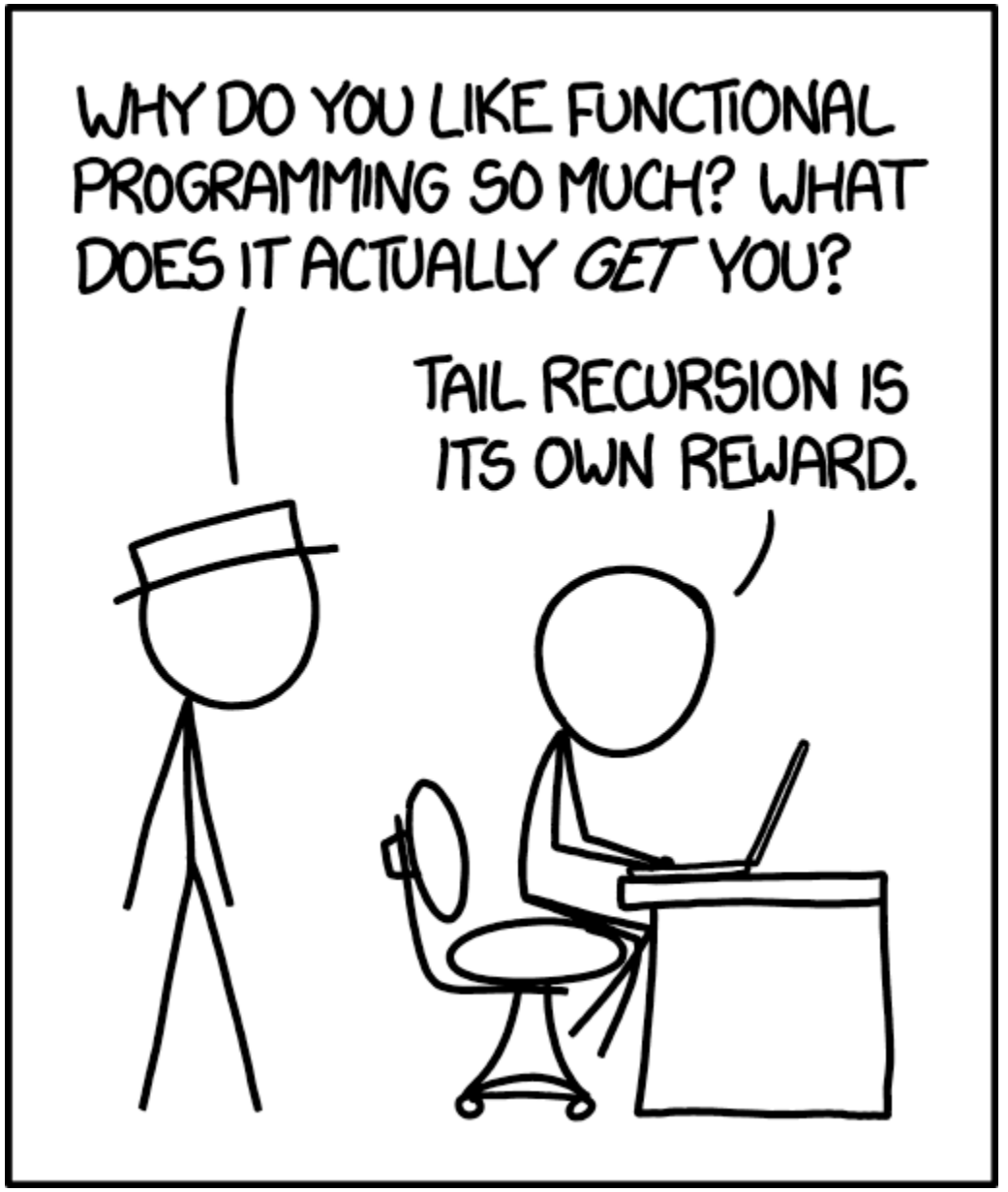
Hva er (greia med)

FUNKSJONELL PROGRAMMERING

?

