Introduktion til Programmering og Problemløsning (PoP)

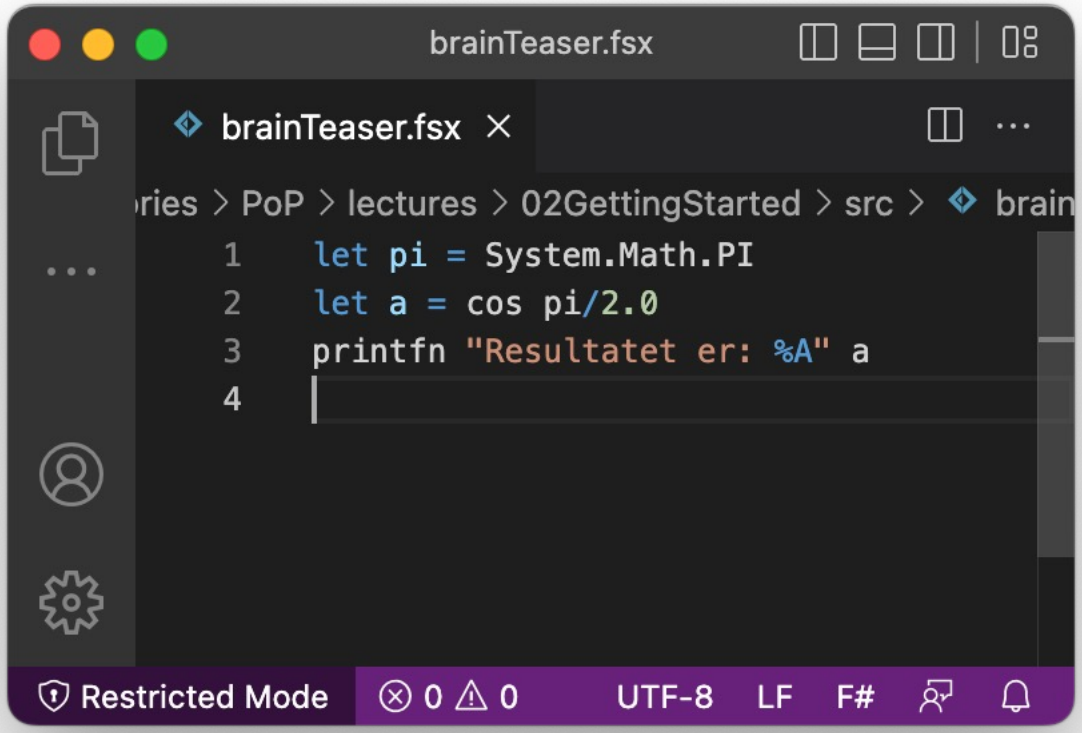
Option typer

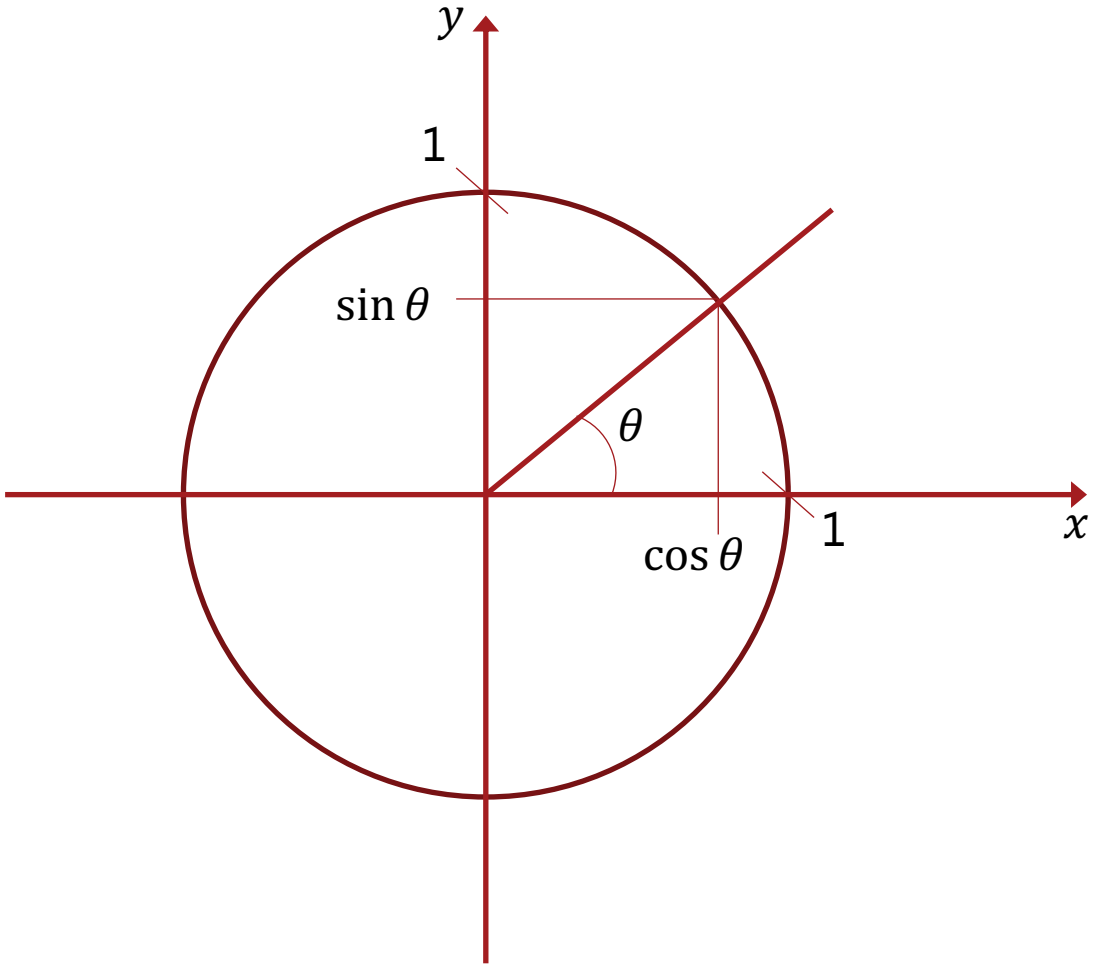
Jon Sporring
Department of Computer Science
2022/09/12

