Topic 2, currying

- Uncurried functions: One function, many arguments
- Curried functions: Many functions, one argument apiece

En curried funktion tager mere end et arg men tar dem et af gangen.

A curried function is a function that takes multiple arguments *one at a time.* Given a function with 3 parameters, the curried version will take one argument and return a function that takes the next argument, which returns a function that takes the third argument. The last function returns the result of applying the function to all of its arguments.

Eksempel hævekort:

atm = \x -> (\y -> (\z -> w))

atm = \kort -> (\pin -> (\beløb -> dkk))

Eksempel Elm:

multThree : number -> number -> number -> number

multThree x y z = x * y * z

virker sådan her:

multThree : number -> (number -> (number -> number))

```
addStuff :: Integer -> Integer -> Integer
addStuff a b = a + b + 5

let addTen = addStuff 5
let fifteen = addTen 5
```

```
fifteen
```

15

addTen 15

25

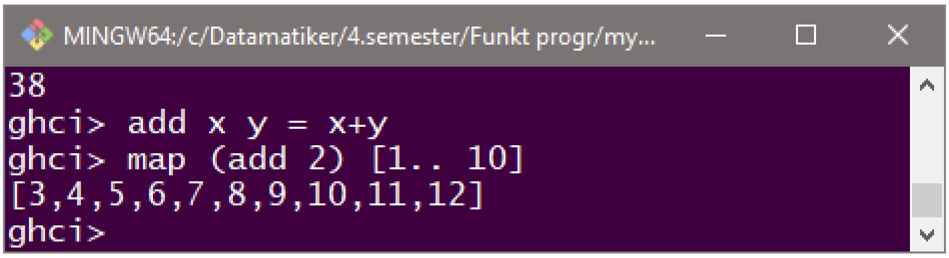
addStuff 5 5

15

addTen med et arg giver det ene manglende argument til addStuff, kalder man addTen med arg 15 får man 5 + 5 +15 = 25.

addFifteen giver et arg til addTen som tilføjer et arg og kalder addStuff med disse to args.

Esksempel Haskell:



```
38
ghci> add x y = x+y
ghci> map (add 2) [1.. 10]
[3,4,5,6,7,8,9,10,11,12]
ghci>
```