

# 1. Malgo リファレンスマニュアル

Malgo は、多相型、型推論、カーリー化、パターンマッチなどの機能を持つ静的型付き関数プログラミング言語である。

## 1.1. 字句

### 1.1.1. コメント

`--` から行末まではコメントになる。`{- -}` で囲んだ部分もコメントになる。これは入れ子になってもよい。

### 1.1.2. 識別子

識別子は、正規表現 `[a-zA-Z_][a-zA-Z0-9_#']*` で表現される文字列、あるいは `+--*/\%=><:;|&!#.` の一文字以上の繰り返しで表現される文字列である。

この文書では、`Int32`、`Type` のように大文字から始まる識別子を upper id、`putStrLn`、`main` のように小文字から始まる識別子を lower id、`->`、`+` のように記号から構成される識別子を operator id と呼ぶ。

### 1.1.3. 予約語

以下の文字列は予約語であり、識別子として使うことはできない。

`data, foreign, import, infix, infixl, infixr, let, module`

### 1.1.4. リテラル

リテラルには、末尾に `#` がつく `unboxed` リテラルと、通常の値を表す `boxed` リテラルがある。それぞれのリテラルの例を列挙する。

- 数値リテラル
  - 32bit 符号付き整数 : `unboxed 42#`, `boxed 42`

- 64bit 符号付き整数 : `unboxed 42L#`, `boxed 42L`
- 単精度浮動小数点数 : `unboxed 3.14F#`, `boxed 3.14F`
- 倍精度浮動小数点数 : `unboxed 3.14#`, `boxed 3.14`
- 文字リテラル : `unboxed 'a' #`, `boxed 'a'`
- 文字列リテラル : `unboxed "hello" #`, `boxed "hello"`

## 1.2. 宣言

### 1.2.1. モジュール

モジュールは一連の型や関数宣言の並びのまとまりであり、名前空間として振る舞う。例えば、モジュール `List` で宣言された関数 `map` と、モジュール `Set` で宣言された関数 `map` は別の名前として扱われる。

```
module = "module" upper_id "=" "{" decs "}" ;

decs = { dec ";" } ;

dec = data_dec | infix_dec | foreign_dec | import_dec
    | signature_dec | function_dec ;
```

### 1.2.2. 関数宣言

```
signature_dec = lower_id "::" type ;

def_dec = lower_id "=" "{" { pat } "->" stmts "}"
    | lower_id "=" "{" stmts "}" ;
```

### 1.2.3. データ型宣言

```
data_dec = "data" upper_id { lower_id } "=" con_defs ;

con_defs = con_def "|" con_defs
    | con_def ;
```

```
con_def = upper_ident { type } ;
```

#### 1.2.4. 結合性宣言

```
infix_dec = "infix" decimal "(" operator_id ")" ;
```

#### 1.2.5. 外部関数宣言

```
foreign_dec = "foreign" "import" lower_ident "::" type ;
```

#### 1.2.6. インポート宣言

```
import_dec = "import" upper_id ;
```