UNIVERSITY OF COPENHAGEN

Ugens opgave

https://tinyurl.com/242jdzmw





Besvar følgende med den du sidder ved siden af:

https://tinyurl.com/yykhuczh

areaAnnulus.fsx

areaAnnulus.fsx

Users > jrh630 > repositories > PoP > lectures > 03ValuesTypesCom

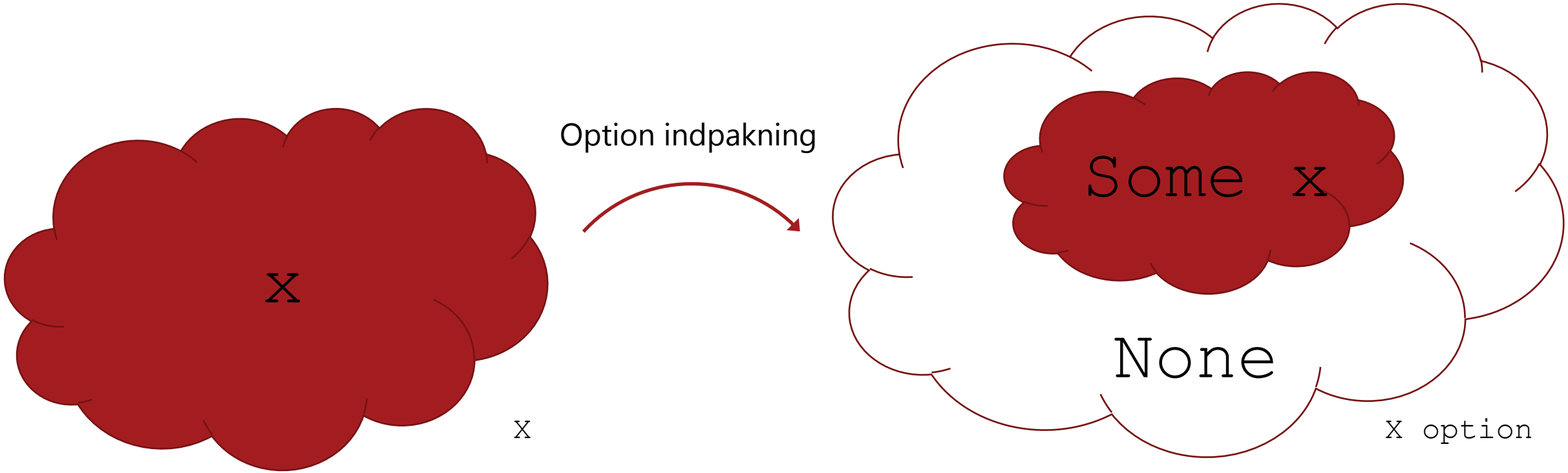
```
1  let areaCircle r =
2    printfn "%A" (System.Math.PI * r * r)
3  let areaAnnulus R r =
4    printfn "%A" (areaCircle R - areaCircle r)
5  printfn "%A" (areaAnnulus 3.5 1.5)
6
```

