Variables & Values

Basically all operations use values

e.g. 2 + 5 performs the “add” operation on the values 2 and 5

Variables are “**data containers**” for values

- Allow for value re-usage
- Can store (intermediate) results
- Can be used in operations and re-assigned

Go is a statically typed language

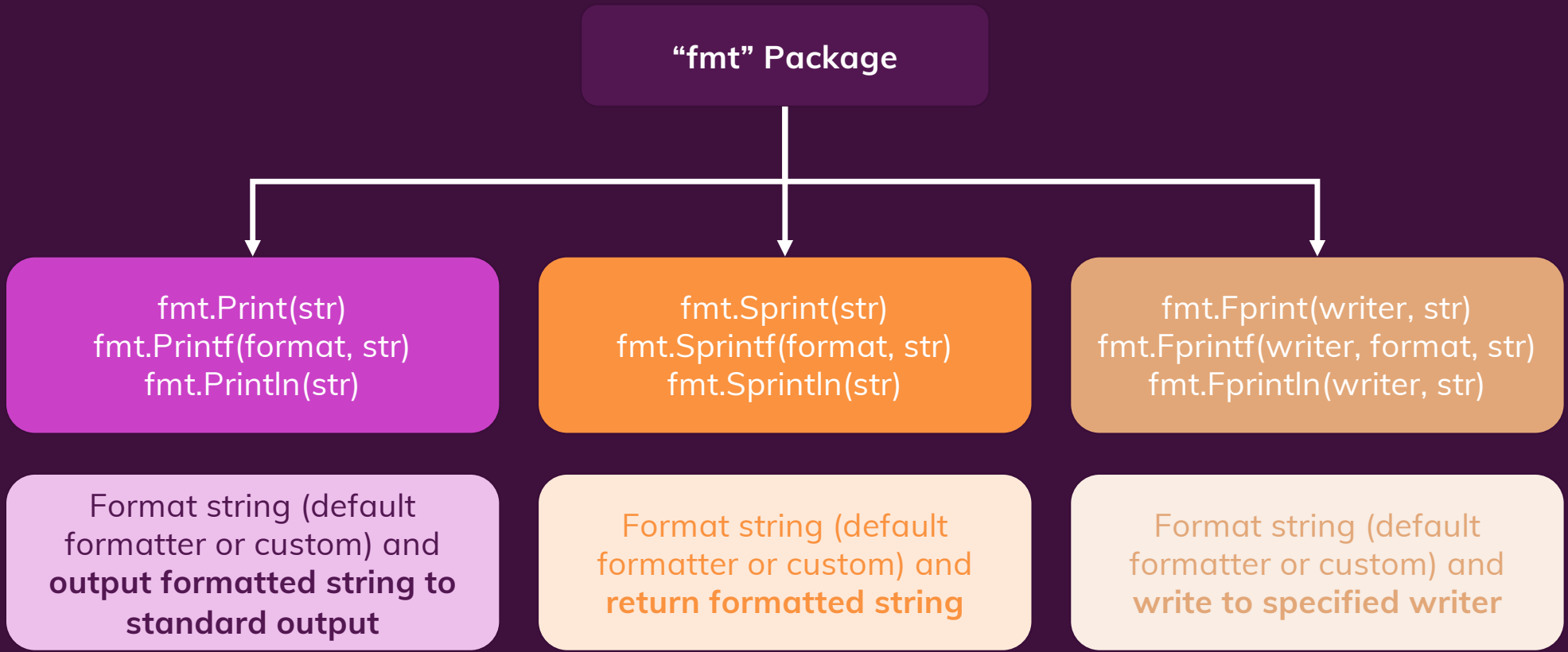
Key Value Types

string	int	float64	bool
Text	Number without decimal places	Number with decimal places	Yes / No
”Hello there!” “Why?”	-10 151	-8.151 99.99	true false

rune	byte
Single unicode character	Single byte (ASCII character)
‘a’ ‘❤️’	‘a’ ‘9’

The standard library is a "built-in module" full of core packages and functionalities

Formatting & Outputting Strings



Packages & Modules

Go code is organized into modules and packages

Modules	Packages
A module has a unique identifier and can be distributed (e.g. library)	Every module contains at least one package (the “main” package)
Every Go project goes into a new module	Multiple files can make up the same package (via `package` instruction)
Projects can use (i.e. import from) multiple modules (custom and third-party)	A module may contain multiple packages, stored in subfolders
Created and managed via `go mod` commands and `go.mod` file	Can be imported

package main

Executable programs must contain a “package main”

Inside the “main” package, Go looks for a “main” function

It’s the **main entry point** of the program

Library packages may also use other names than “main”