**Content**

[1. Intro to functional programming 2](#_Toc13831818)

[1. Lambda calculus 2](#_Toc13831819)

1. Intro to functional programming

Pure FPLs:

Haskell, Clean, Go, F#, ML /OCaml, Lisp/Scheme, Scala, Clojure, XSLT, Erlang, SQL, Matematica

* Functional programming is a programming paradigm – a way thinking about solving problems (toolset)
* Rooted in mathematics
* Language independendt (but Haskell is good )
* All computation is series of mathematical functions.

**Function**: mapping from input to output.

Cant change the inputs? Cant change the variables? – programming without variable reassignment. (nothing like x = x + 1, for (i=0; i<10; i+=)). Trade the immutability for some benefits. What are these?

Programs are logical, perfect for distributed computing –idea that we separate computation into lots of small parts that we give to a bunch of different computers to process. Coordinate bunch of processors – cloud computing.

X=5; blackbox(x) == blackbox(x) (in JS it can happen that will be not true, the first function can change something, some counter so the second gives different result)

* 1. Lambda calculus