

OCaml を始めるには時期が良い

yuki @lmdexpr

2025-01-18

はじめに

**この中に最近 OCaml を使ったことのある方はいますか？
(直近 2 年以内とします)**

はじめに

**この中に最近 OCaml を使ったことのある方はいますか？
(直近 2 年以内とします)**

→ ごめんなさい。今回の話は知っている内容かもしれません。

今回の話の対象者

L1

- 関数型プログラミングに興味があるが触ったことがない

L2

- 関数型プログラミング言語を触ったことがあるが普段は別の言語

L3

- 他の関数型言語を主に使っている
- 昔の OCaml は触ったことがある

OCaml について (L1 ~ L2)

OCaml とは、

- 非純粋
 -
- 正格評価
 -
- 静的型付けと強力な型推論
 -

を特徴とした関数型プログラミング言語です。

OCaml について (L3)

OCaml とは、

- 非純粋
 - 手続き型プログラミングや副作用を直接扱える
- 正格評価
 - モナドが不要
- 静的型付けと強力な型推論
 - 基本的には型を書かなくても良いが、抽象化は出来る

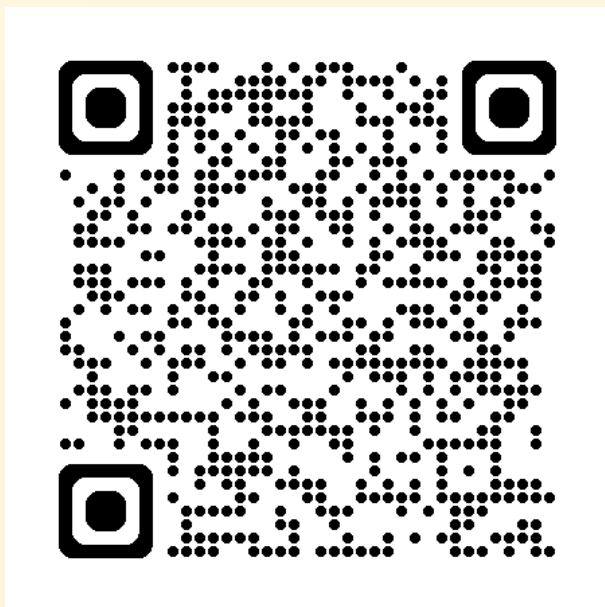
を特徴とした関数型プログラミング言語です。

L3 向け補足

モナドは不要ですが、モナドっぽい書き方も出来ます。

Chapter 12 Language extensions

<https://ocaml.org/manual/5.3/bindingops.html>

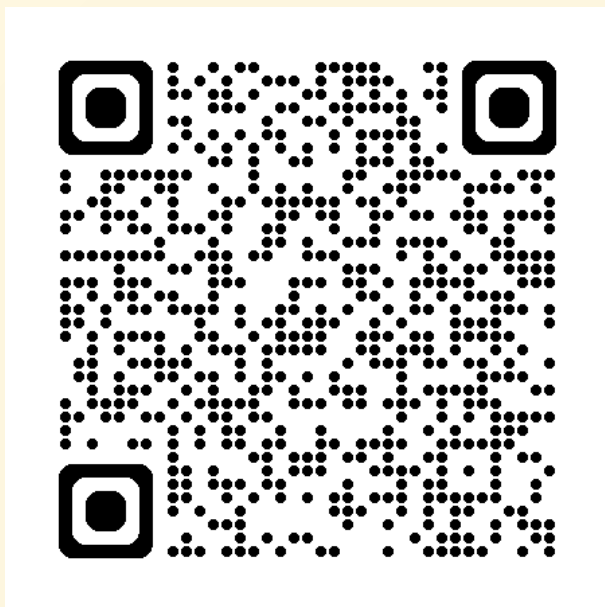


L3 向け補足

OCaml における (IO) モナドに関するスレッド:

IO Monad for OCaml

<https://discuss.ocaml.org/t/io-monad-for-ocaml/4618/11?u=Imdexpr>



今熱い理由

- 並行並列プリミティブが 5.0 で導入
 - それに伴い、ドキュメントも一新
- 5.1, 5.2, 5.3 で数々の改善
 - エラーメッセージの改善
 - 標準ライブラリに 97 個の新しい関数
 - その他、ランタイム、パフォーマンスの改善などなど
- 更に 2025-01-08 リリースの 5.3 では Deep Handler の syntax support も入り、本格的に使えるように！(後述)

