

OCaml最高!
スマホ開発にヨ

Prepared for NGK2023S LT



Follow

Haochen Kotoi-Xie

@haochenxie

ぶろぐらみんぐ ちょっと できる (ちゅうい: つぶやき きえること ある)

◎ Metaland ☺ twitter.com/kxcinc Joined March 2013

593 Following 356 Followers

Haochen Kotoi-Xie (twitter: @haochenxie) on 2023-01-21

OCaml最高!

スマホ開発にも使えちやう？!

Prepared for NGK2023S LT

Haochen Kotoi-Xie (twitter: @haochenxie) on 2023-01-21

OCamlはなに？

みんな大好き ChatGPT 先生に聞いてみた！



OCaml はなに？ おいしいなにか？



OCamlはプログラミング言語です。それは、コンピュータに命令を与えるために使用されます。食べ物とは関係ありません。



OCamlはなに?

みんな大好き ChatGPT 先生に聞いてみた!



OCaml はなに? おいしいなにか?



OCaml
用され



OCaml はなに? すごい言語なの?



OCaml (Objective Caml) は、静的型付けのプログラミング言語で、関数型プログラミングのパラダイムをサポートしています。それは、高速で型安全で、柔軟な開発を支援することができることで知られています。OCamlは、大規模なプロジェクトにも適しており、特に、高性能のシステムプログラミングや、理論計算機科学の分野で広く使用されています。



ChatGPT 先生が教えてもらった..

OCamlは高速で型安全！



OCaml (Objective Caml) は、静的型付け機能を備えた実行可能言語で、オブジェクト指向プログラミングのパラダイムをサポートしています。また、並列計算の開発を支援することができることで知られており、特に、高性能のシステム開発にも適しており、特に、高性能のシステム開発を支援することができます。OCaml は、多くの分野で広く使用されています。



