

Programmazione funzionale con Haskell

Daniele D'Orazio

Software Developer, Develer



Daniele D'Orazio daniele@develer.com

Risorse

- Slide: <http://dvlr.it/haskell-fp101>
- Repo git: <https://github.com/d-dorazio/haskell-fp101>



Daniele D'Orazio daniele@develer.com

Obiettivi

- Breve introduzione ai concetti principali della programmazione funzionale
- Prendere familiarità con il linguaggio di programmazione Haskell
- Realizzare un interprete Brainf*ck in Haskell!



Daniele D'Orazio daniele@develer.com

Perché imparare la programmazione funzionale?

- Dichiarativo
- Il codice è ~~tendenzialmente~~ più terso e facile da capire
- Facile da testare e da riusare
- Apre la mente
- [Why Functional programming matters](#)



Daniele D'Orazio daniele@develer.com

Fun, fun, functions!

$$f(x) \rightarrow y$$



Daniele D'Orazio daniele@develer.com

Functions assemble!

$$f(x) \rightarrow y$$

$$g(y) \rightarrow z$$

$$g(f(x)) \rightarrow z$$



Daniele D'Orazio daniele@develer.com

Caratteristiche della programmazione funzionale

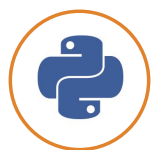
- Functions as first class citizens
- Immutabilità
- Funzioni pure (aka senza side effect invisibili)



