

STATEMENTS

instructs Go to execute something

one statement at a time
(unless you use semicolons ;)

STATEMENTS

*Statements tell Go to **execute** something
and they can **change** the **execution flow** of code*

```
package main
import "fmt"

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

STATEMENTS

*Almost all the code are **statements** here
They can stand on their own on a single line of code*

```
package main
import "fmt"

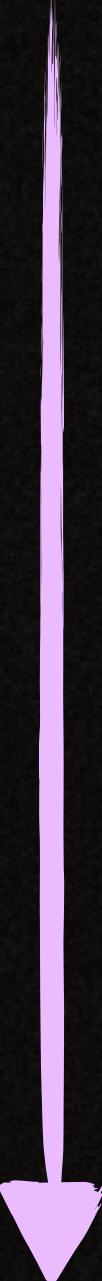
func main() {
    fmt.Println("Hello!")
}
```

STATEMENTS

Statements control the execution flow

```
package main
import "fmt"

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



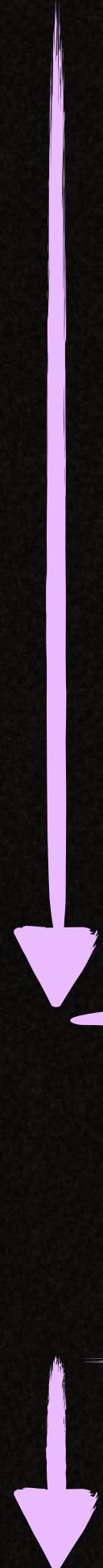
STATEMENTS

Statements control the execution flow. Look at how if statement changes the execution.

```
package main
import "fmt"

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

    if 5 > 1 {
        fmt.Println("bigger")
    }
}
```



STATEMENTS

Now, there are two *statements*: both are `fmt.Println` function calls.

```
package main
import "fmt"

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

STATEMENTS

*Go adds a **semicolon** between statements, behind the scenes.*

```
package main
import "fmt"

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



There are still two statements here

EXPRESSIONS

computes one or multiple values

expressions should be used **with** or **within** a *statement*

some statements like func calls can also act like expressions

EXPRESSIONS

Expressions calculate (evaluate) and return one or more values (multiple).

```
package main
import "fmt"

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

EXPRESSIONS

*There's **only one expression here: "Hello!" string literal.***

```
package main
import "fmt"

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

EXPRESSIONS

*Operators allow you to **combine** expressions.*

```
package main
import "fmt"

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

"Hello!!"

EXPRESSIONS

*Functions can be used as **expressions** inside statements.*

```
package main
import (
    "fmt"
    "runtime"
)

func main() {
    fmt.Println(runtime.NumCPU())
}
```



EXPRESSIONS

You can use *operators* with function *call expressions*.

```
package main
import (
    "fmt"
    "runtime"
)

func main() {
    fmt.Println(runtime.NumCPU() + 1)
}
```

EXPRESSIONS

*You can use **operators** with function **call expressions**.*

```
package main
import (
    "fmt"
    "runtime"
)

func main() {
    fmt.Println(runtime.NumCPU() + 1)
}
```

COMMENTS

documentation

