# F# Computation Expressions

and a whirlwind language tour

#### Whoami



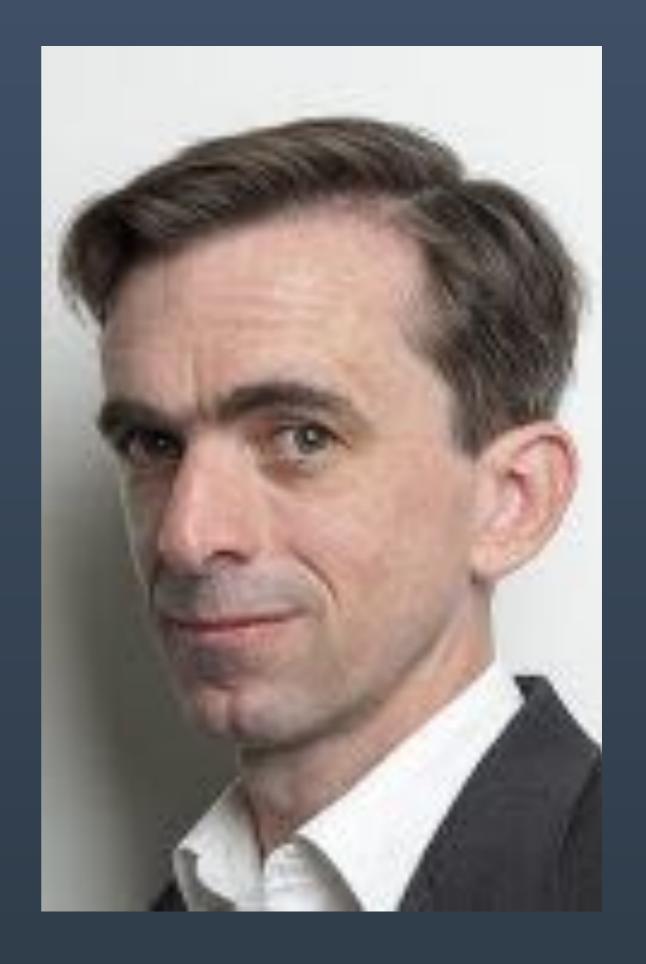
#### **Andrew McGuier**

- Software Developer/Consultant
- F#, Scala, C#, Python
  - Generally Friendly

#### F# in a Nutshell

F# is a mature, open source, cross-platform, functional-first programming language. It empowers users and organizations to tackle complex computing problems with simple, maintainable and robust code.

-fsharp.org



"I generally prefer arguments in utilitarian terms (bug reduction, safety under refactoring, stability of coding patterns under changing requirements, does a mechanism promote team-cooperation etc.)."

–Don Syme - Benevolent Dictator For Life

## High Level

- Functional
- Strongly/Statically Typed \*
- Immutable by default\*
- ML-Family, descended from OCaml
- Eager Evaluation \*\*
- Pragmatic/Hybrid (OO/.NET Interop prioritized)

## History

- Bootstrapped with the OCaml Compiler
- 2005 1.0 (.NET 3.5)
- Current Release 4.1

#### Myth 1: Vendor Lock-In

- F# compiler has been open source since 2.0
  - Currently MIT Licensed
- .NET Specification has been open
- Mono has always been an option
- Microsoft's .NET Implementations are now also open source under very friendly licenses
  - MIT Apache2

#### Myths 2: Target Platforms

- Windows (Full .NET Framework)
- Windows/Linux/OSX (.NET Core/Mono)
- Javascript (<u>http://fable.io</u>)
- IOS/Android (Xamarin Mono)

#### Myths 3: Editors

- "You have to have windows/visual studio"
- Ionide currently the most popular F# editor
  - VS Code/Atom (<a href="http://ionide.io/">http://ionide.io/</a>)
- Emacs/Vim support
- Jetbrains' Rider

#### Basic Language Features

- Type Inferencing
- Reified Generics
- Stack allocated structs
- Records, Classes, Interfaces, Inheritance
- Pipe Operator
- Currying
- Pattern Matching

# Quick Syntax Tour

## Lets get the boos early

- No Type Classes\*
- No scala style traits/ implicits
- No Higher-Kinded Types \*\*
- No Parameterized Modules (yet)
- Exceptions
- Null free, except when it's not\*\*\*
- Look at the F\* language

# Interesting Language Features

- Code Quotations
- Sequence/Query syntax
- Units of Measure
- Type Providers
- Computation Expressions

#### Computation Expressions

 http://tomasp.net/academic/papers/ computation-zoo/

# Coding Time

#### Examples In Practice

- Hopac (https://github.com/Hopac/Hopac)
  - Concurrent ML style concurrency in F#
- MBrace (<u>http://mbrace.io/</u>)
  - What lambda would like to be when it grows up
- Freya (<u>https://freya.io/</u>)
  - Functional/typesafe web stack

#### Thank You!



• andrew@echogatetech.com

#### Additional References

- Great tutorial on computation expressions (https://fsharpforfunandprofit.com/posts/ computation-expressions-intro/#series-toc)
- http://fsharp.org/
- Community Slack (<a href="http://fsharp.org/guides/slack/">http://fsharp.org/guides/</a>
   slack/)