

Haskell 入門関数型プログラミング言語の基礎と実践（初版第 1 刷）

正誤情報

平成 29 年 10 月 26 日

書籍の内容に誤りのあったことをおわびいたします。

P.25 脚注 10

-
- 誤 中置演算子とあえて書いているように Haskell では中置しない演算子もあります。ただし、Haskell における演算子はほぼすべて 2 つの項 (2 引数) をとる中置演算子です。
2.4.5 で解説します。
- 正 中置演算子とあえて書いているように Haskell では中置しない演算子もあります。ただし、Haskell における演算子はほぼすべて 2 つの項 (2 引数) をとる中置演算子です。
2.4.5 で解説します。
-

P.30 コラム「文字のコードポイント」

-
- 誤 Enum は列挙型 (enumeration) の略で、順に番号を振れるデータの型に対して使う関数です。
- 正 Enum は列挙型 (enumeration) の略で、順に番号を振れるデータの型に対して使う型クラスです。
-

P.43 ページ下部のリスト

-
- 誤 $-(x + 1)^2$ 引数と区別するために $f . g$ にかっこを付ける
- 正 $-(3 + 1)^2$ 引数と区別するために $f . g$ にかっこを付ける
-

P.44 コラム内の表記

-
- 誤 $\lambda x y \rightarrow (x + 2) * (3 + y)$
- 正 $|x y \rightarrow (2 + x) * (3 + y)$
-

P.49 ページ下部の本文

-
- 誤 中置演算子は (と) で囲むことで通常の関数と同じように利用できます。第一引数が中置演算子の左側、第二引数が中置演算子の右側だったものになると覚えてください。

正 中置演算子は (と) で囲むことで通常の関数と同じように利用できます。第一引数が中置演算子の左側、第二引数が中置演算子の右側だったものになると覚えてください。

P.75 本文

誤 () で明示的に参照する識別子を列挙するか、qualifiedas を使ってどのモジュール由來の識別子なのか区別できるように修飾しておくのがいいでしょう。

正 () で明示的に参照する識別子を列挙するか、*qualified ... as* を使ってどのモジュール由來の識別子なのか区別できるように修飾しておくのがいいでしょう。

P.92 脚注 29

誤 以後も同様に deriving 句を指定します。

正 以後も同様に *deriving* 句を指定します。

P.104 本文

誤 例えば、Either a b 型 (2.7.4 参照) とともに アプリケーション内で

正 例えば、Either a b 型 (2.7.4 参照) とともにアプリケーション内で

P.136 最上段のリスト

最上段の何もしない *Prelude*>を入力する必要はありません。

P.142 コラムのリスト

あらかじめ *import System.IO* しておく必要があります。

P.151 本文

誤 同じく System.Drectory モジュールの

正 同じく System.*Directory* モジュールの

P.153 コラム

誤 System.Drectory モジュールの

P.154 コラム 「Windows の実行ファイル対策」

いずれの実行例も、+ではなく++にする。

誤

```
Prelude System.Directory> findExecutable $ "ghc" + exeExtension
Prelude System.Directory> findExecutables $ "ghc" + exeExtension
```

正

```
Prelude System.Directory> findExecutable $ "ghc" ++ exeExtension
Prelude System.Directory> findExecutables $ "ghc" ++ exeExtension
```

P.161 本文

誤 throwMyException 関数が引数の値によって独自の例外を throwIO で発生させます。

正 throwMyException 関数が引数の値によって独自の例外を *throwIO* で発生させます。

P.169 リスト※誤っていたのでまるまる差し替えてください。

誤 game :: [Card] -> [(Score, Hand, Player)]

正 game :: *Stock* -> [(Score, Hand, Player)]

P.178 本文

誤 Human データ構造体は、String -> Int -> Gender...

正 Human データ構造体は、*String* -> Int -> Gender...

P.190 本文

誤 第二引数で指定した初期値のリストの長さが配列のサイズよりも

正 第二引数で指定した初期値のリストの長さが配列のサイズよりも

P.216 下部のリスト

filePath の大文字小文字表記が誤っていました。

誤

```
concatMultiFiles filePaths dst =  
  handleMultiFiles filePaths (\hdl -> copyFile hdl dst)
```

正

```
concatMultiFiles filePaths dst =  
  handleMultiFiles filePaths (\hdl -> copyFile hdl dst)
```

P.219 リスト

本来必要な *do* が抜けていました。

誤

```
handleMultiFiles filePaths $ \hdl ->  
  hClose hdl
```

正

```
handleMultiFiles filePaths $ \hdl -> do  
  hClose hdl
```

P.223 リスト

インデントが紙面上では誤っていました。

誤

```
class (Eq a) => Ord a where  
  compare           :: a -> a -> Ordering  
  ((<), ((<=)), ((>)), ((>=)) :: a -> a -> Bool  
  max, min          :: a -> a -> a
```

正

```
class (Eq a) => Ord a where  
  compare           :: a -> a -> Ordering  
  ((<), ((<=)), ((>)), ((>=)) :: a -> a -> Bool  
  max, min          :: a -> a -> a
```

P.224 下部のリスト

コードの一番上に `{-# LANGUAGE FlexibleInstances #-}` を追加します。

P.240 下部のリスト

simon :: Text など実行できない例が含まれていました。

```
ghci> :set -XOverloadedStrings
ghci> import qualified Data.Text as T
ghci> simon :: Text
ghci> simon = "Many Simons."
ghci> :t T.pack
T.pack :: String -> Text
ghci> :t T.unpack
T.unpack :: Text -> String
```

