



FPV Week5 - FPV Summary for the week 5

Funktionale Programmierung (Technische Universität München)

2 Syntax

- if-then-else: don't drop else!
- pattern matching: `(match a with b -> ... — c -> ...)` compiler does not easily deduct the type, since we don't add type at notations
- `begin...end`: useful to make codes more readable, better than the brackets
- declaration and assignment: `let a = b in...`, overwriting the previous variable when the same name used
- record: simple type with several properties, no function defined
- tuple: normal but **not recommended (by me)** necessary.

3 Functions

1. Idea: functions as variables, can be passed or a property
2. Definition: `let (rec) func a b c ... =`, where `func` is the name of the function, `rec` is the recursive sign, `a b c` are all variables concerned.
Important: don't pass variables as tuple into the function all, no comma is needed!
3. `function`: `function` is a keyword for pattern matching esp. for a function.
No difference just for preference.
4. `λs`: given by the keyword `fun`, `fun x -> your function body`