Intro: Golang

$ whoami

- Masahiro Kitamura (@nasa9084)
- Hokkaido/Sapporo -> Saitama/Asaka
- Python / Go / emacs / Whisk(e)y
- PyCon JP / builderscon

Golang

- Googleが開発しているプログラミング言語
- 静的型付けで堅く書ける
- 動的型付け言語のように軽く書ける
- 言語使用がシンプル
- クロスプラットフォーム開発が(比較的)容易
- 並列処理を簡単に書ける

シンプルな言語仕様

Do you need “while” loop?

- Goにwhileはないのでforのみ

```
for i:=0; i<10; i++ {
    fmt.Println(i)
}

i := 0
for i < 10 {
    fmt.Println(i)
    i++
}
```

わかりやすいアクセス制限

- 大文字で始まる名前はpublic (exported)
- 小文字で始まる名前はprivate (unexported)

Battery Included

便利な標準パッケージ

- crypto/md5, crypto/sha512などのハッシュアルゴリズム
- encoding/csv, encoding/jsonなどのデータエンコーディング
- flag: コマンドラインフラグパーサ
- image/jpeg, image/pngなどの画像操作
- net/http: httpクライアント・httpサーバ
- strconv, strings: 文字列の操作

便利なツール群

- gofmt, goimport: ソースコード整形ツール
- golint: 静的解析ツール
- go vet: ヒューリスティック問題のレポートツール
- present: プレゼンテーション作成ツール

“堅く”

小煩いコンパイラ

- 宣言したのに使っていない変数はコンパイルエラー
- インポートしたのに使っていないパッケージもコンパイルエラー
- 暗黙の型変換はしない

エラーハンドリング

- **例外はない**
  - 多値を返せるので、エラー変数を返す
    - これをその場その場で処理する
- エラー安全な遅延実行(defer)
  - 関数終了時に指定処理を実行できる

TEST

testingパッケージ

- $ go test
  - UnitTestを簡単に実行
  - TestXXXという名前の関数を順に実行
    - ExampleXXXという名前の関数のエラーを検出
- $ go test -bench
  - ベンチマークを簡単に実行
  - BenchmarkXXXという名前の関数を順に実行

I ♥ Golang


builderscon