今熱い理由

- 並行並列プリミティブが 5.0 で導入
 - それに伴い、ドキュメントも一新
- 5.1, 5.2, 5.3 で数々の改善
 - エラーメッセージの改善
 - 標準ライブラリに 97 個の新しい関数
 - その他、ランタイム、パフォーマンスの改善などなど
- 更に 2025-01-08 リリースの 5.3 では Deep Handler の syntax support も入り、本格的に使えるように！(後述)

→ 激熱🔥

Algebraic Effects and Handlers とは (L1, L2)

- 復帰できる例外のこと
- それだけなのに色々なものが実装できてしまう
 - async/await
 - トランザクション
 - ジェネレータ
 - 状態管理
 - その他色々...

Algebraic Effects and Handlers とは (L1, L2)

復帰とは？

→ 例外の発生地点に戻ってくること

```
try {  
  let name = user.name;  
  if (name === null) {  
    throw new Error('error');  
  }  
  return name; // null の場合、ここには到達しない  
} catch (e) {  
  console.log(e);  
}
```

Algebraic Effects and Handlers とは (L1, L2)

※ 嘘文法です

出典: <https://overreacted.io/algebraic-effects-for-the-rest-of-us/>

```
try {  
  let name = user.name;  
  if (name === null) {  
    name = perform 'ask_name';  
  }  
  return name;  
} handle (effect) {  
  if (effect === 'ask_name') {  
    resume with 'Arya Stark';  
  }  
}
```

Algebraic Effects and Handlers とは (L3)

- 限定継続が取れる例外のこと
- Monad / Extensible Effects でやってたようなことがやりたい

5.3 で出来るようになったこと

```
let rec spawn f =  
  match_with f {  
    retc = (fun () -> dequeue ());  
    exnc = (fun e -> print_string (Printexc.to_string e); dequeue ());  
    effc = (fun (type a) (e : a Effect.t) ->  
      match e with  
      | Yield -> Some (fun k : (a, unit) continuation) ->  
        enqueue k; dequeue ()  
      | Fork f -> Some (fun k : (a, unit) continuation) ->  
        enqueue k; spawn f  
      | _ -> None  
    );  
  };
```

5.3 で出来るようになったこと

```
let rec spawn f =  
  match f with  
  | ()                -> dequeue ()  
  | exception e       -> print_string (Printexc.to_string e); dequeue ()  
  | effect Yield, k    -> enqueue k; dequeue ()  
  | effect (Fork f), k -> enqueue k; spawn f
```

出典:

https://x.com/kc_srk/status/1789140451766779983

未来の話

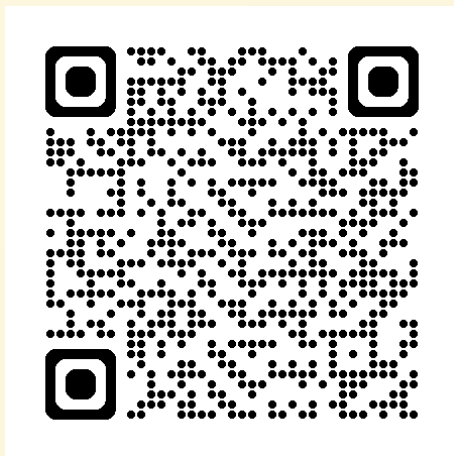
- 現状
 - 並行並列が色々まとまってきた
- これから
 - Flambda2 (OCaml の最適化フレームワーク後継) の開発が進行中
 - WASM へのコンパイル対応も絶賛進行中 ← 熱い
- もう少し先
 - Moduler Implicits
 - Typed Algebraic Effects

OCaml を始めるには時期が良い

これから始めるなら？

まずは公式サイトから playgroundで触ってみましょう

<https://ocaml.org>

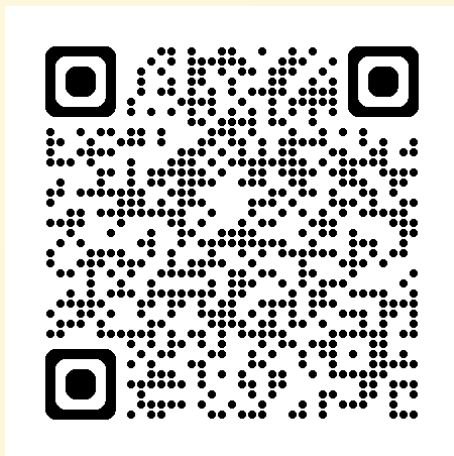


(デフォルトのサンプルがゴリゴリで面白い)