Daniele D'Orazio daniele@develer.com

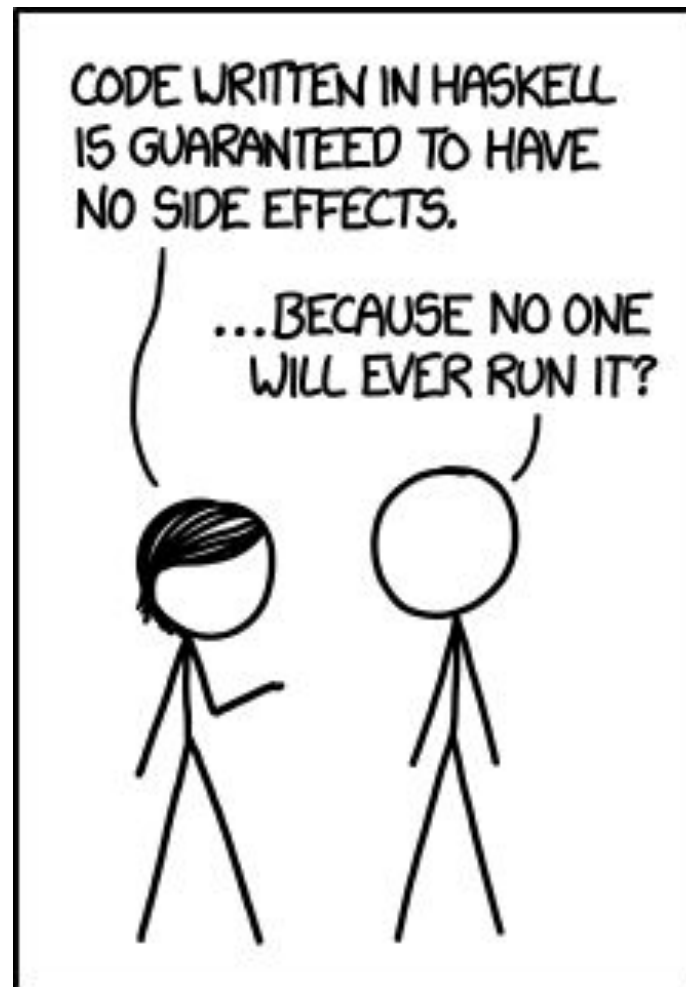
Haskell

- Purely functional
- Static type system
- Type inference
- Lazy evaluation



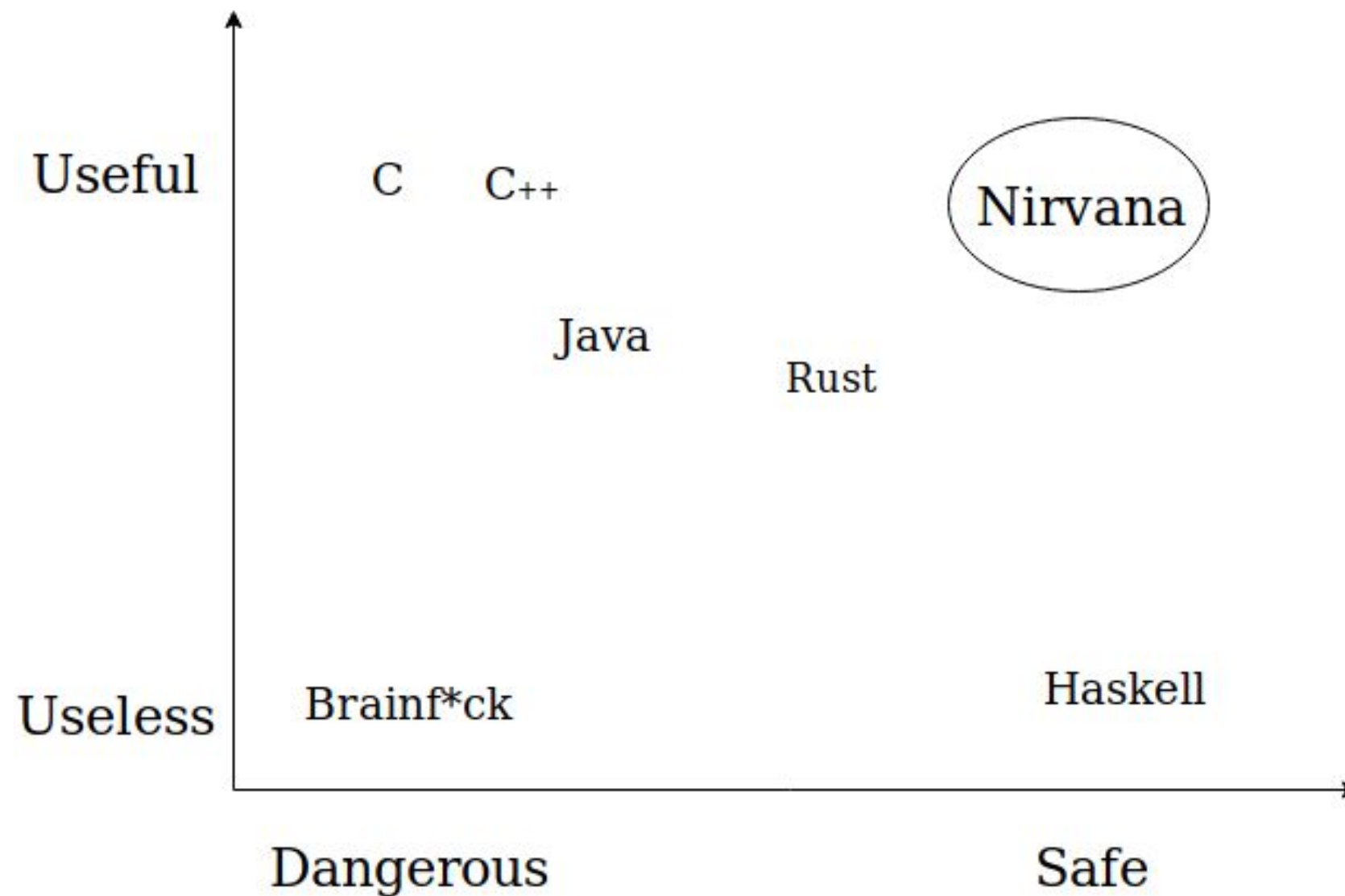
Daniele D'Orazio daniele@develer.com

Haskell humor #1



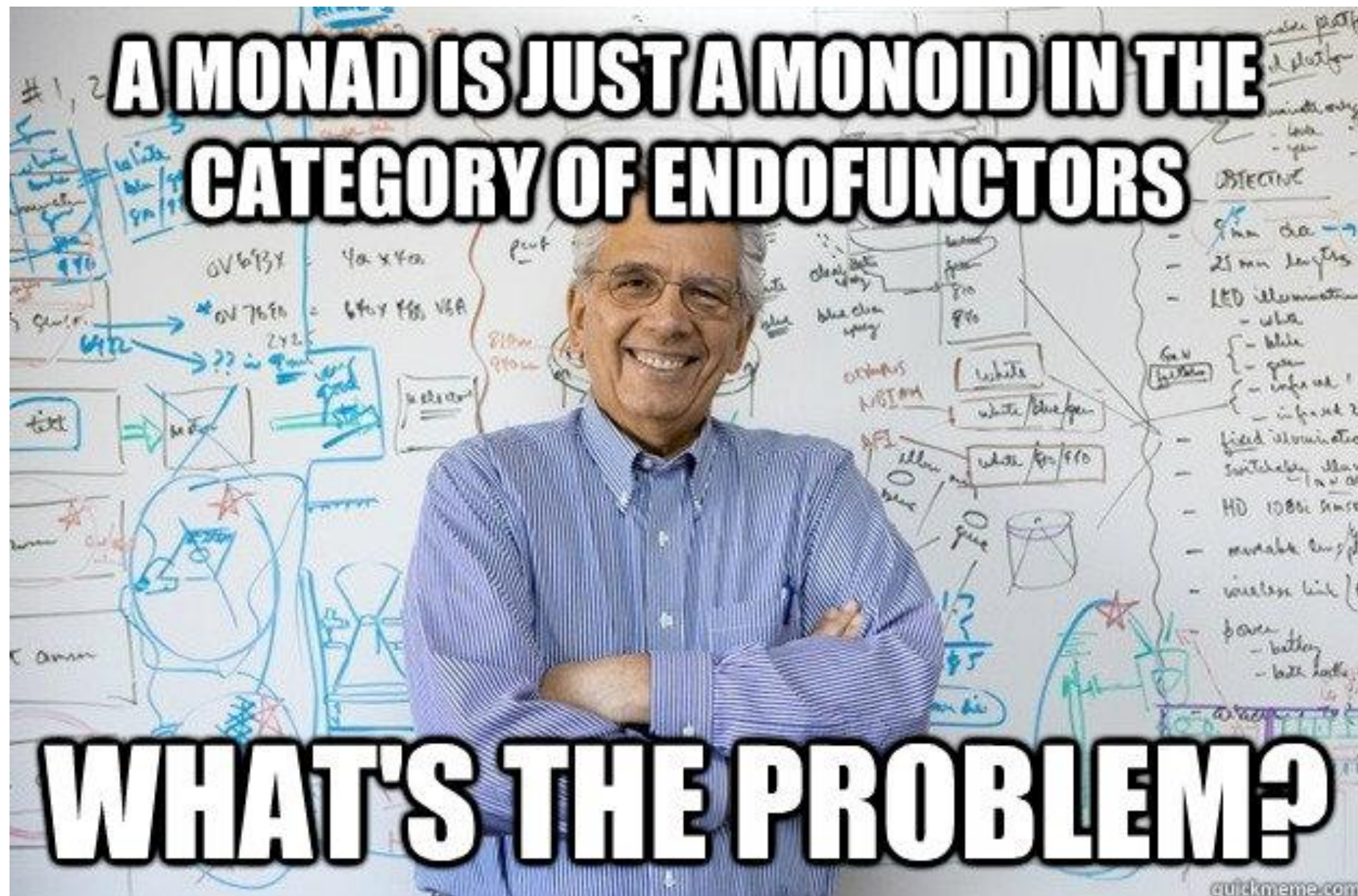
Daniele D'Orazio daniele@develer.com

Haskell humor #2



Daniele D'Orazio daniele@develer.com

Haskell humor #3



