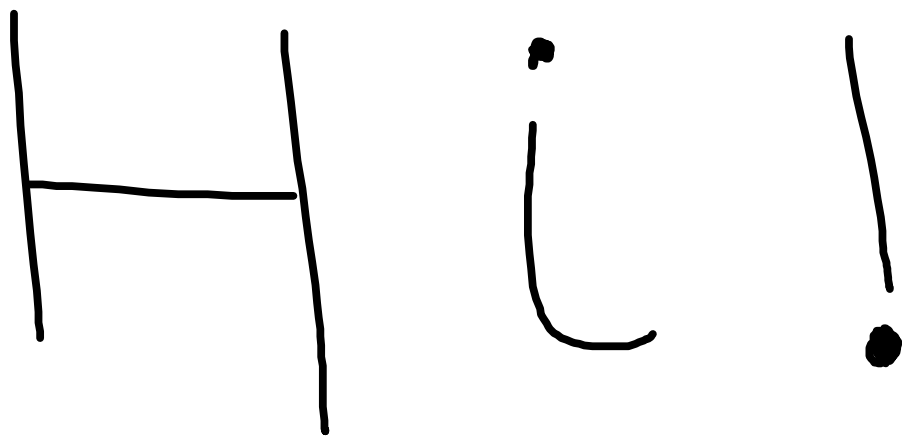
Functional Languages
5th Lecture

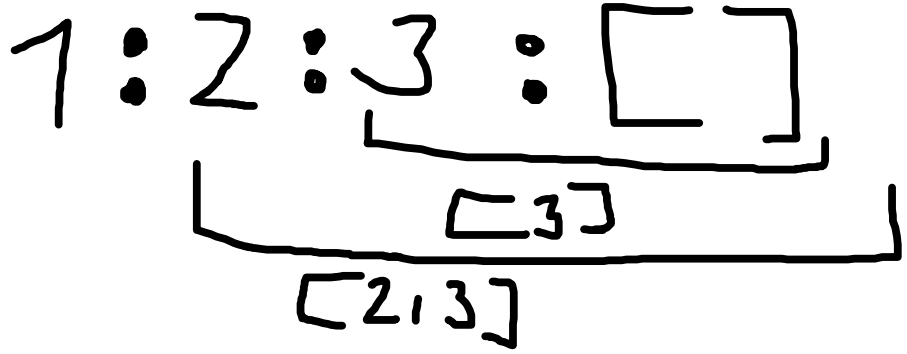


Lists recap

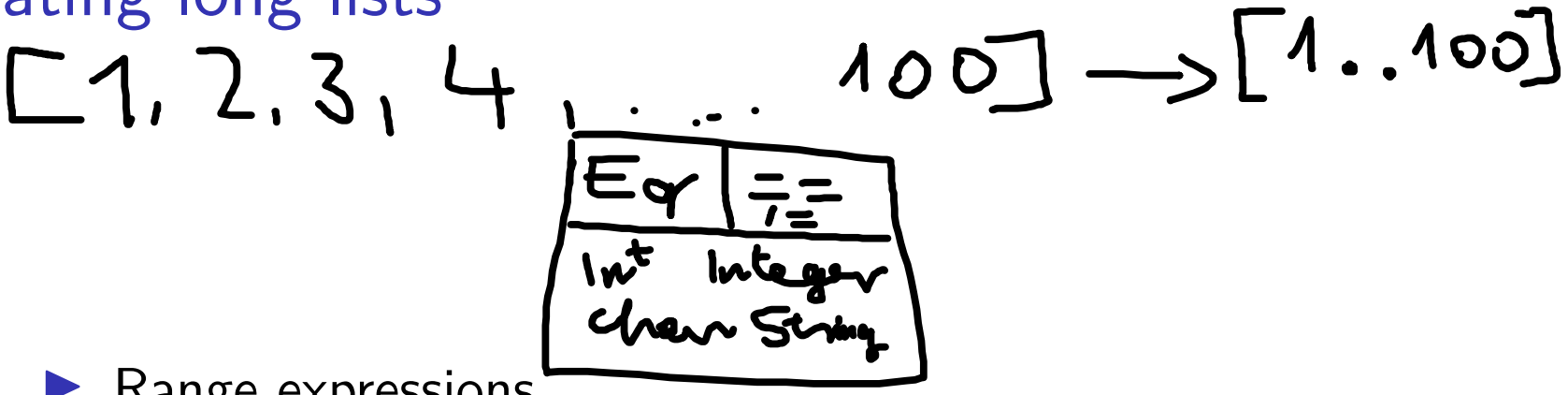
[1,2,3] ✓ [True, False] ✓

[1, True] ✗

- Homogenous data structure
- Every list can be built using `[]` and `(:)`
- Type is written in brackets. e.g. `[Integer]`

