Basics of Haskell

Ver.0.1

iPon研資料

基本的な関数

- 四則演算 + - * /
- rem 剰余
  例: `rem 5 2` → 1
  例: `5 `rem` 2` → 1
- div 整数同士の割り算
  例: `div 5 2` → 2
  例: `5 `div` 2` → 2
  返値も整数

組み込み型

- Int 整数型
- Integer 多倍長整数型
- Float 単精度実数型
- Double 倍精度実数型
- Bool 真理値型
  True、False
- Char 文字型

Typeclass

タイプクラス(typeclass)は、タイプの集合。

- Eq
  ==や/=が使える型
- Ord
  順序関係が定義されている型
- Bounded
  最大値と最小値が定義されている型
- Integral
  IntとInteger

- Floating
  FloatとDouble
- Num
  数値型、IntegralとFloating?

リスト

リストの要素は、全て同じ型でなければならない。
例: `["Alice", "Bob", "Lisa"]` → [“Alice", "Bob", "Lisa”]
例: `["Alice", 1, 2]` → エラー

- [] リテラル
  例: `[1, 2, 3, 4]` → [1, 2, 3, 4]
- : 要素の追加
  例: `7:[4, 5, 6]` → [7, 4, 5, 6]
  例: `[1, 2]:[3, 4, 5]` → エラー
  例: `[1, 2]:[[3], [4], [5]]` → [[1, 2], [3], [4], [5]]
- !! インデックスを指定して参照
  例: `[1, 2, 3, 4] !! 2` → 3
- ++ 連結
  例: `[1, 2, 3] ++ [4, 5, 6]` → [1, 2, 3, 4, 5, 6]
  例: `[]++[]++[]` → []
- drop 先頭から指定した個数分の要素を削除
  例: `drop 2 [1, 2, 3, 4]` → [3, 4]
- length 要素数を返す
- head 先頭要素を返す
- tail　先頭要素を除いたリストを返す
- last 最後尾要素を返す
- init 最後尾要素を除いたリストを返す
- reverse 逆順にならべかえる

filter、mapについては、別に説明。

関数定義

- 名前を指定して定義

```
関数名 引数1 引数2 ... 引数n = 関数本体
```

- ガードを使った定義

```
関数名 引数1  引数2 ... 引数n
    | 条件1          = 関数本体
```

```
    | 条件2          = 関数本体
    ....
    | 条件n          = 関数本体
    | otherwise   = 関数本体
```

- λ抽象
  例: `(\x -> x * x) 2` → 4
  例: `let fun = \x -> x * x`