# F# for Financial Computing

*Tomas Petricek*

*http://tomasp.net*

## Agenda

***Why F# matters***

*Accessing data with type providers*

*Functional & explorative programming*

*Scientific computing with F#*

## Analytical Components

| | |
|---|---|
| **Finance** | • Valuation Engines<br>• Risk Analysis |
| **Trading** | • Trading Platforms<br>• Algorithmic Trading |
| **Web** | • Ranking<br>• Face Recognition |
| **Retail** | • Recommender Systems<br>• Fraud detection |

## Developing Analytical Components

Time to Market

Efficiency

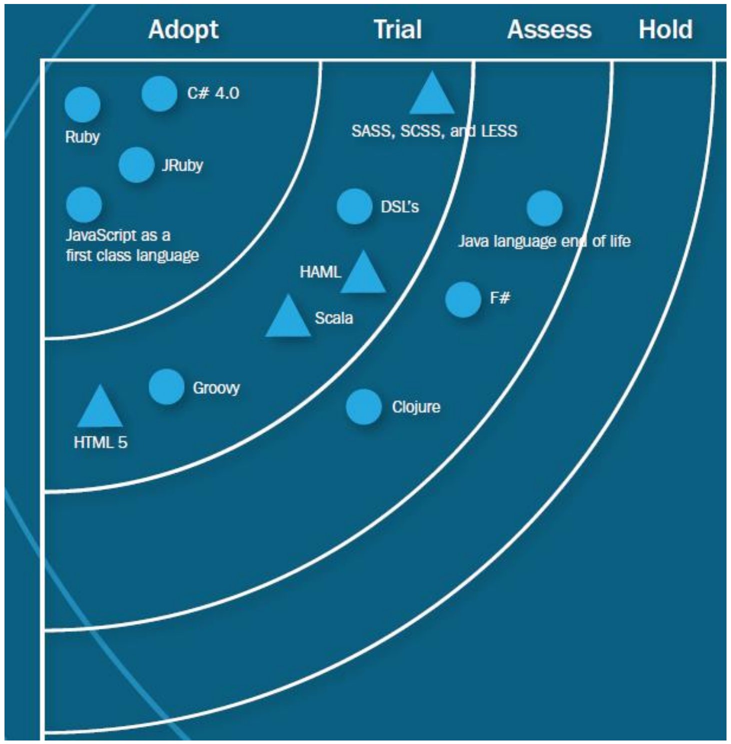
Correctness

Complexity

## ThoughtWorks Technology Radar

Developers trying to achieve explicit business logic within an application may opt to express their domain in F# with the majority of plumbing code in C#.