Restricted Mode ⊗ 0 ⚠ 0 Spaces: 2 UTF-8 LF F#

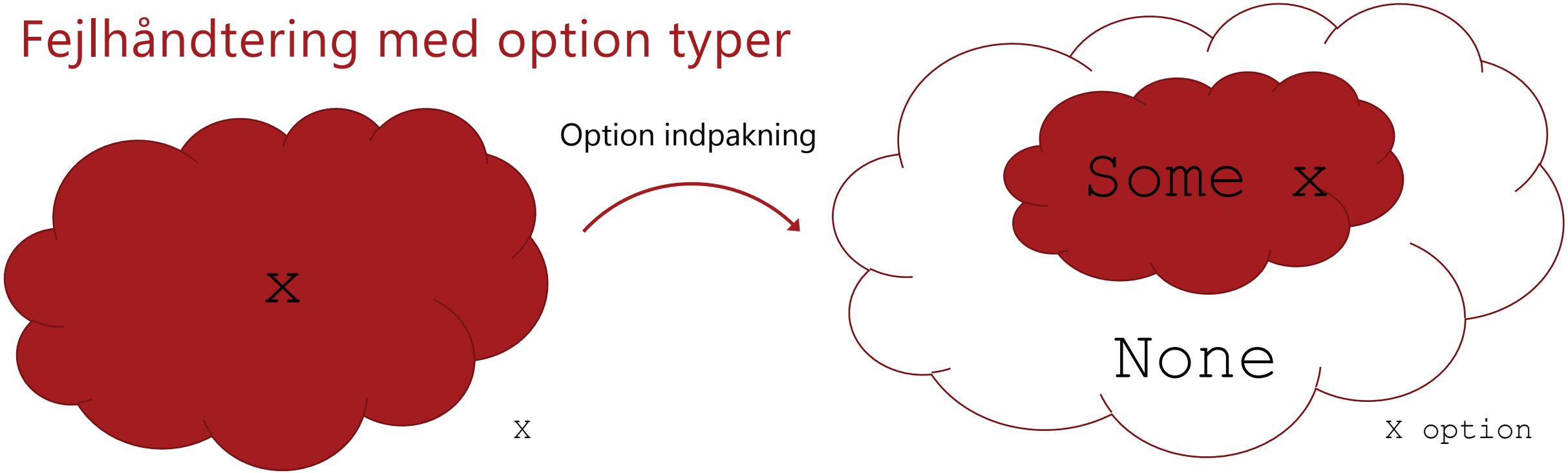
Fejlhåndtering med option typer

Divisions med 0

```
> let div a b = a/b
- div 3 0;;
System.DivideByZeroException: Attempted to divide by zero.
   at <StartupCode$FSI_0002>.$FSI_0002.main@() in
/Users/jrh630/repositories/PoP/lectures/03ValuesTypesComments/stdin:line 2
Stopped due to error
```



Fejlhåndtering med option typer

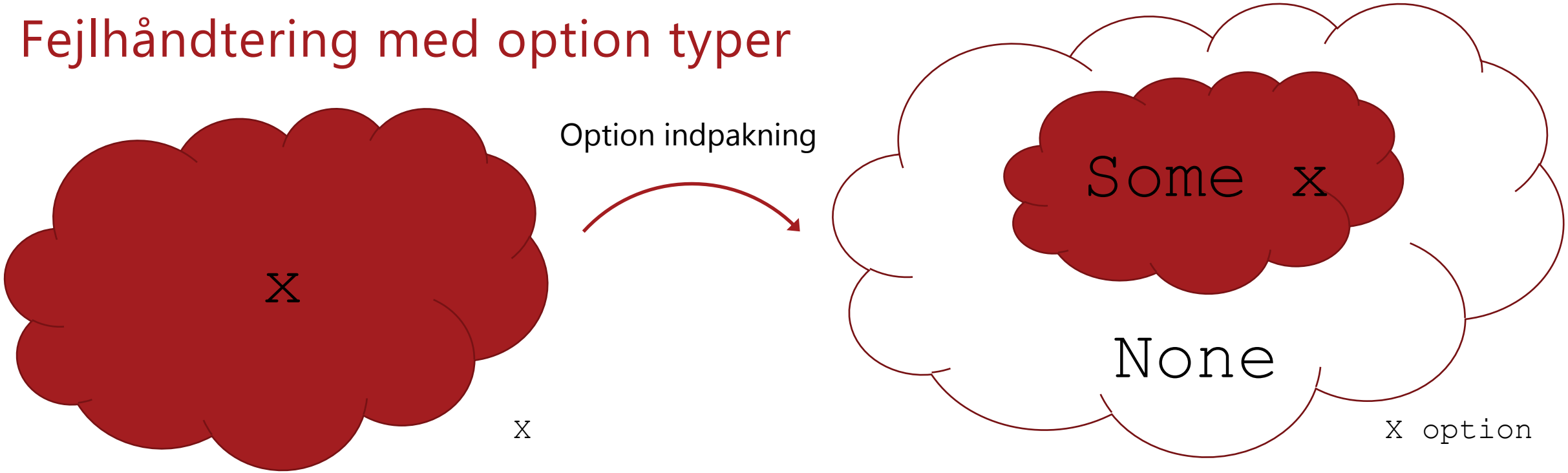



```
> type packet = Error | Value of int;;
type packet =
  | Error
  | Value of int

> let a = Error;;
val a: packet = Error
> let b = Value 3;;
val b: packet = Value 3
```

