Go exercises

WASA: Web and Software Architecture

Enrico Bassetti

Hello world

Copy the “Hello World” source (from “Go Basics” slides) in a file *`hello-world.go`*.

Then, run *`go run hello-world.go`*.

Build an executable

```
$ go build -o hello-world hello-world.go
```

The `-o` flag specifies the name/path of the executable.

Exercise 1

Write a Go program that prints if the current “second” is even or odd.

To get the current “second”, use *`time.Now().Second()`*.

Hint:

```
import (
   "time"
   "fmt"
)
```

Exercise 2

Generate random numbers until you find an even number.

Hints:

- Use for *`math/rand`*: https://pkg.go.dev/math/rand
- Add *`rand.Seed(time.Now().UnixNano())`*

Exercise 3

Define a function that, given an integer, returns `true` if the number is even, `false` otherwise.

Add (and use!) the function in exercise 1 or 2.

Exercise 4

Create a simple web server using *net/http* package from the standard library. It should serve a plain text web page on port *8090* with the output of exercise 2.

See https://pkg.go.dev/net/http for package documentation.

Hints:

- Read doc for *http.ListenAndServe()* and *http.HandleFunc()*
- A *http.ResponseWriter* "contains" *io.Writer* - you can pass a *http.ResponseWriter* in place of *io.Writer* in any function — e.g., those in *fmt*
  - next lecture we'll see what "contains" means
- Test it with your browser OR using cURL
  - *curl http://localhost:8090/*

Exercise 5

Create a simple web server using *net/http* package from the standard library. It should serve a plain text web page greeting you (e.g., “Hi John Doe!”). The name should be sent via query string.

The URL should be something like *http://localhost:8090/?name=John+Doe*.

Hints:

- The query string is in *r.URL.Query()*
  - doc for type *url.URL*: https://pkg.go.dev/net/url#URL
- Test it with your browser OR using cURL
  - *curl "http://localhost:8090/?name=John+Doe"*

Exercise 6

Create a simple web server using `net/http` package from the standard library. It should serve a plain text web page greeting you (e.g., “Hi John Doe!”). The name should be sent as POST request body.

Hints:

- The body is in `r.Body`
  - you can read it all using e.g. `io.ReadAll()`
- Test it using cURL (command is one line)
  - `curl -d 'John Doe' -H 'Content-Type: text/plain' http://localhost:8090`