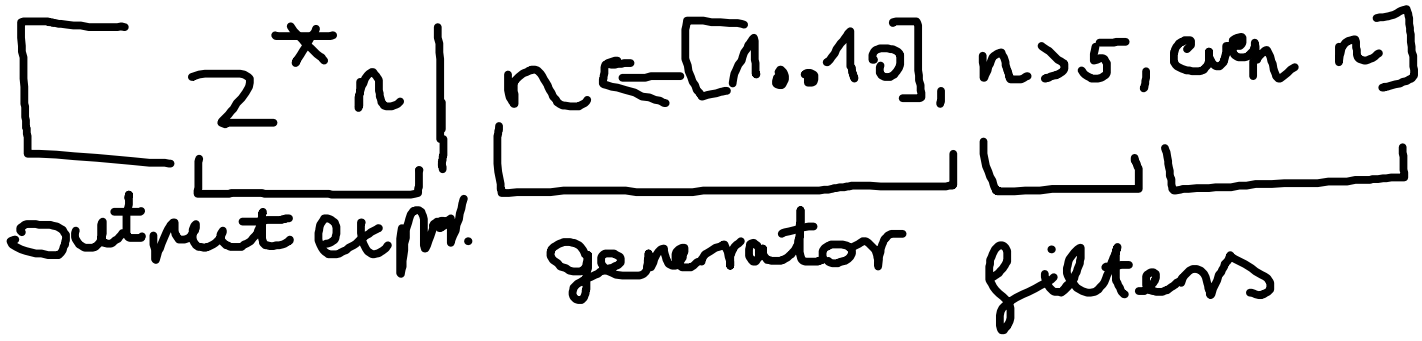


Creating long lists

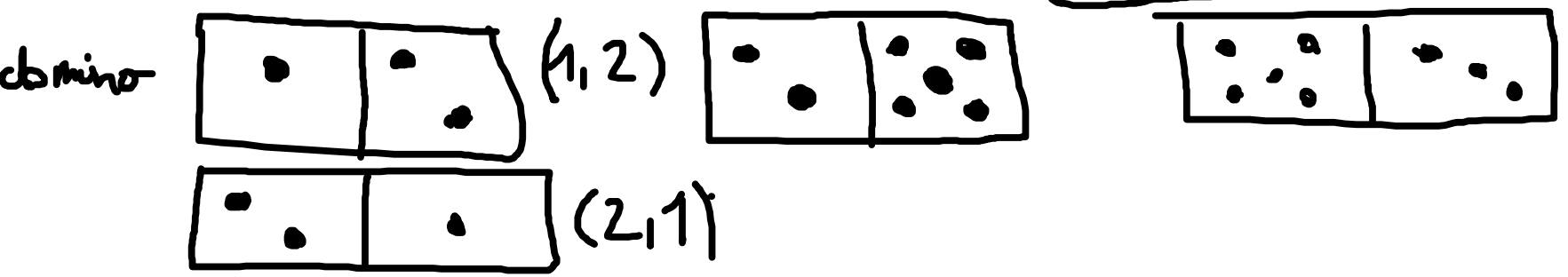


- Range expressions
- Can be used with any type in `Enum` type class
- Check with `:info Enum` in ghci
- Many forms:
  - increasing, decreasing
  - finite, infinite
- Example: odd, even numbers, alphabet

List comprehensions



- Creating list from one input list
- Can filter elements of the input list
- Parts: output expression, generator, filters (optionally)
- Example: square numbers, divisors, positive integers [1, 10, 2]



List comprehensions

- Creating list from one or more input list
- Can filter elements of the input lists
- Cartesian product of input lists
- Parts: output expression, generators, filters (optionally)
- Example: hour-minute pairs, dominoes, nested list of divisors

List comprehensions with `Strings`

- Remember, `String` is `[Char]` in Haskell
- Can be input of a list comprehension
- Example: upper case letters, upper case to lower case, all letters are upper case, count words