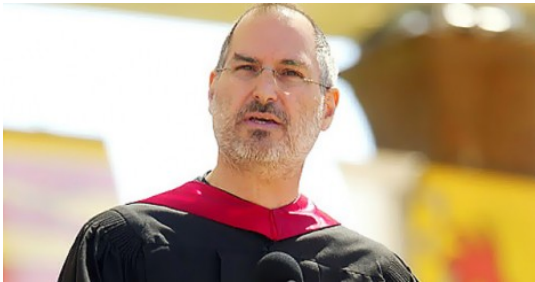# Using Haskell Professionally

Alfredo Di Napoli

% Using Haskell Professionally % Alfredo Di Napoli % 28th of March, 2015

Quoting Steve Jobs..

Today, I'm gonna tell you three stories.

- My story
- My company story
- Your story

My story

Back in 2007, I had my first exposure to FP, under the form of a university course. The course taught us OCaml.

Being focused on data structures and algorithm, and not on the "real world", I did lack the motivation of building things in it.

As a young undergraduate, I caught the OOP fever shortly after, and didn't look at OCaml again, despite finding it fascinating.

The fact my university was (**is**) mono-culture (i.e. Java), together with my refuse of the *status quo*, pushed me to "greener pastures", where I found Python, falling in love with it.

Fast-forward 2009..

I was still in love with Python. At the time I was a regular attendees of **Python-it.org**, a popular Italian community.

The "off-topic" forum room was the place where we talked about technology and every non-related Python topic, and more than once, a user I respected mentioned Scheme and its cousin **Clojure**.

I fell in love with Clojure.

Pushing myself forward in my holy grail search, I was exposed to a huge number of programming language, trying to find the "perfect" one: Scheme, Clojure, Common Lisp, **Haskell**, Io, Ruby, *put-yet-another-language-here*. I did look at Haskell with interest, but I find it quite arcane. I did order and started reading the "Real World Haskell" book, but I found it disappointing (at the time): "Why on earth did I need to wrap my values inside this *Just* thing, if then I need to unwrap it to inspect the value?" - I thought at time.

So, despite the interest, I did went back to Clojure and lisp-family languages.

Fast-forward 2011..

When the time of choosing my master degree's project, I had no doubt but asking to my professor Franco, which at the time was teaching "Parallel and Distributed Computing". I was extremely intrigued by the topic, and Franco revamped in me the holy flame of fast and compiled languages. So he said "Use C++, and live long and prosper".

Your story

What can **you** do and can **we** do as a community to embrace, support and spread functional programming?

Reject the status-quo. Keep an open mind. Be pragmatic.

Quoting Bret Victor's "The future of programming"

[..]So the most dangerous thought that you can have as a creative person is to think that you know what you're doing. Because once you think you know what you're doing, you stop looking around for other ways of doing things. And **you stop being able to see other ways** of doing things.[..]

[..] if you don't want to be this guy, if you want to be open or receptive to new ways of thinking, to invent new ways of thinking, I think the first step is you have to say to yourself, **"I don't know what I'm doing.** [..] I think you have to say,"we don't know what programming is. We don't know what computing is. We don't even know what a computer is."

And once you truly understand that - and once you truly believe that - then you're free.

Thank you.

Questions?