Golang

The following list of resources contains everything one may need to get started with Go and progress into writing clean, simple and extensible Go code following all the community guidelines and also being aware of the most tricky and advanced Go traps.

All the links contain very valuable information but if you are short on time make sure to atleast read the links that are **bolded** and **bolded.**

1. Basic Go

   1. **Do the getting started tutorials A Tour of Go**
   2. **Do the TDD exercises**
   3. The list from Go by example is a great list of small hands on examples regarding the different aspects of the langauge
   4. Go over the examples from the ultimate go guide
   5. Look around, read anything that seems interesting, there are a lot of examples in the basic guidelines collection
   6. **Learn more about Go interfaces**
      a. **How To Use Go Interfaces**
      b. **Interfaces in Go (part I) - Polymorphism**
      c. **Interfaces in Go (part II) - Type assertion & type switch**
      d. **Interfaces in Go (part III) - Deep Dive**
      e. Preemptive Interface Anti-Pattern in Go

2. Basic Project Layout

   1. **Opinionated Project Layout**
   2. **Learn how to structure your applications**
   3. **Best practices around structuring your project**
   4. Read about the aspects of a good library

3. Practical Guidelines

   1. **Read about Peter Bourgon · Go best practices, six years in**
   2. **Read about Practical Go | Dave Cheney is a great collection of resources, atleast the Presentations section is a must read**
   3. Read about Peter Bourgon Industrial Programming with Go

4. Common Mistakes and traps

   1. **Avoid falling into the CommonMistakes**
   2. **Mind the 50 go traps - this is a great collection of "traps"**

5. Go clean code guidelines

   1. **Make sure you address the community code review comments**
   2. **Have in mind Uber's Golang guide**
   3. **Write effective go**
   4. **And effective go part 2**
   5. **Follow the SOLID principles and look at some great examples here**
   6. **Follow these 12 best practices**
   7. **Write clean Go - A reference for the Go community that covers the fundamentals of writing clean code and discusses concrete refactoring examples specific to Go**
   8. Follow the go proverbs
   9. Follow the The Zen of Go and read about The Zen of Go | Dave Cheney
   10. Follow these Golang tricks and advices
   11. Check out project SEN's guidelines
   12. Learn how to style your packages
   13. Write effective Ginkgo

6. Advanced Go

   1. Long but great series on microservices with Go and a few more great opensource technologies
   2. Write high performance Go
   3. Handle errors gracefully and consider using stacktraces
      a. Go 1.13 supports natively wrapping, unwrapping and error checking
   4. Construct objects using functional options
      a. https://dave.cheney.net/2014/10/17/functional-options-for-friendly-apis
      b. https://sagikazarmark.hu/blog/functional-options-on-steroids/