Text Formatting: fmt

01

Verbs

02

Escape Sequences

03

Printf / Sprintf / Fprintf

fmt Package

- **fmt** package provides terminal printing and string formatting
- Provides functions:
  - **Printf** - custom format
  - **Print** - simple print
  - **Println** - simple print with a newline
- **F** and **S** variants of the above functions:
  - **F** prints to a data stream: **Fprintf, Fprint, Fprintln**
  - **S** prints to a new string: **Sprintf, Sprint, Sprintln**

Printf

- **Printf** uses **verbs** to describe how something should print

Verb	Description
%v	default
%t	"true" or "false"
%c	Character
%X	Hex
%U	Unicode format
%e	Scientific notation

Escape Sequences

- **Escape sequences** allow insertion of special characters in strings

Escape Sequence	Description
\\	Backslash
\'	Single quote
\"	Double quote
\n	Newline
\u or \U	Unicode (2byte & 4byte)
\x	Raw bytes (as hex digits)

Example: Printf

```go
fmt.Printf("Hello, world!\n")
fmt.Printf("%v, %v!\n", "Hello", "world")
fmt.Printf("This is a \"Quote\"\n")
```

Example: Sprintf

```go
func surround(msg string, left, right rune) string {
	return fmt.Sprintf("%c%v%c", left, msg, right)
}

surrounded := surround("this message", '(', ')')
fmt.Println(surrounded)
// (this message)
```

Recap

- **Printf** uses **verbs** to format and print data
  - **Sprintf** prints to a new **string** instead of the terminal
  - **Fprintf** prints to a data stream instead of the terminal
- **Escape sequences** can be used to print special characters