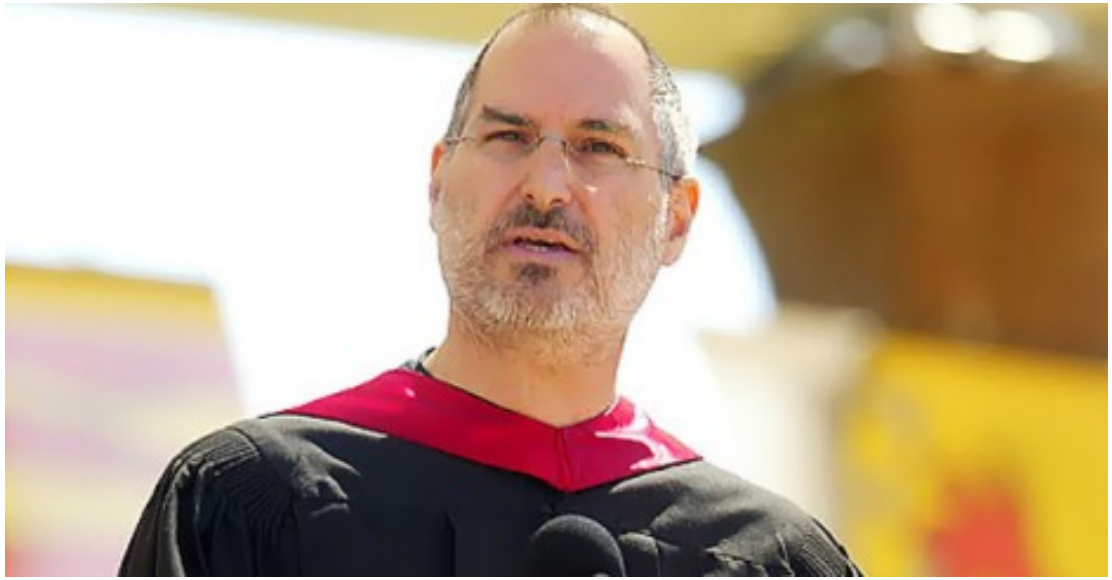
Using Haskell Professionally

Alfredo Di Napoli

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.

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.

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*.

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".

The Manchester era

The typical Sinatra hello world app...

```
require 'sinatra'

get '/hi' do
  "Hello World!"
  end
```

. . . and the Snap equivalent

```haskell
{-# LANGUAGE OverloadedStrings #-}

import Snap

main :: IO ()
main = quickHttpServe $ do
  route [("/hi", method GET $ writeBS "Hello World!")]
```

In a nutshell...

Haskell is a very pragmatic language...

In a nutshell...

Haskell is a very pragmatic language...

...but not all the people using it are!

Your story

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

Be pragmatic.

Keep an open mind.

Reject the status-quo.

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?