Daniele D'Orazio daniele@develer.com

Haskell humor #4



Daniele D'Orazio daniele@develer.com

Haskell 101



Daniele D'Orazio daniele@develer.com

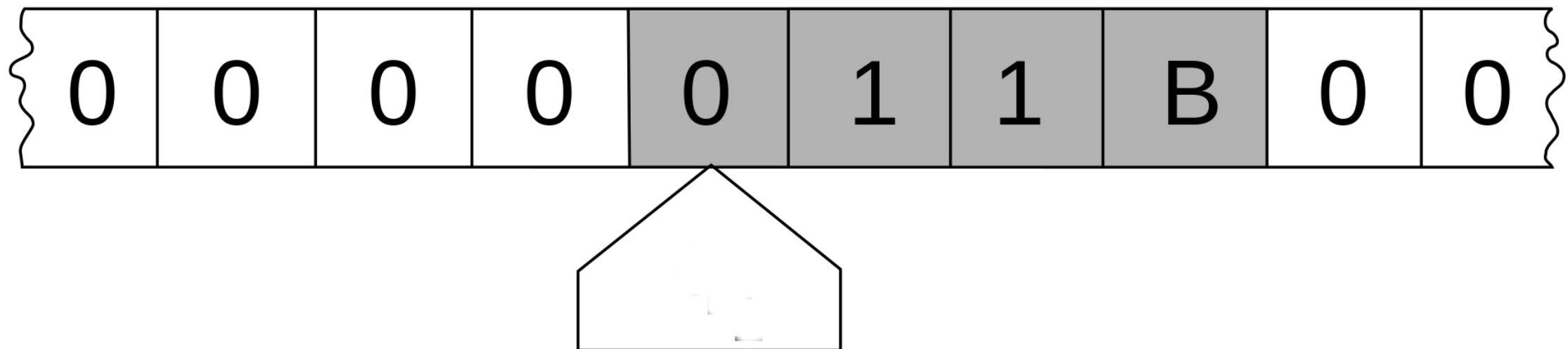
🔥 Brainf*ck 🔥



Daniele D'Orazio daniele@develer.com

Brainf*ck

>+[<+.>,-.



Daniele D'Orazio daniele@develer.com

CONTATTI

github

github.com/d-dorazio

e-mail

daniele@develer.com

web

develer.com/daniele-dorazio



Daniele D'Orazio daniele@develer.com