正

```
ghci> :set -XOverloadedStrings
ghci> import qualified Data.Text as T
ghci> simon = "Many Simons." :: T.Text
ghci> :t T.pack
T.pack :: String -> T.Text
ghci> :t T.unpack
T.unpack :: T.Text -> String
```

P.250 ページ最上部のリスト

コード中から明らかではない、*Animal* の定義については *data Animal = Dog / Pig deriving Show* としてください。

P.254 下部のリスト

1行目の *import* を言語拡張の指定に変更します。

誤

```
import qualified Data.Text as T
```

正

```
{-# LANGUAGE OverloadedStrings #-}
```

P.258 下部のリスト

deriveJSON defaultOptions "Human" の記述がないとエンコードに失敗します。

P.259 ページ下部のリスト

encodeだけでは出力できないので、適宜 B.putStrLn と組み合わせてください。

誤

```
data IntStr = IntData Int | StrData String
encode $ IntData 999
encode $ StrData "World!"
```

正

```
data IntStr = IntData Int | StrData String
B.putStrLn $ encode $ IntData 999
B.putStrLn $ encode $ StrData "World!"
```

P.260 2つめのリスト※誤っていたのでまるまる差し替えてください。

コードの一番上に `{-# LANGUAGE OverloadedLists,OverloadedStrings #-}` を追加します。

誤

```
nameListValue :: Value
```

正

```
{-# LANGUAGE OverloadedLists #-}
nameListValue :: Value
```

P.263 コラム「インスタンスの自前実装」の中段のリスト

言語拡張とライブラリの指定が抜けていました。

誤

```
data Person = Person
```

正

```
{-# LANGUAGE OverloadedStrings #-}
import Data.Aeson
import Data.Aeson.Types

data Person = Person
```

P.263 コラム「Generics の利用」のリスト

import Data.Aeson が抜けていました。

誤

```
{-# LANGUAGE DeriveGeneric #-}  
import GHC.Generics
```

正

```
{-# LANGUAGE DeriveGeneric #-}  
import GHC.Generics  
import Data.Aeson
```

P.269 UserName の定義

-
- 誤 *userName :: Functor f => Lens User User String String*
正 *userName :: Lens User User String String*
-

P.311 本文

-
- 誤 Stack による hjq.cabal の初期の設定では、テストは src ディレクトリ次のモジュールを読み込むようになっているため、すぐにテストが書ける
正 Stack による hjq.cabal の初期の設定では、テストは src ディレクトリ以下のモジュールを読み込むようになっているため、すぐにテストが書ける
-

P.313 本文

-
- 誤 さっそく、HUnit ライブラリを提供している HUnit パッケージを build-depends に追加しましょう。 hs-source-dirs を次のようにします。
正 さっそく、HUnit ライブラリを提供している HUnit パッケージを build-depends に追加しましょう。 *test-suite* を次のようにします。
-

P.318 本文（1段落目）

-
- 誤 このテストがグリーンになるように実装 (*JqFilterParser*) を修正しなくてはいけません。
正 このテストがグリーンになるように実装 (*jqFilterParser*) を修正しなくてはいけません。
-

P.318 本文（リストの下に続く文章）

-
- 誤 これで、jqFilterparserTest も成功するようになりました。
正 これで、*jqFilterParserSpacesTest* も成功するようになりました。
-

P.326 脚注 11

-
- 誤 インストールせず試したいときは stack build 後に stack exec hjq と実行して下さい。
正 インストールせず試したいときは stack build 後に *stack exec hjq-exe* と実行して下さい。
-

P.340 本文

-
- 誤 RDBMS に対応したデータ型生成用のドライバ、スキーマ名、テーブル名、driving 句に追加する型クラス名を渡す必要があります。
正 RDBMS に対応したデータ型生成用のドライバ、スキーマ名、テーブル名、*deriving* 句に追加する型クラス名を渡す必要があります。
-

P.341 脚注 25

-
- 誤 スキーム名とは SQLite ではデータベース名のことであり、
正 スキーマ名とは SQLite ではデータベース名のことであり、
-

P.349 本文

-
- 誤 Model.WeightRecorder についても同様にデータ登録と読み込みを実装します。
正 *Model.WeightRecord* についても同様にデータ登録と読み込みを実装します。
-

P.359 本文

-
- 誤 続いて data-files 次のファイルにアクセスするため、特殊なモジュールを import します。
正 続いて data-files 以下のファイルにアクセスするため、特殊なモジュールを import します。
-

P.369 脚注 61

-
- 誤 外部へのインターフェースなのに対し、src/Web.WeightRecorder.hs はアプリケーション本体としてここでは記述しています。
- 正 外部へのインターフェースなのに対し、src/**Web.WeightRecorder.hs** はアプリケーション本体としてここでは記述しています。
-

P.377 本文

-
- 誤 src では Client.hs と Server.hs に加えて...
- 正 src では Client.hs と Server.hs に加えて...
-

P.386 本文

-
- 誤 次の内容を最後にまとめて書いて、stack runghc counter.hs で実行しましょう。
- 正 次の内容を最後にまとめて書いて、stack runghc counter.hs で実行しましょう。
-

P.387 コラム

-
- 誤 ライブドリ依存の部分を適切に切り出すと方法などがあります。
- 正 ライブドリ依存の部分を適切に切り出す方法などがあります。
-

P.389 本文

-
- 誤 返り値はオークションが開催していない場合を考慮して Maybe AuctionItem とします。
- 正 返り値はオークションが開催していない場合を考慮して **Maybe AuctionItem** とします。
-