Pre-defined types

• We already saw function types: \( a \rightarrow b \)

• We also saw elementary types: \( \text{Int, Float, Double, Char} \), and so on

• Tuples group multiple types: \( (), (a, b), (a, b, c) \), and so on

\[
\text{harmonicMeanT} :: (\text{Double, Double}) \rightarrow \text{Double} \\
\text{harmonicMeanT} (x, y) = \frac{2 \times x \times y}{x + y}
\]

\[
\text{harmonicMeanT} :: (\text{Double, Double}) \rightarrow \text{Double} \\
\text{harmonicMeanT} \ pxy \\
= \frac{2 \times (\text{fst} \ pxy) \times (\text{snd} \ pxy)}{(\text{fst} \ pxy) + (\text{snd} \ pxy)}
\]

\[
\text{fst} :: (a, b) \rightarrow a \\n\text{snd} :: (a, b) \rightarrow b
\]
Lists

- Lists are either empty: [ ]

- ...or consist of a head and a tail: x : xs

- Lists are homogenous — all elements in one list have the same type

- List are parametric — different lists may contain elements of different type
Some operations on lists

- Length of a list
- Concatenating two lists
- Reversing the elements of a list
- Mapping a function over a list

In Haskell!