Created with DALL·E, an AI system by OpenAI

スマホアプリ開発にも使いたいね

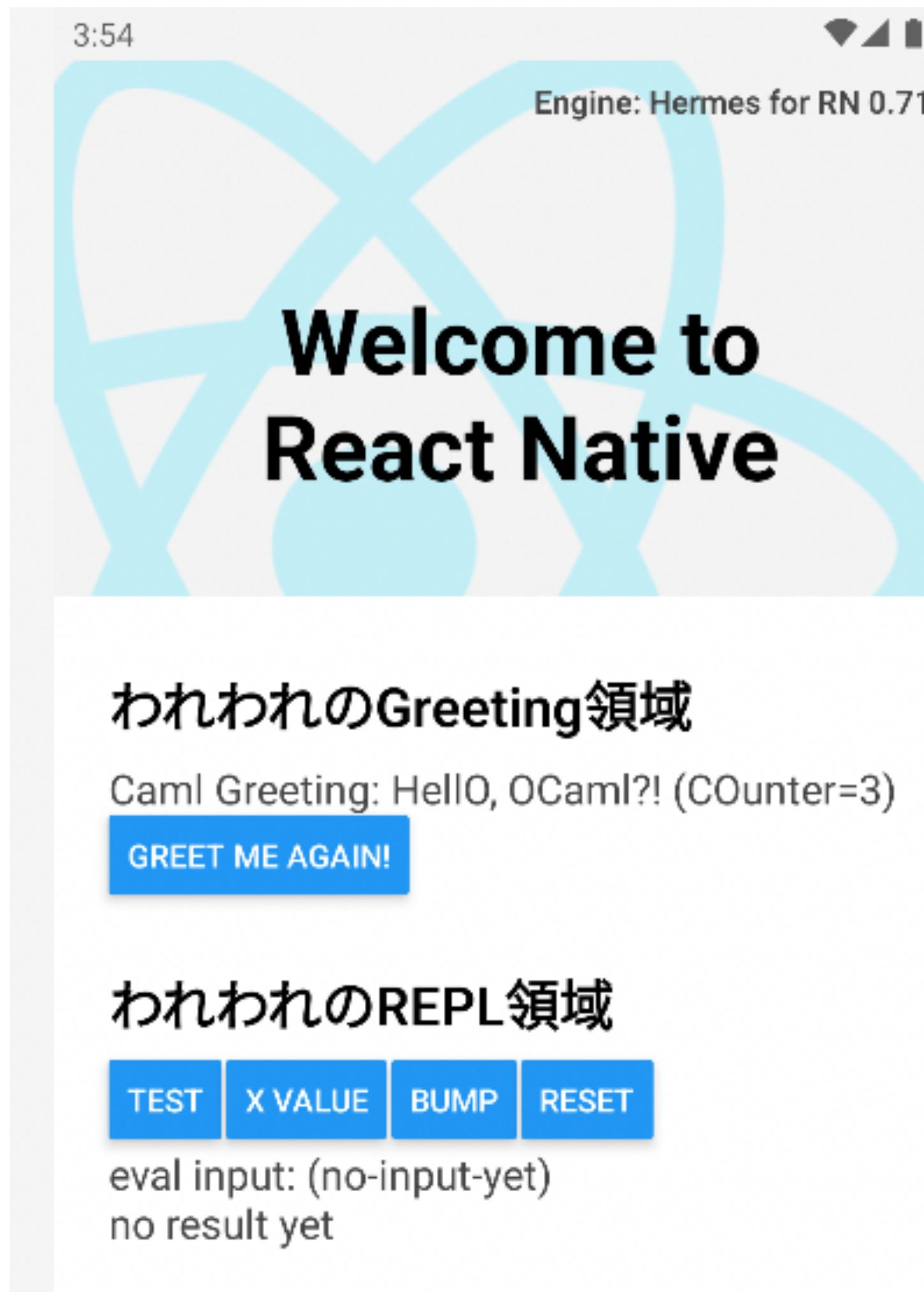


ちょっと調べてみれば...

- js_of_ocaml : OCaml バイトコード → JavaScript
- ReScript/ReasonML : OCaml ソースコード → JavaScript
- React Native : JavaScript/TypeScript でアプリ開発

==> できちゃうのでは 😊

実際やってみよう



```
<Section title="われわれのGreeting領域">
  <Text>`Caml Greeting: ${Caml.caml_greeting()}`</Text>
  <Button title={'Greet me again!'} onPress={stepRefreshCounter} />
</Section>
<Section title="われわれのREPL領域">
  <View style={{flex: 1, flexDirection: 'row', columnGap: 2}}>
    <Button title={'Test'} onPress={() => replEval( input: 'test')}>/>
    <Button title={'X Value'} onPress={() => replEval( input: 'var_x')}>/>
    <Button title={'bump'} onPress={() => replEval( input: 'bump')}>/>
    <Button title={'reset'} onPress={() => replEval( input: 'reset')}>/>
  </View>
  <Text>`\n`</Text>
  <Text>`eval input: ${replInput}`</Text>
  <Text>`\n`</Text>
  {replResult === null ? (
    <Text>no result yet</Text>
  ) : (
    <Text>`eval result: ${JSON.stringify(
      replResult,
      replacer: null,
      space: 2,
    )}`</Text>
  )}
</Section>
```