Option typen kan holde vilkårlig type

Fejlhåndtering med option typer




```
> let div a b =
-   match b with
-     0 -> None
-     | _ -> Some (a/b)
- div 3 0;;
val div: a: int -> b: int -> int option
val it: int option = None
```

```
> let print (x: 'a option) =
-   match x with
-     None -> printfn "Divide by 0 error"
-     | Some y -> printfn "%A" y
- div 3 0 |> print
- div 3 2 |> print;;
Divide by 0 error
1
val print: x: 'a option -> unit
val it: unit = ()
```

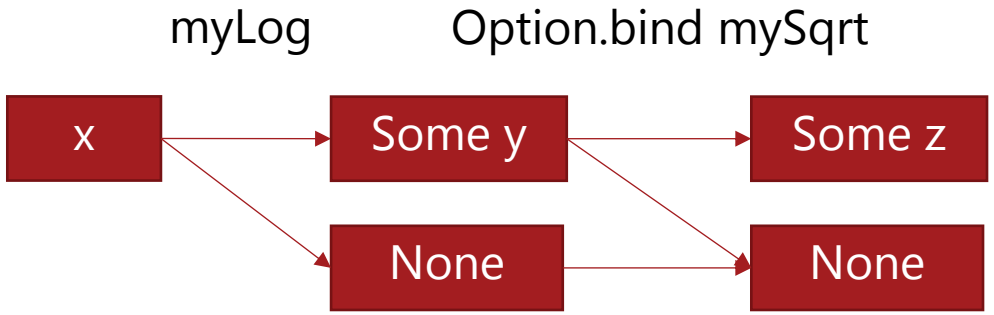
Sammensætning af funktioner f: ‘a -> ‘b option

Følgende funktioner håndterer evt. fejl med option-typer:

```
let myLog (x: float) : float option =
  if x > 0.0 then Some (log x) else None
let mySqrt (x: float) : float option =
  if x >= 0.0 then Some (sqrt x) else None;;
```

Direkte sammensætning er besværlig:

```
let mySqrtLog x : float option =
  let logX = myLog x
  match logX with
    None -> None
    | Some y -> mySqrt y
mySqrtLog 1.0;;
val it : float option = Some 0.0
```



Denne funktionalitet er allerede tilgængelig med Option.bind funktionen:

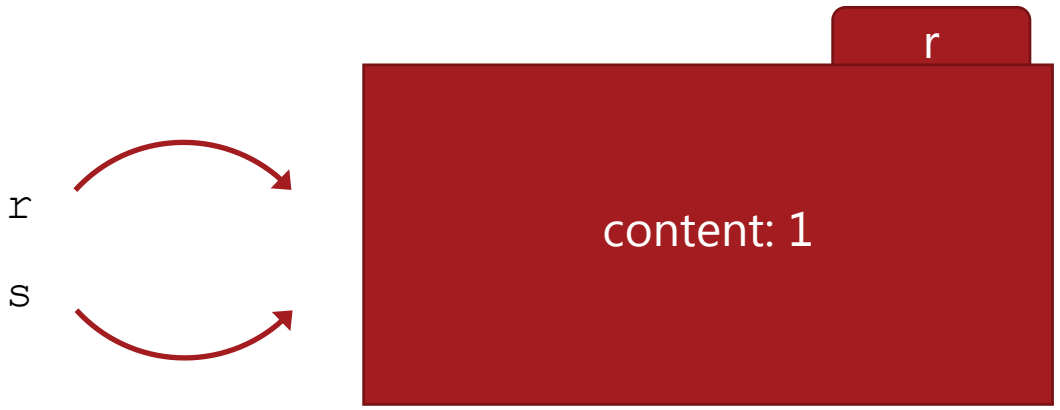
```
val bind : (('a -> 'b option) -> 'a option -> 'b option)
1.0 |> myLog |> Option.bind mySqrt
val it : float option = Some 0.0
```

Aliasing: referencekopi og ikke værdikopi!

Records er reference types

```
> type record = {mutable content: int}
- let r = {content = 1}
- let s = r
- r.content <- 2
- printfn "%A" s;;

{ content = 2 }
type record =
  { mutable content: int }
val r: record = { content = 2 }
val s: record = { content = 2 }
val it: unit = ()
```



Resumé

I dag har vi talt om:

- Fejlhåndtering med option typer
- Records og aliasing