Conclusion and Wrap-Up

Kit Eason

@kitlovesfsharp | www.kiteason.com

F# for the Win!

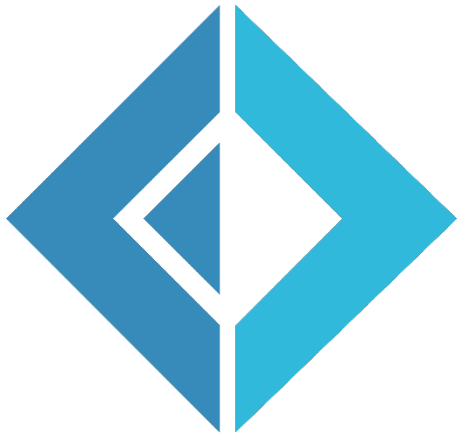
A CLI language

.NET/Mono

Visual Studio/Xamarin Studio

F# Interactive (FSI)

F# for the Win!



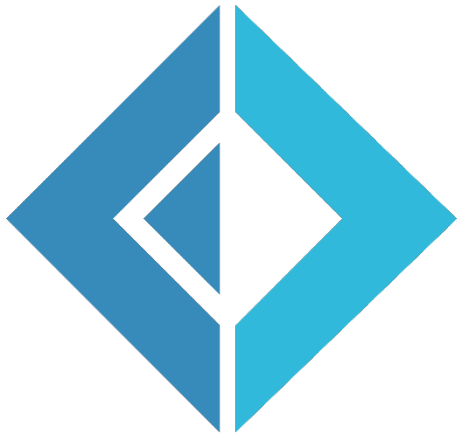
'let' to define values and functions

Functions return result of last expression

Indents instead of {}

```
let Area x y =
    x * y
```

F# for the Win!



Type inference, forcing types

If statements

For loops

```
let CircleArea (r : float) =
    if r > 0. then
        System.Math.PI * r * r
    else
        0.0
```

```
let GreenBottles start stop =
    for bottle in start .. -1 .. stop do
        printfn "%i green bottles" bottle
```

F# for the Win!

Arrays

[| … |]

‘yield’

a.[i]

```
let words = [| "the"; "quick"; "brown"; "fox" |]
let numbers = [| 0..99 |]
let primes =
   [| for i in 0..99 do
         if isPrime i then yield i |]
```

F# for the Win!

Array.init

Array.filter

Array.map

Array.iter

Array.sort

```
let candidates = Array.init 10 (fun n -> (pown 2 n) - 1)
let mersennes = candidates |> Array.filter isPrime

let words = [|"the"; "quick"; "brown"; "fox"|]
let lengths = words |> Array.map (fun w -> w.Length)
words |> Array.iter (fun w -> printfn "%s" w)
let sorted = words |> Array.sort
```

F# for the Win!

Tuples

Forward pipe |>

```
let TopWords (words : string[]) =
    words
    |> Seq.map (fun w -> w, w.Length)
    |> Seq.sortBy (fun (_, len) -> -len)
    |> Seq.truncate 10
```

F# for the Win!

Sequences

seq { … }

Seq.map, Seq.filter etc.

```
let (cols, rows) = (10, 10)
seq { for row in 0 .. rows - 1 do
          for col in 0 .. cols - 1 do
            yield (row, col, row*rows + col)
    }
```

F# for the Win!

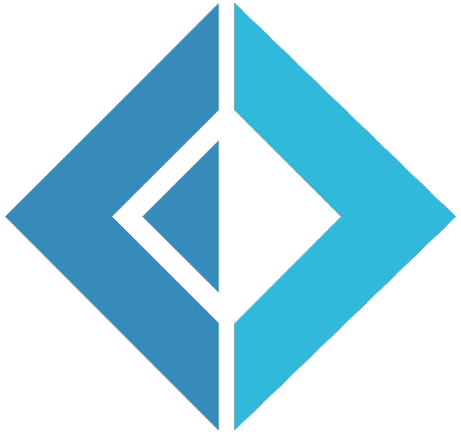
- Records
- Option types
- Discriminated unions
- Pattern matching

```
type Location =
   | Analog of band:string * freq:float
   | Digital of id:int
   override this.ToString() =
      match this with
      | Analog(b, f) -> sprintf "%.1f (%s)" f b
      | Digital id -> sprintf "%i (DAB)" id
```

```
type Channel = {
   name : string
   licence : string option
   location : Location
}

let heavyRockFM = {
   name = "Heavy Rock FM"
   licence = None
   location = Analog("FM", 98.3)
}
```

F# for the Win!