実際やってみよう

われわれのGreeting領域

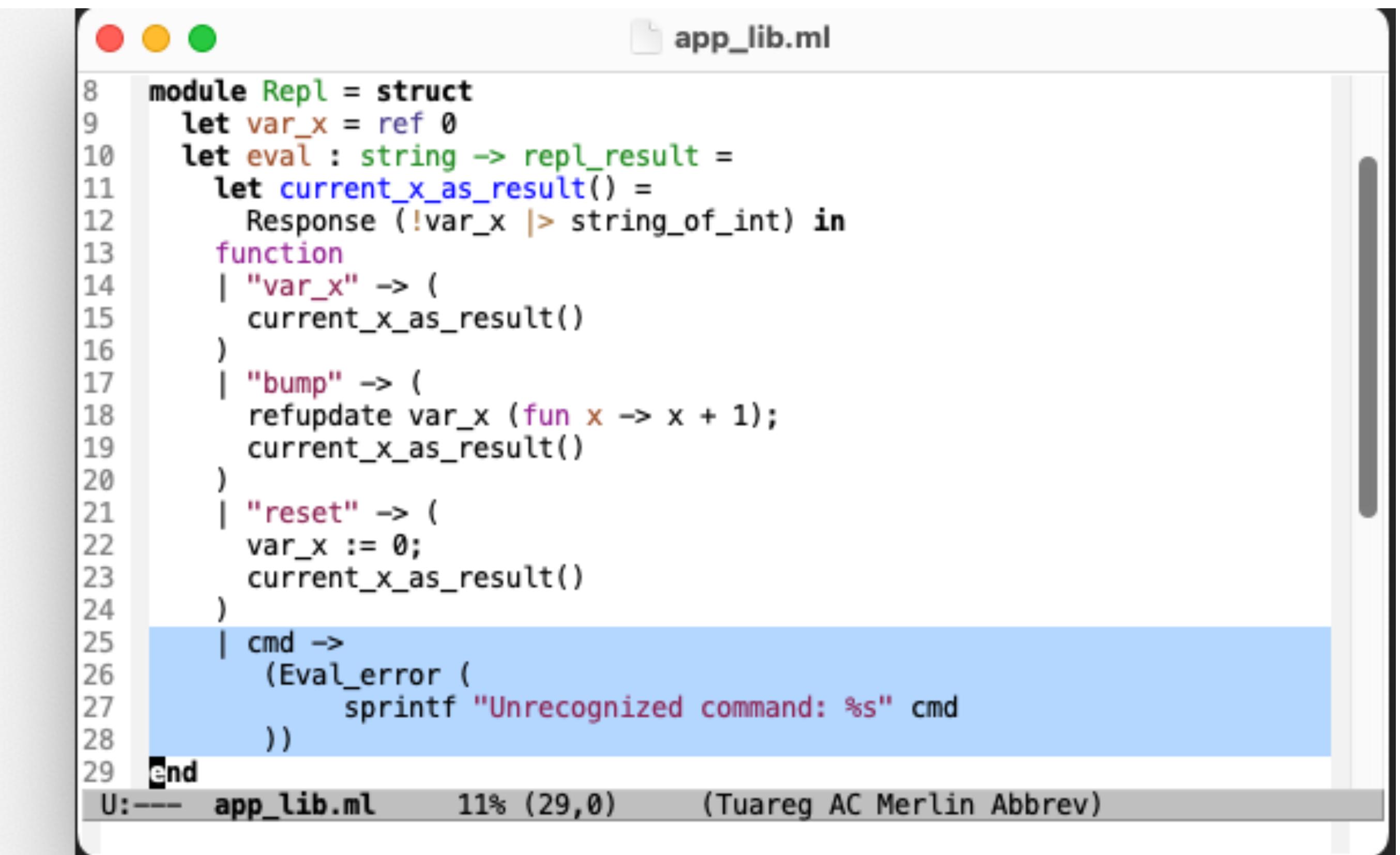
Caml Greeting: Hello, OCaml?! (CCounter=5)

GREET ME AGAIN!

われわれのREPL領域

TEST X VALUE BUMP RESET

```
eval input: test
eval result: [
  "Eval_error",
  "Unrecognized command: test"
]
```



The screenshot shows a code editor window with the file `app_lib.ml` open. The code defines a module `Repl` with a structure that includes a reference to a counter and a function `eval` that handles various commands. The command `cmd` is highlighted with a blue background, indicating it is the current input being processed.

```
8  module Repl = struct
9    let var_x = ref 0
10   let eval : string -> repl_result =
11     let current_x_as_result() =
12       Response (!var_x |> string_of_int) in
13     function
14     | "var_x" -> (
15       current_x_as_result()
16     )
17     | "bump" -> (
18       refupdate var_x (fun x -> x + 1);
19       current_x_as_result()
20     )
21     | "reset" -> (
22       var_x := 0;
23       current_x_as_result()
24     )
25     | cmd ->
26       (Eval_error (
27         sprintf "Unrecognized command: %s" cmd
28       ))
29   end
```

