

# 1 Forelesning 1

## 1.1 Plan for forelesningen

- Gjennomgang av forventninger
- Online ressurser
- Eksempler på programmer i Haskell
- Funksjoner i Haskell
- Strukturen til et Haskellprogram

## 1.2 Online Haskellressurser

- *Learn You a Haskell*: [learnyouahaskell.com](http://learnyouahaskell.com)
- *Haskell* *wikibok*: [en.m.wikibooks.org/wiki/haskell](http://en.m.wikibooks.org/wiki/haskell)
- *Hoogle*: [hoogle.haskell.org](http://hoogle.haskell.org)
- *Mer*: [haskell.org/documentation](http://haskell.org/documentation)

## 1.3 Eksempler på programmer laget i Haskell

- Pandoc
- Xmonad
- Darcs
- GF - Grammatical Framework
- GitHub's semantic tool

Og andre diverse selskaper som Standard Chartered og Klarna.

## 1.4 Funksjoner

Hva er en funksjon? Vi bruker en funksjon ved å få en verdi ved å gi den et argument.

I matematikken brukes  $f(x)$  for å bruke en funksjon  $f$  på en verdi  $x$ . Hvis funksjonen tar imot flere argumenter skriver man  $f(x, y, z)$  for å gi dem.

I Haskell droppes parantesene, og man skriver bare `f x`, og dersom det er flere argumenter skrives det `f x y z`. For å sette sammen funksjoner, må vi likevel bruke paranteser: `f (g x)`. Dersom vi hadde skrevet `f g x` ville vi gitt to argumenter til funksjonen.

## 1.5 Haskellprogrammer

Filnavn i haskell slutter på `.hs` - ellers er hver fil ofte en *modul*, hvor filnavn ofte er det samme som modulnavn. Modulnavn kommer øverst i filen, og er på formen `module moduleName where`. Verdien `main` er en spesiell verdi som har typen `IO ()`. For å lage en kjørbar fil må `main` verdien ligge i modulen `main`.

### 1.5.1 Presidentsregler

I Haskell binder funksjonene sterkes, det vil si at koden under tolkes på følgende måte.

```
f x = x + 1 .1
f 1 * 3 = (f 1) * 3 = 12 .2
```