

Chapter 2. Happy を使う (Using Happy) より

チュートリアルを写経など：<https://sites.google.com/site/paclearner/happy-jp/sec-using-html>

```
{
module Main where

import Char
}

%name calc
%tokentype { Token }
%error { parseError }

%token
  let { TokenLet }
  in { TokenIn }
  int { TokenInt $$ }
  var { TokenVar $$ }
  '=' { TokenEq }
  '+' { TokenPlus }
  '-' { TokenMinus }
  '*' { TokenTimes }
  '/' { TokenDiv }
  '(' { TokenOB }
  ')' { TokenCB }

%%

Exp : let var '=' Exp in Exp { Let $2 $4 $6 }
    | Exp1 { Exp1 $1 }

Exp1 : Exp1 '+' Term { Plus $1 $3 }
      | Exp1 '-' Term { Minus $1 $3 }
      | Term { Term $1 }

Term : Term '*' Factor { Times $1 $3 }
      | Term '/' Factor { Div $1 $3 }
      | Factor { Factor $1 }
```

```

Factor : int { Int $1 }
      | var { Var $1 }
      | '(' Exp ')' { Brack $2 }

{
parseError :: [Token] -> a
parseError _ = error "Parse error"

data Exp
  = Let String Exp Exp
  | Exp1 Exp1
  deriving Show

data Exp1
  = Plus Exp1 Term
  | Minus Exp1 Term
  | Term Term
  deriving Show

data Term
  = Times Term Factor
  | Div Term Factor
  | Factor Factor
  deriving Show

data Factor
  = Int Int
  | Var String
  | Brack Exp
  deriving Show

data Token
  = TokenLet
  | TokenIn
  | TokenInt Int
  | TokenVar String
  | TokenEq
  | TokenPlus
  | TokenMinus

```

```

    | TokenTimes
    | TokenDiv
    | TokenOB
    | TokenCB
    deriving Show

lexer :: String -> [Token]
lexer [] = []
lexer (c:cs)
    | isSpace c = lexer cs
    | isAlpha c = lexVar (c:cs)
    | isDigit c = lexNum (c:cs)
lexer ('=:cs) = TokenEq : lexer cs
lexer ('+:cs) = TokenPlus : lexer cs
lexer ('-:cs) = TokenMinus : lexer cs
lexer ('*:cs) = TokenTimes : lexer cs
lexer ('/:cs) = TokenDiv : lexer cs
lexer ('(:cs) = TokenOB : lexer cs
lexer (')':cs) = TokenCB : lexer cs

lexNum cs = TokenInt (read num) : lexer rest
    where (num, rest) = span isDigit cs

lexVar cs =
    case span isAlpha cs of
        ("let", rest) -> TokenLet : lexer rest
        ("in", rest) -> TokenIn : lexer rest
        (var, rest) -> TokenVar var : lexer rest

main = getContents >>= print . calc . lexer
}

```