

# Road to Mercari Gopher Dojo module 02

Summary: This document is the subject for the Gopher Dojo module 02 of the Road to Mercari @ 42 Tokyo.





# Contents

Ι	Instructions	2
II	Foreword	8
III	Exercise 00: Typing game	4
IV	Exercise 01 : Split Downloader	6

### Chapter I

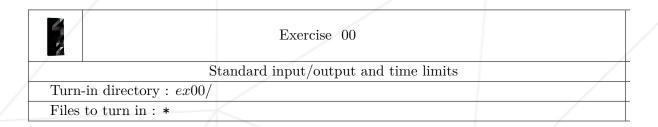
### Instructions

- Only this page will serve as reference; do not trust rumors.
- Watch out! This document could potentially change up to an hour before submission.
- These exercises are carefully laid out by order of difficulty from easiest to hardest. We will not take into account a successfully completed harder exercise if an easier one is not perfectly functional.
- Make sure you have the appropriate permissions on your files and directories.
- You have to follow the submission procedures for every exercise.
- Your exercises will be checked and graded by your fellow classmates.
- You <u>cannot</u> leave <u>any</u> additional file in your directory than those specified in the subject.
- Got a question? Ask your peer on the right. Otherwise, try your peer on the left.
- Your reference guide is called Google / man / the Internet / ....
- Examine the examples thoroughly. They could very well call for details that are not explicitly mentioned in the subject...
- If no other explicit information is displayed, you must use the latest versions of Go.

# Chapter II Foreword https://gopherdojo.org/ 3

### Chapter III

### Exercise 00: Typing game



Create a game using standard input/output that meets the following specifications.

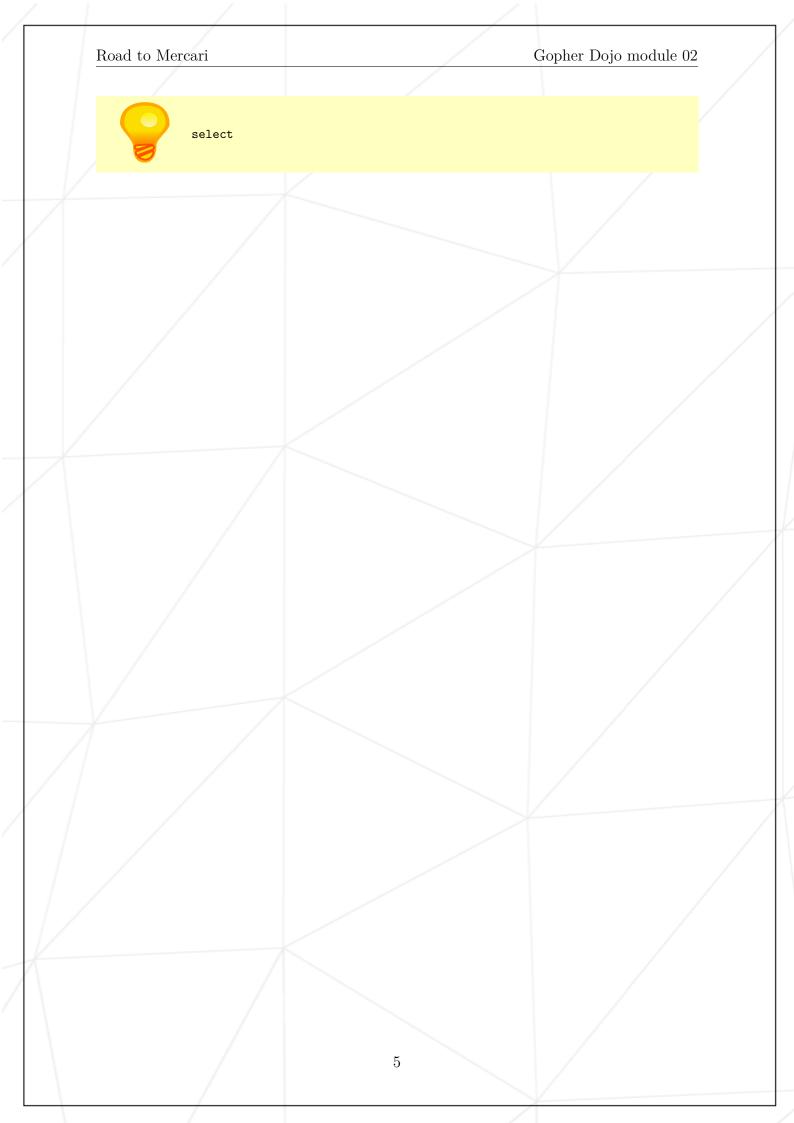
- Print a english word to the standard output.
- Print a prompt to the standard output. You are free to choose what to print as a prompt.
- Accept one line from the user through standard input.
- Verify if the users input and the english word matches.
- Repeat items 1-2 until the game reaches a time limit of 30 seconds.
- At the end of the time limit, display how many times user was able to type the word correctly. You are free to choose how to print the result.

### Example:

```
?> ./typing_game
Water
-> Water
Bulbasaur
-> Bulbasaur
Pokemon
-> Pokemon
Flamethrower
-> Thunder
Time's up! Score: 3
```

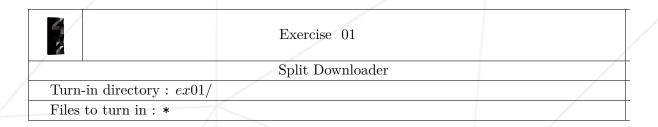


time.After, context.WithTimeout



### Chapter IV

## Exercise 01: Split Downloader



Create a program that download using a parallel requests, meeting the following specifications.

• Accept one url as command-line arguments.

?> ./download https://speed.hetzner.de/100MB.bin

- Use the Range access.
- Download using multiple concurrent goroutines and then merge those downloaded data into a single data in loca
- You must devise a method to handle errors. (use the golang.org/x/sync/errgroup package)
- You must handle if a cancellation occurs.



Accept-Ranges: bytes



Do not access unnecessarily large amounts of time.