U:--- app_lib.ml 11% (29,0) (Tuareg AC Merlin Abbrev)

実際やってみよう

われわれのGreeting領域

Caml Greeting: Hello, OCaml?! (CCounter=7)

GREET ME AGAIN!

われわれのREPL領域

TEST X VALUE BUMP RESET

eval input: var_x

eval result: ["Response", "0"]

```
8  module Repl = struct
9    let var_x = ref 0
10   let eval : string -> repl_result =
11     let current_x_as_result() =
12       Response (!var_x |> string_of_int) in
13     function
14     | "var_x" -> (
15       current_x_as_result()
16     )
17     | "bump" -> (
18       refupdate var_x (fun x -> x + 1);
19       current_x_as_result()
20     )
21     | "reset" -> (
22       var_x := 0;
23       current_x_as_result()
24     )
25     | cmd ->
26       (Eval_error (
27         sprintf "Unrecognized command: %s" cmd
28       ))
29   end
```

U:--- app_lib.ml 11% (14,0) (Tuareg AC Merlin Abbrev)

実際やってみよう

われわれのGreeting領域

Caml Greeting: Hello, OCaml?! (CCounter=11)

GREET ME AGAIN!

われわれのREPL領域

TEST X VALUE BUMP RESET

eval input: bump

eval result: [

"Response",
 "2"

]

```
8  module Repl = struct
9  let var_x = ref 0
10 let eval : string -> repl_result =
11   let current_x_as_result() =
12     Response (!var_x |> string_of_int) in
13   function
14   | "var_x" -> (
15     current_x_as_result())
16   )
17   | "bump" -> (
18     refupdate var_x (fun x -> x + 1);
19     current_x_as_result())
20   )
21   | "reset" -> (
22     var_x := 0;
23     current_x_as_result())
24   )
25   | cmd ->
26     (Eval_error (
27       sprintf "Unrecognized command: %s" cmd
28     )))
29 end
```

U:--- app_lib.ml 11% (21,0) (Tuareg AC Merlin Abbrev)

実際やってみよう

われわれのGreeting領域

Caml Greeting: Hello, OCaml?! (CCounter=13)

GREET ME AGAIN!

われわれのREPL領域

TEST X VALUE BUMP RESET

```
eval input: reset
eval result: [
  "Response",
  "0"
]
```



The screenshot shows a Mac OS X window titled "app_lib.ml". The code inside the window is as follows:

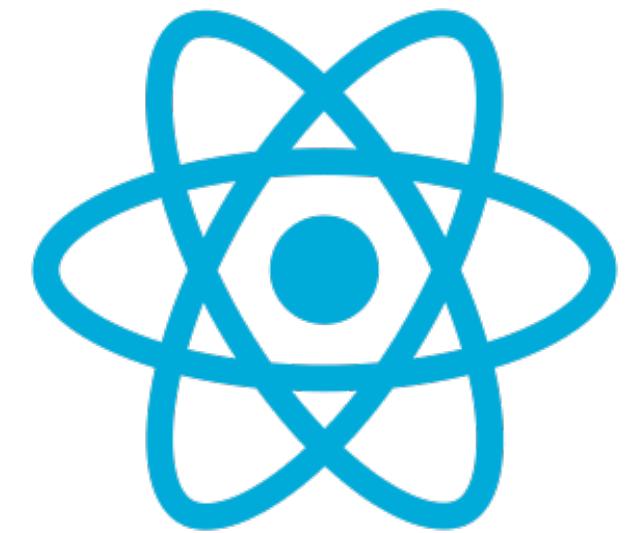
```
8  module Repl = struct
9    let var_x = ref 0
10   let eval : string -> repl_result =
11     let current_x_as_result() =
12       Response (!var_x |> string_of_int) in
13       function
14       | "var_x" -> (
15         current_x_as_result()
16       )
17       | "bump" -> (
18         refupdate var_x (fun x -> x + 1);
19         current_x_as_result()
20       )
21       | "reset" -> (
22         var_x := 0;
23         current_x_as_result()
24       )
25       | cmd ->
26         (Eval_error (
27           sprintf "Unrecognized command: %s" cmd
28         ))
29   end
```