Kjetil Valle

Fagkveld for Oslo-studenter
25. april 2017



xkcd.com/1270

pure

applicative

algebraic datatypes

currying

monoid

β-reduction

FP

existential quantification

functor

endofunctor

the Curry–Howard correspondence

ad hoc polymorphism

monad

homomorphism

method overloading

inheritance polymorphism

S.O.L.I.D.

dependency injection

visitor pattern

OOP

Liskov substitution principle

encapsulation

AbstractSingletonProxyFactoryBean.java

decorator

covariance

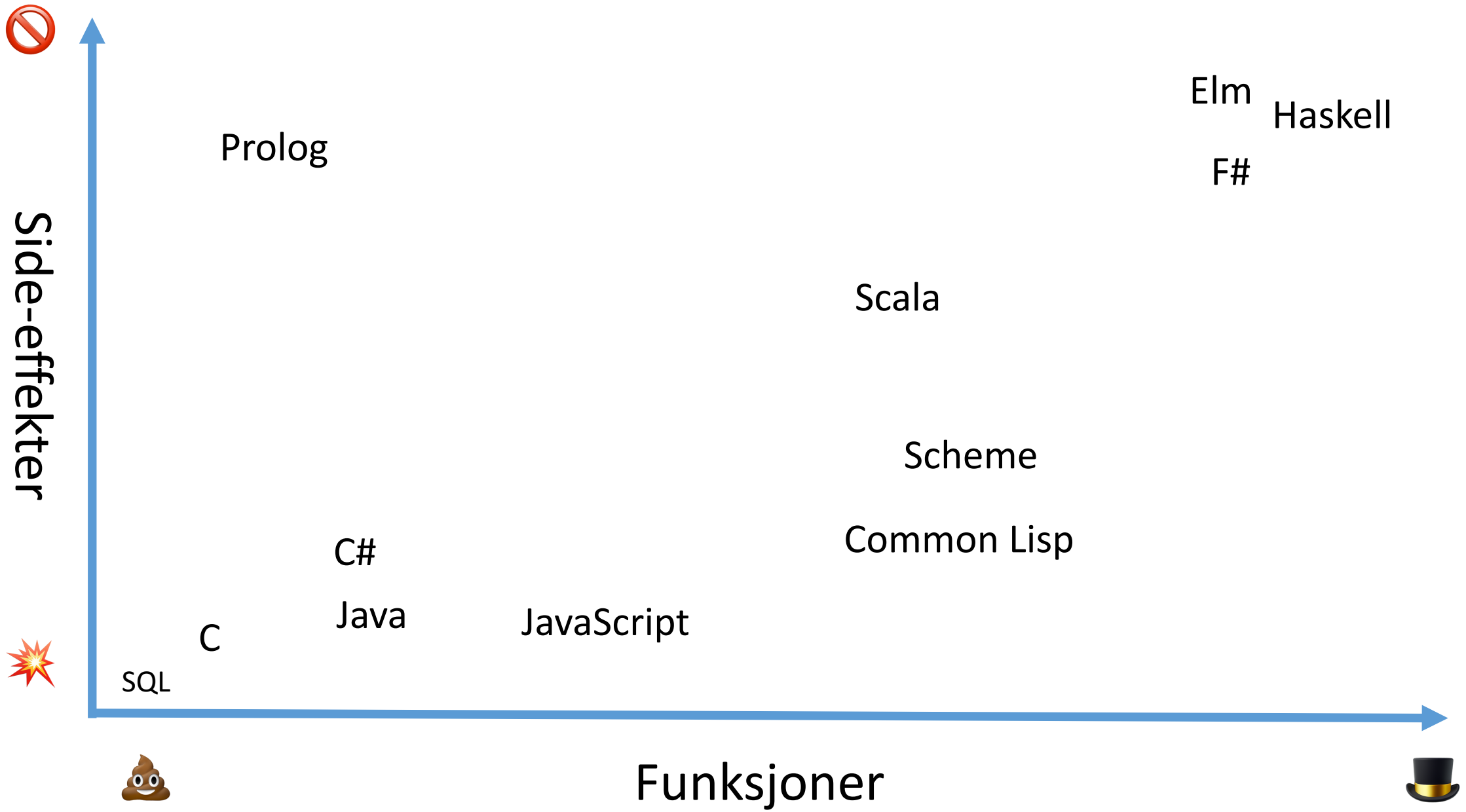
dependency inversion

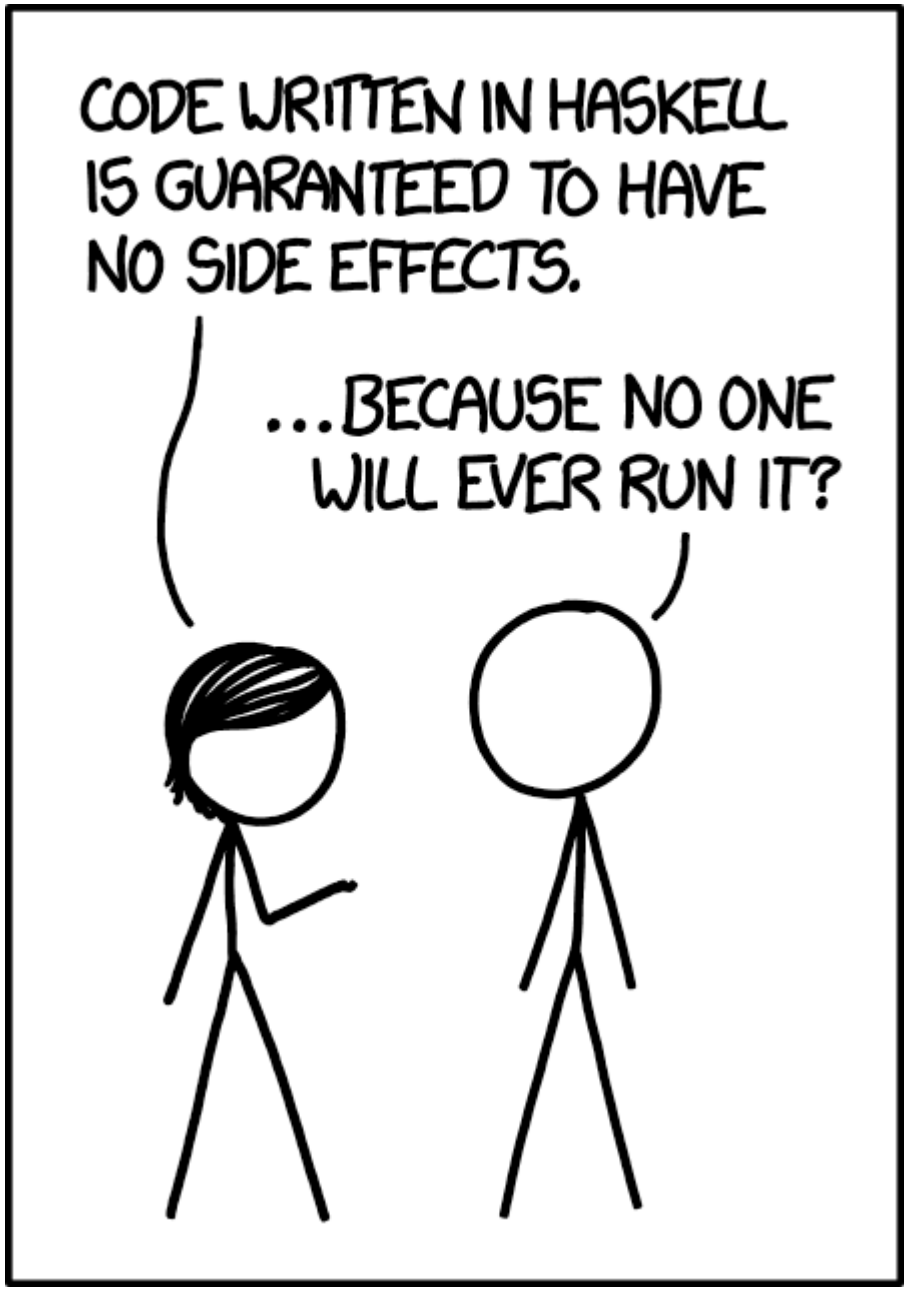
singleton

Så hva er FP *egentlig*?

Fravær av side-effekter

Fancypants funksjoner


Side-effekter
Prolog
Elm
Haskell
F#
Scala
Scheme
Common Lisp
C#
Java
JavaScript
C
SQL
Funksjoner



xkcd.com/1312

```
calc_area radius =
    fire_missiles [🚀,🚀,🚀,🚀]
    radius_squared = radius * radius
    return (2 * pi * radius_squared)
```

```
calc_area radius =
    fire_missiles [🚀,🚀,🚀,🚀]
    radius_squared = radius * radius
    return (2 * pi * radius_squared)
```

```
important_number = 4
// …
important_number = 5
```

```
important_number = 4
// …
important_number = 5
```

