



# 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

<b>I</b>	<b>Instructions</b>	<b>2</b>
<b>II</b>	<b>Foreword</b>	<b>3</b>
<b>III</b>	<b>Exercise 00 : Typing game</b>	<b>4</b>
<b>IV</b>	<b>Exercise 01 : Split Downloader</b>	<b>6</b>

# 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 cannot leave any 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/>

# Chapter III

## Exercise 00 : Typing game

	Exercise 00
Standard input/output and time limits	
Turn-in directory : <i>ex00/</i>	
Files to turn in : *	

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`



`select`

# Chapter IV

## Exercise 01 : Split Downloader

	Exercise 01
Split Downloader	
Turn-in directory : <i>ex01/</i>	
Files to turn in : *	

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

- Accept one url as command-line arguments.

```
?> ./download https://releases.ubuntu.com/focal/ubuntu-20.04.4-live-server-amd64.iso
```

- 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](https://golang.org/x/sync/errgroup) package)
- You must handle if a cancellation occurs.



Accept-Ranges: bytes



Do not access unnecessarily large amounts of time.