Prelude Lists
See Week 2 slides for type signature etc for prelude list functions.
Haskell Syntax
Case sensitive; these are all differenet names:
ab, aB, Ab, AB

- Variable Identifiers (varid) start with lowercase e.g. myNAME
- Constructor Identifiers (conid) start with uppercase letters e.g. Tree
- Variable Operator (varsym) start with any symbol and continue with symbols and the colon e.g. — : —
- Constructor Operators (consym) start with a colon and continue with symbols and the colon e.g. : + :

Operators

- Some operators are left associative such as +, -, *, / e.g. a+b+c parses as (a+b)+c
- Some oeprators are right associative such as &&, —— e.g. a:b:c:[ ] parses as a:(b:(c:[ ]))
- VOther operators are non-associative such as ==./=,¡,¿,¡=,¿= e.g.a¡= b ¡= c is illegal but (a ¡= b) && (b ¡= c) is ok

Functions

- Function application is denoted by juxtaposition and is left-associative
- f x y z parses as ((f x) y) z
- if we want f applied to both x, and to the result of the application of g to y, we must write f x (g y)
- In types, the function arrow is right associative Int -¿ Char -¿ Bool passes as Int -¿ (Char -¿ Bool)
- The type of a function whose first argument is itself a function must be written as (a -¿ b) -¿ c

Variable Declaration

- This can be either a function or a pattern
- A declaration can also have a lhs that is a single pattern: patn = expr