Fmt Package

Created	@Jan 27, 2021 4:32 PM
Tags	Lesson

Fmt Package

- `fmt` helps us *format* data - which is why it's referred to as "format" though technically it's pronounced "fumpt"
- In addition `Println()` there is also...
    - `Print()`
    - `Printf()`
    - `Sprint()`
    - `Sprintln()`
    - `Sprintf()`
    - etc...

Printing

- `fmt.Println()` prints its arguments (data provided within its parentheses `()`) with an included space in between each argument and adds a line break at the end.
    - Code:

      ```
      fmt.Println("Println", "formats", "really well")
      fmt.Println("Right?")
      ```

    - Output:

      ```
      Println formats really well
      Right?
      ```

- `fmt.Print()` does not include the spaces or line break

  - Code:

    ```
    fmt.Print("The answer is", ": ")
    fmt.Print("12")
    ```

  - Output:

    ```
    The answer is: 12
    ```

- `fmt.Printf()` is used to *interpolate* strings
  - `%v` is a placeholder known as a *verb* in Go
  - Code:

    ```
    guess := "C"
    fmt.Printf("Is %v your final answer?", guess)
    ```

  - Output:

    ```
    Is C your final answer?
    ```

Different Verbs

- In addition to `%v` Go has a variety of helpful verbs
  - `%T` prints the type of the argument
  - `%d` interpolates a number into a string
  - `%f` interpolates a float into a string
  - `%.2f` limits the precision to 2 digits

Sprint and Sprintln

- `fmt.Sprint()` and `fmt.Sprintln()` are used to format strings, but do not output anything
- Code:

  ```
  step1 := "Breathe in..."
  step2 := "Breathe out..."
  meditation := fmt.Sprintln(step1, step2)
  ```

- Value of `meditation`:

  ```
  Breathe in... Breathe out...
  ```

Sprintf

- like `fmt.Printf()`, `fmt.Sprintf()`can use verbs
- Code:

  `correctAns := "A"`

  `answer := fmt.Sprintf("And the correct answer is… %v!", correctAns)`
- Value of `answer`:

  `And the correct answer is… A!`

Getting User Input

- `fmt.Scan()` allows us to get user input

```
fmt.Println("How are you doing?")

var response string
fmt.Scan(&response)

fmt.Printf("I'm %v.", response)
```

Terminal:

```
How are you doing?
good
I'm good.
```