



- · records
- · cardinality
- · list all severs
- map filter Henaming import modules
- · sum types
- · expoung
- · howbig
- · predicate
- · function types e point-free style
- · just , maybe

· wrong use records: field in func signature

Lale 1

-NO looping mechanisms

> fact 5 -1 = fact 5 -1 fact(5-1) = fact 4







- - · predicate
- · function types
- point-free style · just, maybe

· wrong use records: field in func signature

Lale 1

- -NO looping mechanisms > fact 5 -1 = fact 5 -1 fact(5-1) = fact 4
- tail recurring returns something computed directly or its recurring call - use additional accumulator - pass 1 as acc each time

Lale 2

TUPLES - heterogenous = contain different types of data







00:52 🚐

·미 중 (34)

All iCloud



Lale 2

TUPLES heterogenous = contain different types of data

- at most 3

RECORDS

- wellection of named fulds - similar to STRUCT

I full Name person = person first ++"

TYPE ALIASES

- give a new name to existing types

full Name User - string

TYPE DEFINITIONS

allow reation of new types

ex: type Color = R 1 b 1 B 1

type constructors

can do line, one tune lout dill kinds

O.









can define one type but diff kinds of type cans ex: type Poent-Point Int Int com name

For SUM TYPES

cardinality = nr. of variants

ix: For () is 3

ex: Int how card = 2

PRODUCT TYPES cardinality = product of the cardinality of each field ex: For (2) is $2^{32} * 2^{32}$

LET IN

- declare leindings and use them in local appe

- used to avoid shadowery dr: double n=n+2

let n=10 in double n

- keep helper functions local

PATTERN MATCHING works like switch actact data from variants order matters (TOP-DOWN) at least one pattern must match







Û



Ocannot define Int as sum type Ocardinality (Bool) = 2

Lale 3

-louercase letters = type variables

-louercase names = type constraints

TYPE VARIABLE?

TYPE CONSTRAINTS

rumber (INT and FLOAT)

comparalle(mr, char, str.)

MAYBE

-enumerated types pwell def res

type Maylee a = Just a Nothing

Long result

RESULT
- signal the failure

type Result irror value = 0 k value

[Err error

00







00:52 🚍

ııl 🛜 34

All iCloud

 \bigcirc



the error

ex: safetrea: Shape -> Result String Float

safetrea shape of

Circle radius ->

if radius co thin

Err " radius 10"

Use

OK (pi # radius # radius)

LISTS

-singly linked lests

-cannot have a list of diff types

Definition: [1, 2, 3]

1::2::3::[]

Functions

·length

· sum

· head: 1st elem

· tail: items after 1 st

· range: List range 15

· hereerse

. append keyword = " ++"

0-

(0)





- · concat
- " = cons operator (used to add an element to the front of

- · drop = drops the specified on of elem-from the beggining of the list and returns a list with remaining elements
- · take = lets the specified nr. of elements in a new list
- · Lip = new lists which contain the first from each tuple

· untip = new lists st: primul au primul

· may = applies the given function to each element and puts the result in a new list

ex: lengths = List map string length quardian List. filter((x > x (6) lengths

1x -> String contains "-" x 2 as long







00:52 🚐

ııl 🛜 (34)

All iCloud





1x → String contains "-" x } as long

String contains "-" x } as long

string contains "-" > the last

arg.

· filter

ex: is Valid char = char /= "-"

String filter is Valid "2225"

if is Valid = true =) keep

if not =) delete

List filter ((X -> x 26))

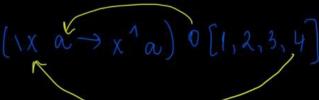
· foldl = reduce a lest

List foldl (1x a -> x +a) 0 [1,2,3,4]

compleining [initial list that
punction [value needs folding
Passes the first element as 1st arg

TAIL REC

· foldr= reduce a list from right



NOT TAIL REC

· all, + amil

00











- · all + any all is Even [2,4] == True
- · EXPOSING A FUNCTION module Date exposing (function 1)
- · EXPOSING a type + all variants module Date exposing (function1, Month(...))
 - · EXPOSING only a type module Date exposing (Date)
- Renaming: import Date as D. daysInMonth
 - Deleug. todo + string => (can replace any fund or arg; you want to implem. a fune. leut wont use it high aneay)
 - · fligher order functions - take fas args or ret a f. -> paranteze

(

righ aneay)

- · fligher order functions - take fas args or ret a f. -> paranteze
 - · POINT FREE STYLE hide param f is applied to

(<u>o</u>

· LAMBDA - anonymous of used as args to other