Immutability

Shadowing

‘mutable’

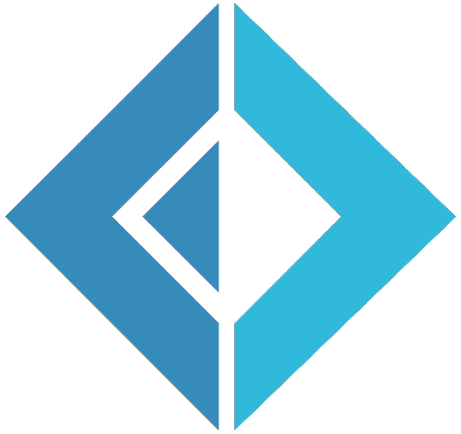
Reference cells

```
let s = 33
let s = 34 // Shadowed
```

```
let mutable m = 33
m <- 34
```

```
let r = ref 33
r := 34
printfn "%i" !r
```

F# for the Win!



OO types (classes)

Methods

Properties

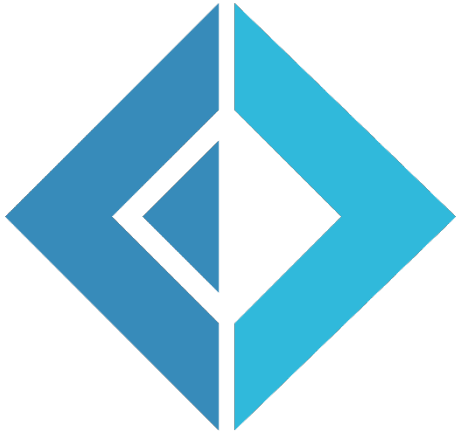
Interfaces

```
type Vehicle(model : string, color : string) =
   do
      printfn "Creating Vehicle"
   member val Model = model with get, set
   member val Color = color with get, set
   interface IDriveable with
      member __.Start() = printfn "Vrooom"
```

```
type IDriveable =
   abstract member Start : unit -> unit
```

F# for the Win!

C# interop



```
var location = Examples.Location.NewAnalog("FM", 98.4);
if (location.IsAnalog)
{
    Console.WriteLine("{0}",
        (location as Examples.Location.Analog).band);
}
```

Type Providers

```
#r "FSharp.Data.dll"
open FSharp.Data

let apiUrl   = "http://api.openweathermap.org/data/2.5/weather?q="
type Weather = JsonProvider<"http://api.openweathermap.org/data/2.5/weather?q=London">

let sf = Weather.Load(apiUrl + "San Francisco")
sf.Sys.Country
sf.Wind.Speed
sf.Main.
```

- Humidity
- JsonValue
- Pressure
- Temp — property JsonProvider<...>.Main.Temp: decimal
- TempMax
- TempMin

F# Functional Data Structures

www.pluralsight.com/courses/fsharp-functional-data-structures

Apps

Pluralsight Digital-Tutors Code School

PLURALSIGHT

Browse courses Blog Teach Search

History Popular New releases Christian

F# Functional Data Structures

This course describes the important data structures - especially collections - available in F#, together with the functions which F# provides for working with them.

by Kit Eason

Table of contents

Expand all 53%

Introduction		13:15
Arrays		41:49
Sequences	✔	49:35
Lists		12:07
Lists, Pattern Matching, and Recursion	✔	16:52

Course content

- Table of contents
- Description
- Transcript
- Exercise files
- Assessment
- Discussion

A Functional Architecture With F#

www.pluralsight.com/courses/functional-architecture-fsharp

Apps

Pluralsight Digital-Tutors Code School

PLURALSIGHT

Browse courses Blog Teach Search

History Popular New releases Christian

A Functional Architecture with F#

Learn how to build mainstream applications with F#.

by Mark Seemann

Table of contents Expand all 44%

Thinking Functionally	✔	27:02
Pipes and Filters		39:03
Map/Reduce		36:16
Cross-Cutting Concerns		45:50

Course content

- Table of contents
- Description
- Transcript
- Exercise files
- Assessment
- Discussion

fsharpforfunandprofit.com

fsharpforfunandprofit.com

Apps

F# for fun and profit | Home | Why use F#? | Help with F# | Site Contents | Search

Are you an experienced C#, Java or Python developer?

Do you want to understand what all the fuss about functional programming is about?

This site will introduce you to F# and show you ways that F# can help in day-to-day development of mainstream commercial business software. On the way, I hope to open your mind to the joys of functional programming – it really is fun!

If you have never heard of F#, it is a general purpose functional/hybrid programming language which is great for tackling almost any kind of software challenge. F# is free and open source, and runs on Linux, Mac, Windows and more. Find out more at the F# Foundation.

Learn to think functionally

"Thinking functionally" is critical to getting the most out of F#, so I will spend a lot of time on getting the basics down, and I will generally avoid too much discussion of the hybrid and OO features.

Useful examples

The site will mostly focus on mainstream business problems, such as domain driven design, website development, data processing, business rules, and so on. In the examples I will try to use business concepts such as Customer, Product, and Order, rather than overly academic ones.