The line "let current_x_as_result() = Response (!var_x |> string_of_int) in" is highlighted with a blue background. The status bar at the bottom of the window shows "U:--- app_lib.ml 11% (25,0) (Tuareg AC Merlin Abbrev)".

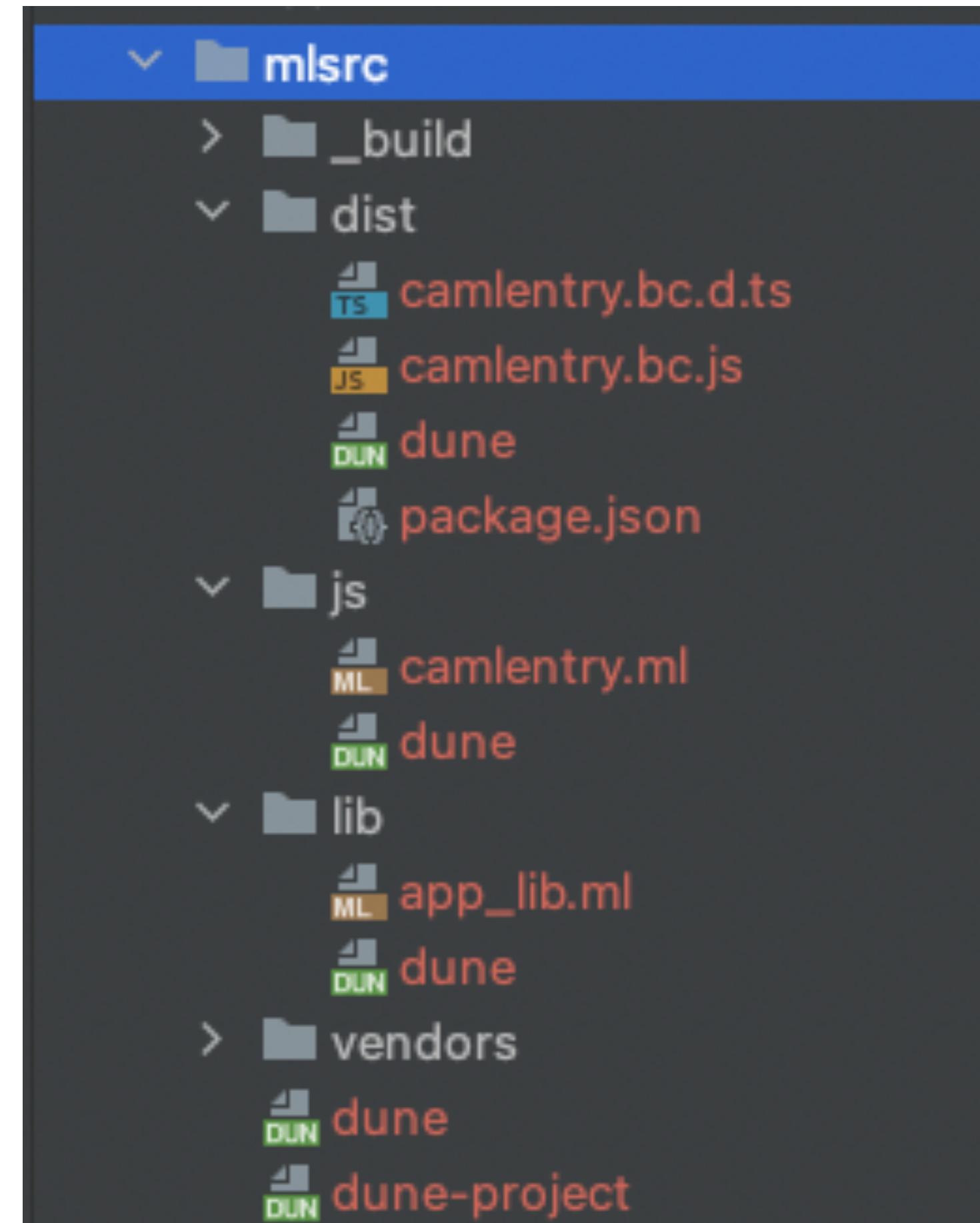
実際やってみよう



×



React Native



- js_of_ocaml : OCaml バイトコード → JavaScript
- ReScript/ReasonML : OCaml ソースコード → JavaScript
- React Native : JavaScript/TypeScript でアプリ開発

==> できちゃうのでは 😁

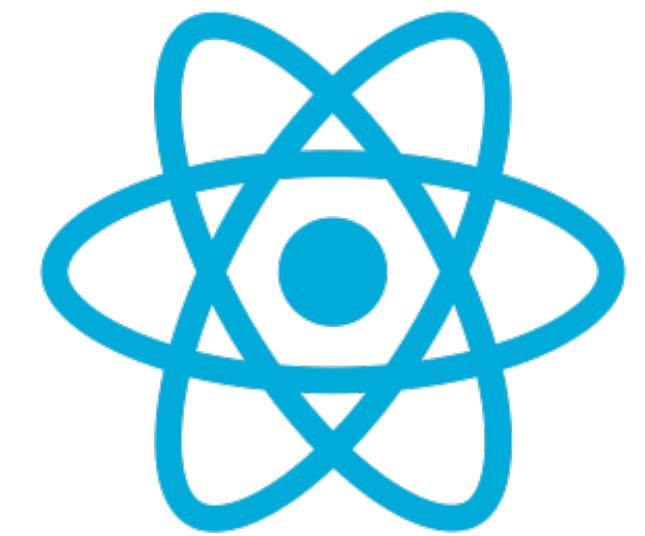
==> できてしまった 😕 😨 😱 😊 😊 😊

世の中便利なものがいっぱい!



OCaml

×



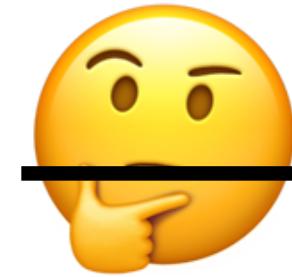
React Native

...

どうせJoy Projectだけじゃない?



どうせJoy Projectだけじゃない?



いやいや

業務でも使っているんだ 😊

アーキテクチャー例

(製品名とかは出せないけど.....)

- SDKプロジェクト (Android/iOS)
- コアアルゴリズム: Proper OCaml (compiled via js_of_ocaml)
- スマホセンサーデータ収集部: React Native Native Modules
- GUI実装: TypeScript (React Native)
 - 実験的に ReScript 導入した → 大人の事情で TypeScript-only になった
- Glue Code: TypeScript (React Native)
- クライエントコード向けAPI: Java / Swift

感想

- マルチプラットフォームでコードシェアリングできて幸せ
- Code ChangeがリアルタイムでプレビューできるReact Nativeは神!
- Native Modules の開発めんどくさい..
- 型安全は福利厚生!

結論

- OCaml はいいぞ! スマホ開発にも使えるんだぞ!

感想

- マルチプラットフォームでコードシェアリングできて幸せ
- Code ChangeがリアルタイムでプレビューできるReact Nativeは神!
- Native Modules の開発めんどくさ
- 型安全は福利厚生!

結論

- OCaml はいいぞ! スマホ開発にも使



FIN.