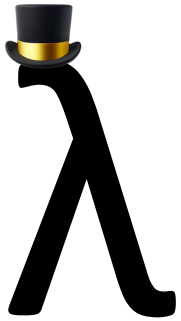
~~important_number = 4~~
~~// …~~
~~important_number = 5~~

```
a = [🐮, 🐷, 🐤]
b = a.add(🦄)

// a: [🐮, 🐷, 🐤]
// b: [🐮, 🐷, 🐤, 🦄]
```

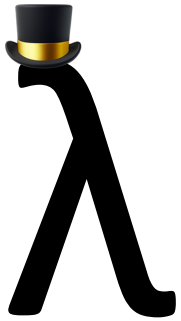
Koden blir…

Enklere å teste

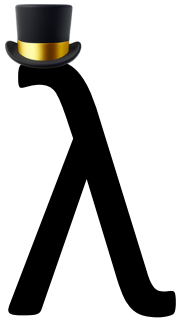
Enklere å resonere rundt



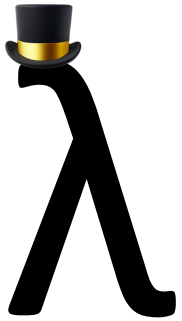
I will have your finest functions, sir!



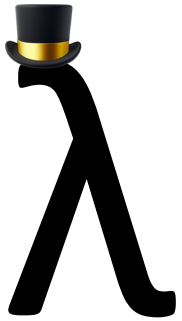
```
filter_odd numbers =
    odd_numbers = []
    for num in numbers:
        if is_odd num:
            odd_numbers += num
    return odd_numbers
```



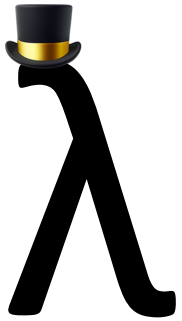
```
filter_odd numbers =
    odd_numbers = []
    for num in numbers:
        if is_odd num:
            odd_numbers += num
    return odd_numbers


filter_odd numbers =
    filter is_odd numbers
```



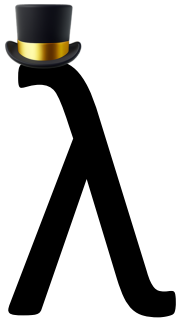
```
filter_odd numbers =
    odd_numbers = []
    for num in numbers:
        if is_odd num:
            odd_numbers += num
    return odd_numbers


filter_odd numbers =
    filter is_odd numbers



    filter is_odd
```



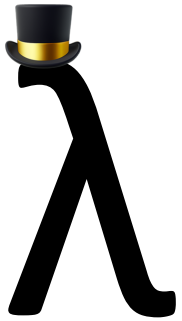
```
filter_odd numbers =
    odd_numbers = []
    for num in numbers:
        if is_odd num:
            odd_numbers += num
    return odd_numbers


filter_odd numbers =
    filter is_odd numbers


filter_odd =
    filter is_odd
```



$$h = f \cdot g$$



```
h = f · g

// h(x) = f(g(x))
```

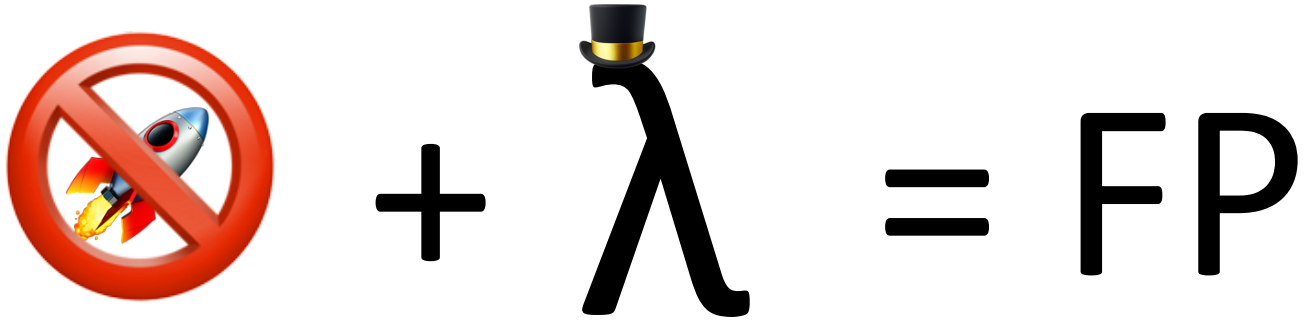


```
h = f · g

// h(x) = f(g(x))

is_even = not · is_odd
```

Koden blir…

Mer konsis

Enklere


+
λ
= FP