Getting Started

Overview of Go Programming Language

Go is a statically typed, systems programming language.

```
// Dynamic Typing
var x = 10; // integer
console.log(10);

x = "messing up data types"; // string
console.log(x);
```

```
// Static Typing
var x = 10; // integer
fmt.Println(x);

x = "messing up data types"; // string, incompatible assignment
```

Setting up Go Development Environment

- Installing Go
- Do `go version` to verify go installation.
- Go environment variables:
    - `GOROOT`: Path to go binary and standard library, usually something like `/usr/local/go` on Linux
    - `GOPATH`: Developer workspace that contains go application source code, dependencies & go application binaries
    - `GOBIN`: Default path to Go application binaries, usually `$GOPATH/bin`
- Modules: Top level parent collection of packages constituting application code
- Packages: Collection of functions providing one functionality, e.g. strings package allows working with strings
    - Package name required as the first line of a go file
    - `package main` and `func main` are required for any go program to run
    - Package name should be unique
    - There cannot be more than one package in a single directory, except package names suffixed with `_test`

Working with Packages and Modules

Basics of Go Syntax and Conventions

- Package
- Functions and the main() function
- Importing and using packages
- Variables
- Struct
- Members
- Anonymous members
- Interfaces
- Defining interfaces
- Implicitness of Interfaces
- panic and recover
- defer
- Concurrency