Don't be scared

F# can look very intimidating if you look at complex code without any background. In the beginning I will keep it very simple, and I have tried to anticipate the questions that a newcomer to functional programming concepts will have. If you work through the examples slowly (and in the right order) you should have no problem understanding everything.

Have fun!

Many people claim that learning to think functionally will "blow your mind". Well, it's true! Learning a completely new paradigm is exciting and stimulating. You may fall in love with programming again.

Getting started

If you are completely new to F#, find out more about F# and how it is used at the F# Foundation. To download

SHARE THE LOVE

F# |>I ♥

SLIDES/VIDEOS

- Functional Design Patterns
- A functional approach to Domain Driven Design
- A functional approach to error handling (Railway Oriented Programming)

SERIES

- Why use F#?
- Thinking functionally
- Expressions and syntax
- Understanding F# types
- Object-oriented programming in F#
- Porting from C#
- Designing with types
- Computation Expressions
- A recipe for a functional app
- Dependency cycles
- Understanding monoids
- Low-risk ways to use F# at work

fsharp.org

fsharp.org

Apps

HOME LEARN USE GUIDES CONTRIBUTE TRAINING FOUNDATION

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.

USE F#

F# runs on Linux, Mac OS X, Android, iOS, Windows, GPUs, and browsers. It is free to use and is open source under an OSI-approved license.

F# is used in a wide range of application areas and is supported by both an active open community and industry-leading companies providing professional tools.

The mission of the F# Software Foundation is to promote and advance the F# programming language, including a diverse and international community of F# programmers.

Getting Started

Learn

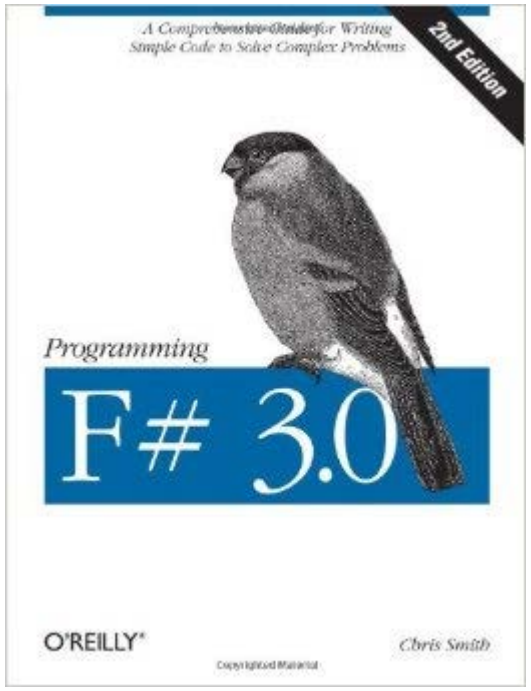
- Learn F# online with Try F#
- Read F# books and tutorials
- Read the documentation
- Watch F# videos

Community

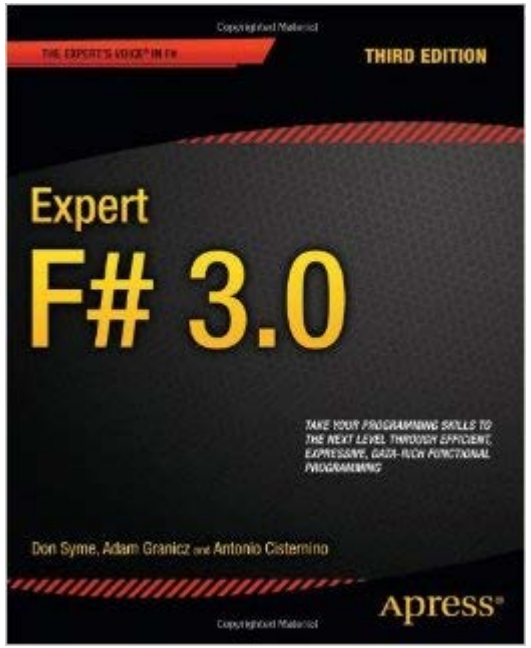
- Ask questions on StackOverflow
- Attend F# events and join F# meetups
- Contribute to F# projects
- Gain skills with professional training and consultants

Books

Programming F# - Chris Smith



Expert F# - Don Syme et al



github.com/fsharp

GitHub, Inc. [US] https://github.com/fsharp/

Apps

Search GitHub

Pull requests Issues Gist

F# Compiler and Components (Open Edition)

The World http://fsharp.org fsharp@fsharp.org

Filters Find a repository...

FAKE F# ★ 420 ⑂ 251

FAKE - F# Make

Updated 17 hours ago

FSharp.Data HTML ★ 228 ⑂ 132

F# Data: Library for Data Access

Updated a day ago

FSharp.Compiler.Service F# ★ 86 ⑂ 50

F# compiler component exposing additional functionality for editing clients

Updated a day ago

People 9 >

Just Dive In!