Chapter 5 - List Comprehensions

→ In Haskell new lists can be constructed using old lists

Ex: `[x*2 | x ← [1..5]]` → will square each of them

→ In list the order matters

→ `x ← [1..5]` This is a generator, comprehensions can have multiple generators seperated by commas.

→ Changing the order of generators changes the order in the final list

→ Multiple generators are like nested for loops.

→ Dependant generators rely on values from earlier generators

Ex:
```
concat :: [[a]] -> [a]
concat xss = [x | xs ← xss, x ← xs]
```
`concat [[1,2,3],[4,5],[6]]`
↳ returns `[1,2,3,4,5,6]`

→ Guards can restrict values from earlier generators

Ex: `[x | x ← [1..10], even x]`
↳ filter / remove

→ String comprehensions

Ex: `"abc" :: String` → List of chars for Haskell

Exercise in `listComprehensionsExercises.hs`