これから始めるなら？

install するなら opam から

<https://ocaml.org/docs/installing-ocaml>



opam は OCaml のパッケージマネージャ
OCaml 自体のインストール、バージョン管理も出来るのでここから

これから始めるなら？

その他のツールチェーン <https://ocaml.org/platform>

- dune : ビルドツール
- utop : REPL
- merlin : Vim, Emacs のためのツール
- ocaml-lsp-server : LSP (個人的には merlin よりこっちがオススメ)
- ocamlformat : コードフォーマッタ

将来的には dune に統一していきたいらしい

参考: <https://ocaml.org/tools/platform-roadmap>

これから始めるなら？

Books

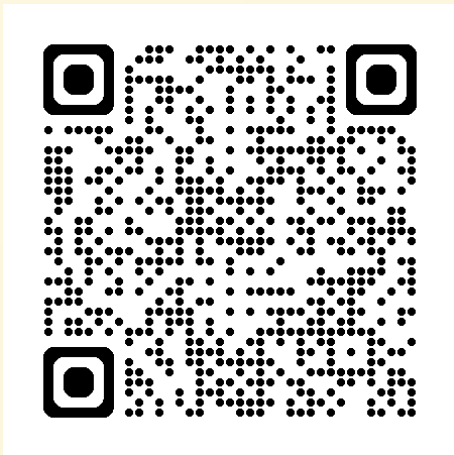
<https://ocaml.org/books?language=All&difficulty=All&pricing=free>

- プログラミングの基礎
 - 日本語の入門書
- OCaml Programming: Correct + Efficient + Beautiful
 - コーネル大学の教材 (CS3110)
- Real World OCaml
 - デファクトスタンダードなライブラリ (Base, Core など) を含め、実践的な内容が記載されている

これから始めるなら？

その他の情報源

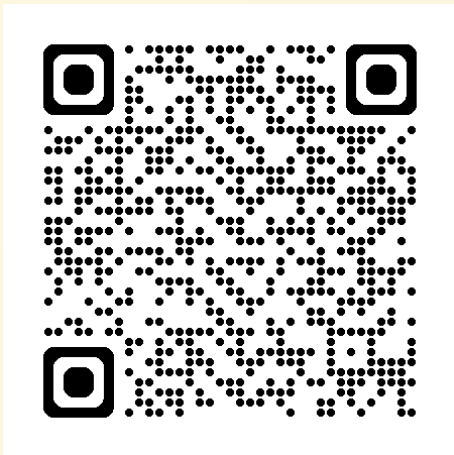
<https://ocaml.org/community>



これから始めるなら？

Exercises

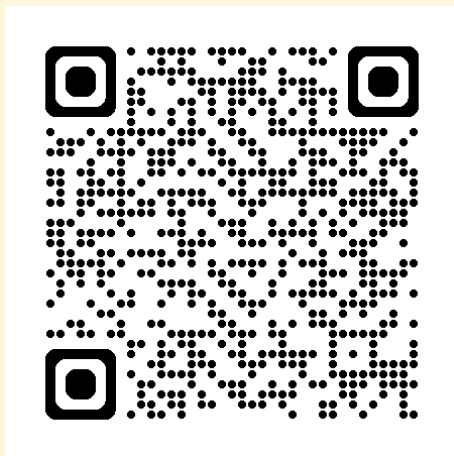
<https://ocaml.org/exercises>



これから始めるなら？

Exercises

<https://ocaml.org/exercises>



AtCoder とかでも使えるので、競プロ好きな方は是非
(現時点でアクティブに使っているのはどうやら私だけ.....)

これから始めるなら？

5.x 目玉機能 Algebraic Effects and Handlers については、

- <https://github.com/ocaml-multicore/effects-examples>
 - 既に 5.3 に対応している
- <https://github.com/ocaml-multicore/ocaml-effects-tutorial>
 - issue #13 にて 5.3 対応しようとしている

さいごに

さいごに

OCaml 始めてくれ頼む！

Questions ?