```
// single line comments
```

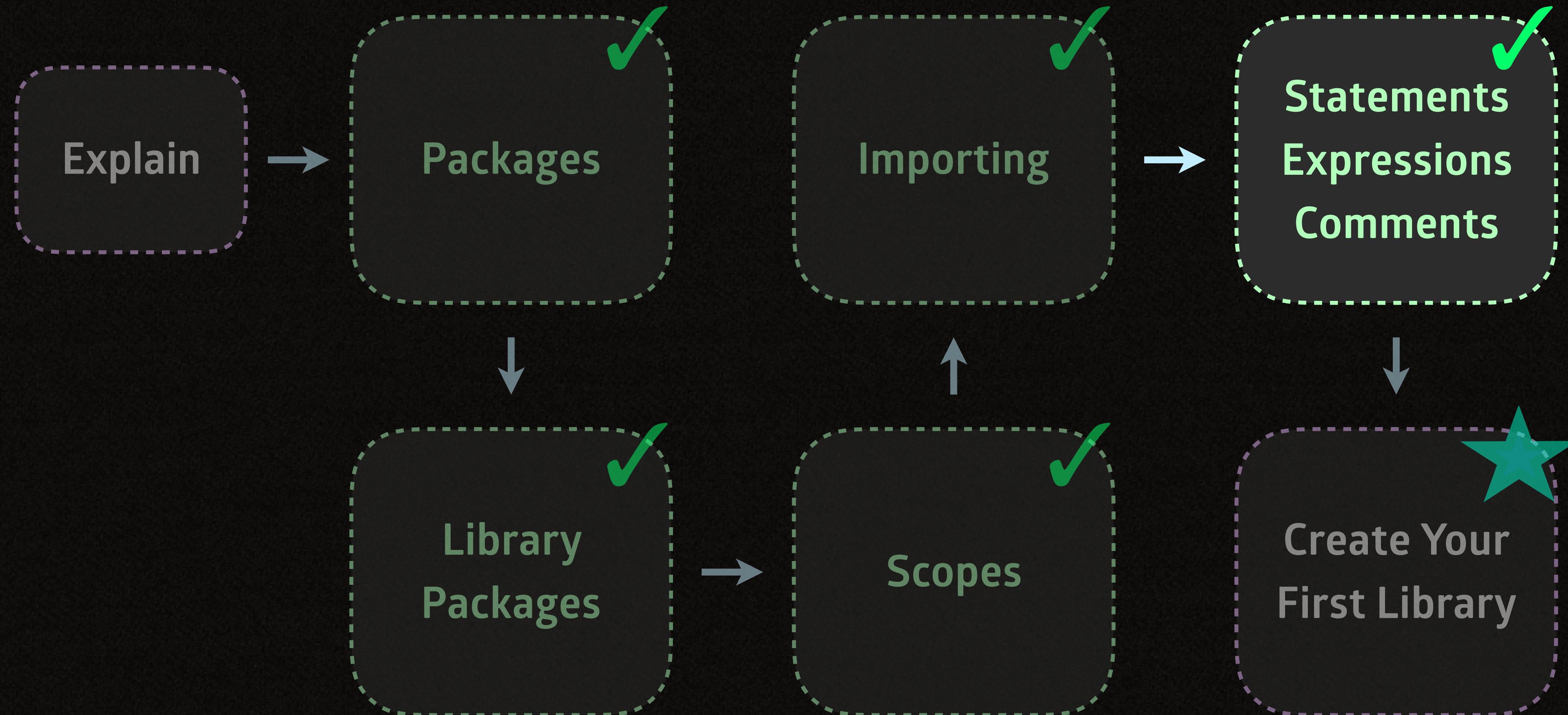
```
/*
    multi line comments
    more than single line
*/
```

GO DOC

creates documentation automatically

EXPLAIN

Congrats! You've completed the fifth step.



CREATE YOUR OWN
LIBRARY PACKAGE

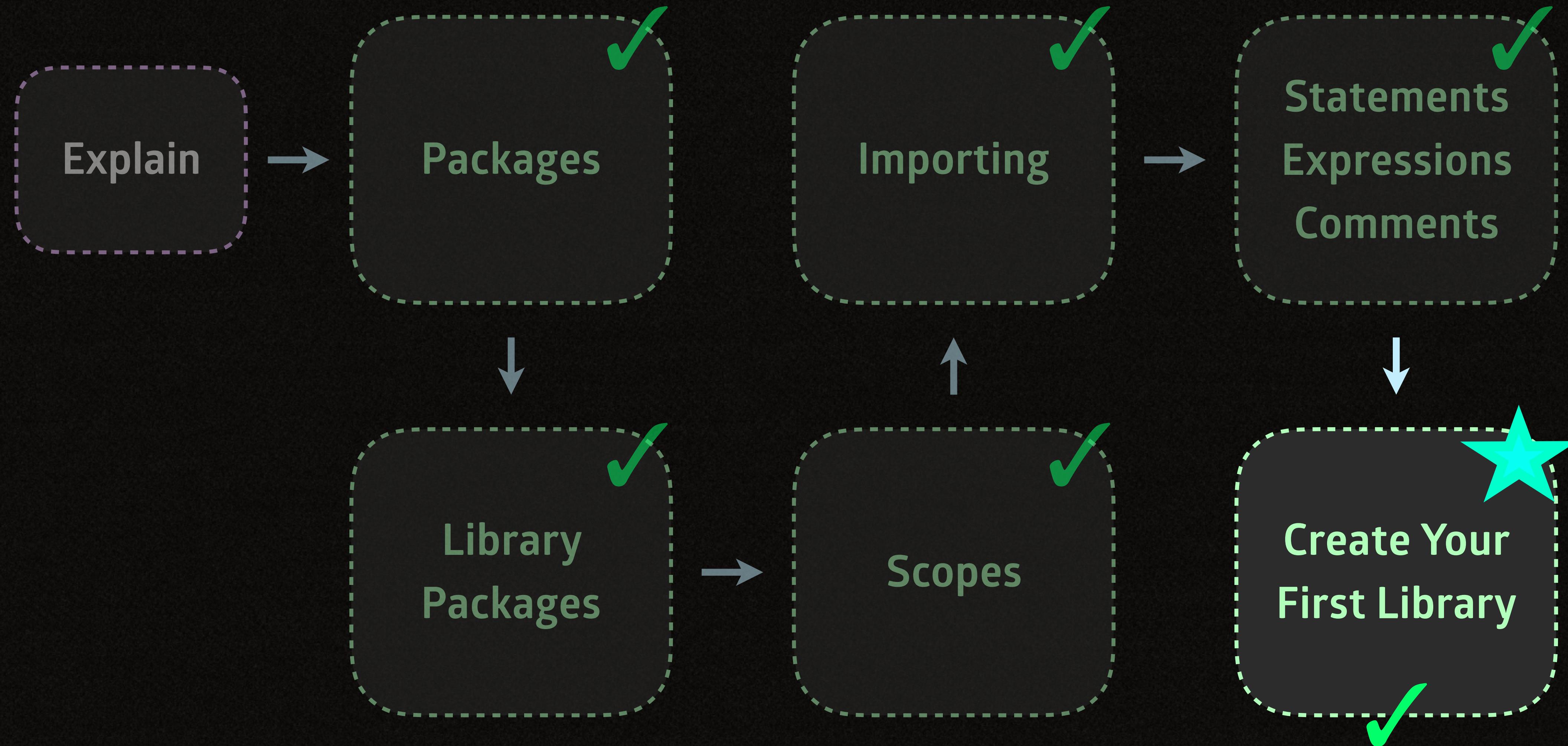
EXPORTING

allows a **package** to make its **functionalities available** to **other packages**

to **export** a name just make its **first letter** an **uppercase letter**

EXPLAIN

Congrats! You've completed the fifth step.



DONE DONE DONE

Congrats! Now you're ready to be a *Gopher*! You're awesome!

