

# Pen 言語の進捗報告

@raviqqe

# 目次

- 進捗報告
  - v0.2 のリリース
  - C 言語呼び出し規約による外部関数エクスポート
  - wasm32 ターゲットでの末尾呼び出し最適化
  - その他諸々
- テストフレームワークのデモ

# 進捗報告

## v0.2 のリリース

- 色々ありがとうございました 🙏
- 多かったフィードバック
  - 言語で書かれたアプリの例がみたい
    - 言語機能の使用例ではよく分からない
    - 標準パッケージの拡張が必要
  - System injection の何がいいのか分からない
    - 短期的 vs 長期的な開発の生産性
    - 様々なバックグラウンドを持った人々に説明することの難しさ

## C 言語呼び出し規約による外部関数エクスポート

- Pen 側で定義された関数を他言語にエクスポートするとき、C 言語の呼び出し規約が使えるようになった
- 今まで C 呼び出し規約はインポートのみ対応
- C 呼び出し規約でエクスポートされた関数内では yield できない
  - Pen 言語の設計上、引数として yield する関数を渡さない限り問題ない

```
foreign "c" pen_cook_pasta = \() Pasta {  
  ...  
}
```

## wasm32 ターゲットでの末尾呼び出し最適化

- wasm32 で末尾呼び出し最適化をするようにした
  - `llc` のオプションを変えるだけ
- `wasm-interp` (`wabt`) と V8 (`Node.js`) しかまだ対応していない
  - `wasm-interp` では WASI が experimental
  - V8 では WASI と末尾呼び出し命令の両方が experimental
  - 2つの処理系で違うエラーが出る...

# Node.js での実行例

```
node \  
  --experimental-wasi-unstable-preview1 \  
  --experimental-wasm-return_call \  
  run.mjs
```

```
import { readFile } from "fs/promises";  
import { argv, env } from "process";  
import { WASI } from "wasi";  
  
const wasi = new WASI({  
  args: argv,  
  env,  
  preopens: { "/" : process.env.PWD },  
});  
  
wasi.start(  
  await WebAssembly.instantiate(  
    await WebAssembly.compile(await readFile("./app.wasm")),  
    { wasi_snapshot_preview1: wasi.wasiImport },  
  ),  
);
```

## その他諸々

- 存在しないモジュールのエラーメッセージが分かりやすくなった

```
# import 'Foo'Bar と書いたとき  
> pen build  
error: module 'Foo'Bar not found
```

- 型の存在チェック

```
# 型 Bar が存在しなくても型 Foo が使われてなければエラーにはならなかった  
type Foo = Bar
```



## その他諸々(続き)

- プライベートモジュール
  - 同じパッケージ以外からアクセスできないモジュール
- 標準パッケージの拡張
  - ファイルの移動: `File'Move(ctx, src, dest)`
  - 数学的関数の追加

テストフレームワーク

## テストフレームワーク (WIP)

- `pen test` コマンドでパッケージ内のテストを実行
- `.test.pen` の拡張子を持ったファイルがテストモジュール
- テスト関数の型: `\() none | error`

```
import 'Foo
import Test'Assert

TestFooBar = \() none | error {
  Assert'True(Foo'Bar(42, 13) == 0)
}
```

# テストフレームワークのデモ