

Packages & Modules

01

About Packages

02

Using Packages

03

Modules

About Packages

- | Packages are Go's way of organizing code
- | Programs are written as one or more packages
 - | Packages can be imported from the Go package registry
- | Packages should be focused and perform a single thing
 - | Argument parsing
 - | Drawing graphics
 - | Handling HTTP requests

Using Packages

```
import "name"
```

```
import (  
    "name"  
    "namespace/packageName"  
)
```


Using Packages

- | Can import everything using a dot (.)
 - | No need to reference package name in code
- | Imports can be renamed

```
import (  
    . "name"  
    pk "namespace/packageName"  
)
```

Modules

- | **Modules** are a collection of packages
- | Created by having a **go.mod** file in the root directory of your project
 - | Can be managed by the Go CLI
- | Contains information about your project
 - | Dependencies, Go version, package info
- | All Go projects have a **go.mod** file

Example Module

```
module example.com/practice
```

```
go 1.17
```

```
require (  
    github.com/alexflint/go-arg v1.4.2  
    github.com/fatih/color v1.13.0  
)
```

Recap

- | **Packages** are Go's way of organizing code
- | Packages should be a single program or single-purpose library
- | Packages can be published to and imported from the Go package registry
- | **Modules** are a collection of packages
 - | `go.mod`
 - | Can be managed by the Go CLI