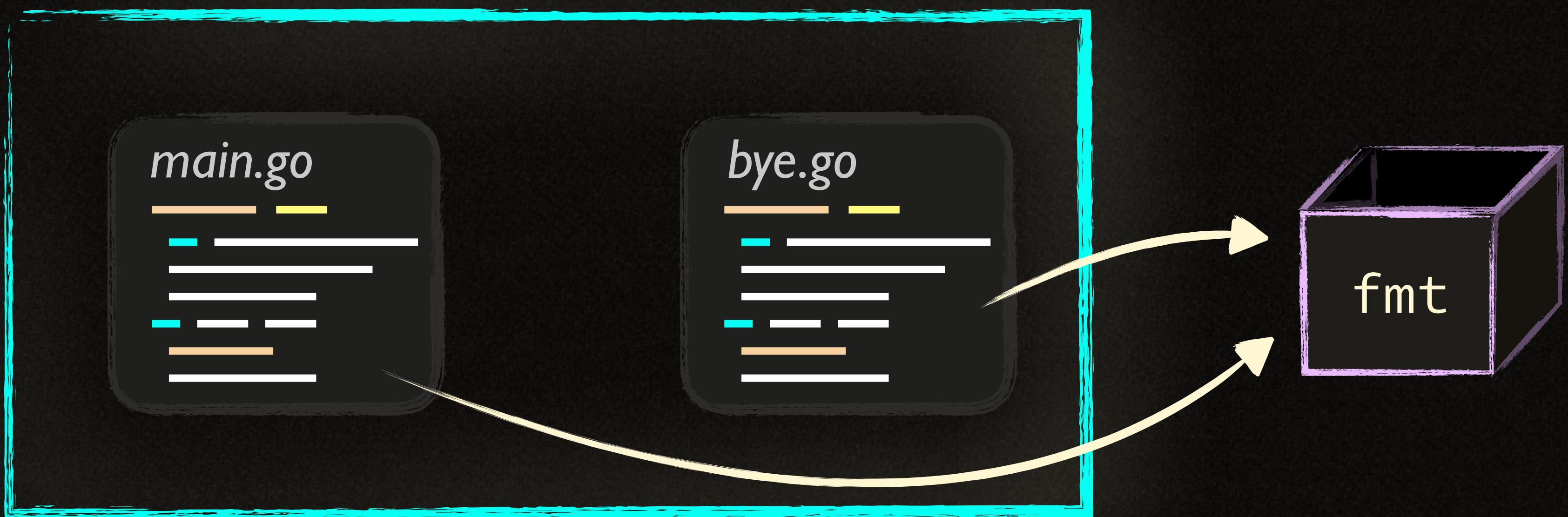
func bye() {
    fmt.Println("Bye!")
}
```



# FILE SCOPE

*Each file has to import external packages on its own*

**main package**



*learngolang/first/explain/importing*

# IMPORTING #2

you can **rename** the name of a declared package

# RENAMING IMPORTS

*Multiple import declarations should be unique (in the same Go file).*

**import declaration**  
this should be after  
the package clause

```
package main
import "fmt"
import "fmt" // multiple imports are not allowed
func main() {
    fmt.Println("Hello!")
}
```

you can *import*  
*multiple packages*  
but they all should  
be *unique*

*within the same scope, any new declarations require unique names*

# RENAMING IMPORTS

You can import packages with the same name into the same file  
by giving one of them another name

```
package main
import "fmt"
import f "fmt" ----->
func main() {
    fmt.Println("Hello!")
    f.Println("There!")
}
```

now you can use  
fmt package  
using these names:

fmt  
or  
f

# EXPLAIN

*Congrats! You've completed the fourth step.*