*ThoughtWorks Technology Radar (March 2012)*



## Agenda

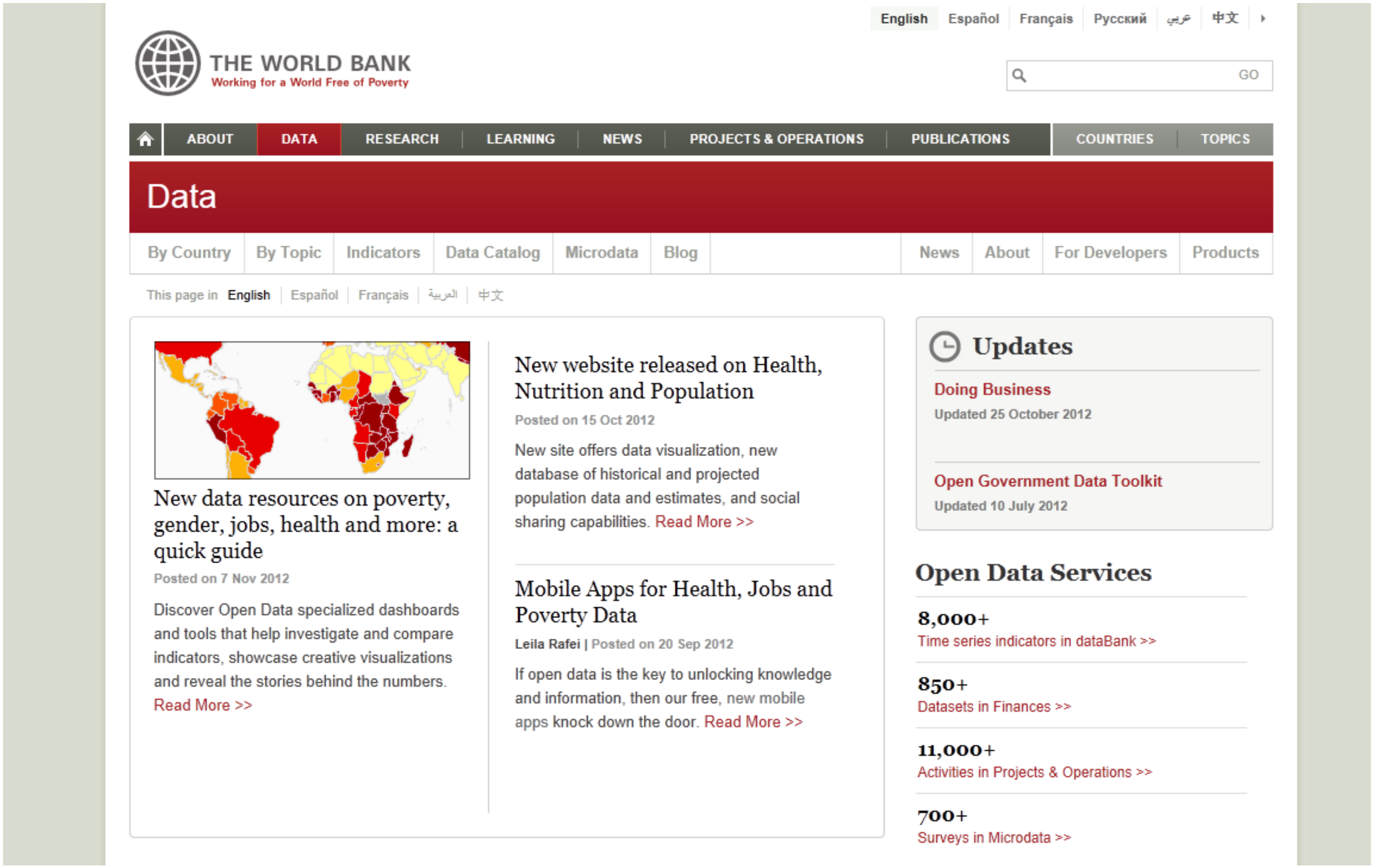
*Why F# matters*

***Accessing data with type providers***

*Functional & explorative programming*

*Scientific computing with F#*

# Accessing data today



English | Español | Français | Русский | عربي | 中文

THE WORLD BANK
Working for a World Free of Poverty

GO

ABOUT | DATA | RESEARCH | LEARNING | NEWS | PROJECTS & OPERATIONS | PUBLICATIONS | COUNTRIES | TOPICS

Data

By Country | By Topic | Indicators | Data Catalog | Microdata | Blog | News | About | For Developers | Products

This page in English | Español | Français | العربية | 中文

New data resources on poverty, gender, jobs, health and more: a quick guide

Posted on 7 Nov 2012

Discover Open Data specialized dashboards and tools that help investigate and compare indicators, showcase creative visualizations and reveal the stories behind the numbers. Read More >>

New website released on Health, Nutrition and Population

Posted on 15 Oct 2012

New site offers data visualization, new database of historical and projected population data and estimates, and social sharing capabilities. Read More >>

Mobile Apps for Health, Jobs and Poverty Data

Leila Rafei | Posted on 20 Sep 2012

If open data is the key to unlocking knowledge and information, then our free, new mobile apps knock down the door. Read More >>

Updates

Doing Business
Updated 25 October 2012

Open Government Data Toolkit
Updated 10 July 2012

Open Data Services

8,000+
Time series indicators in dataBank >>

850+
Datasets in Finances >>

11,000+
Activities in Projects & Operations >>

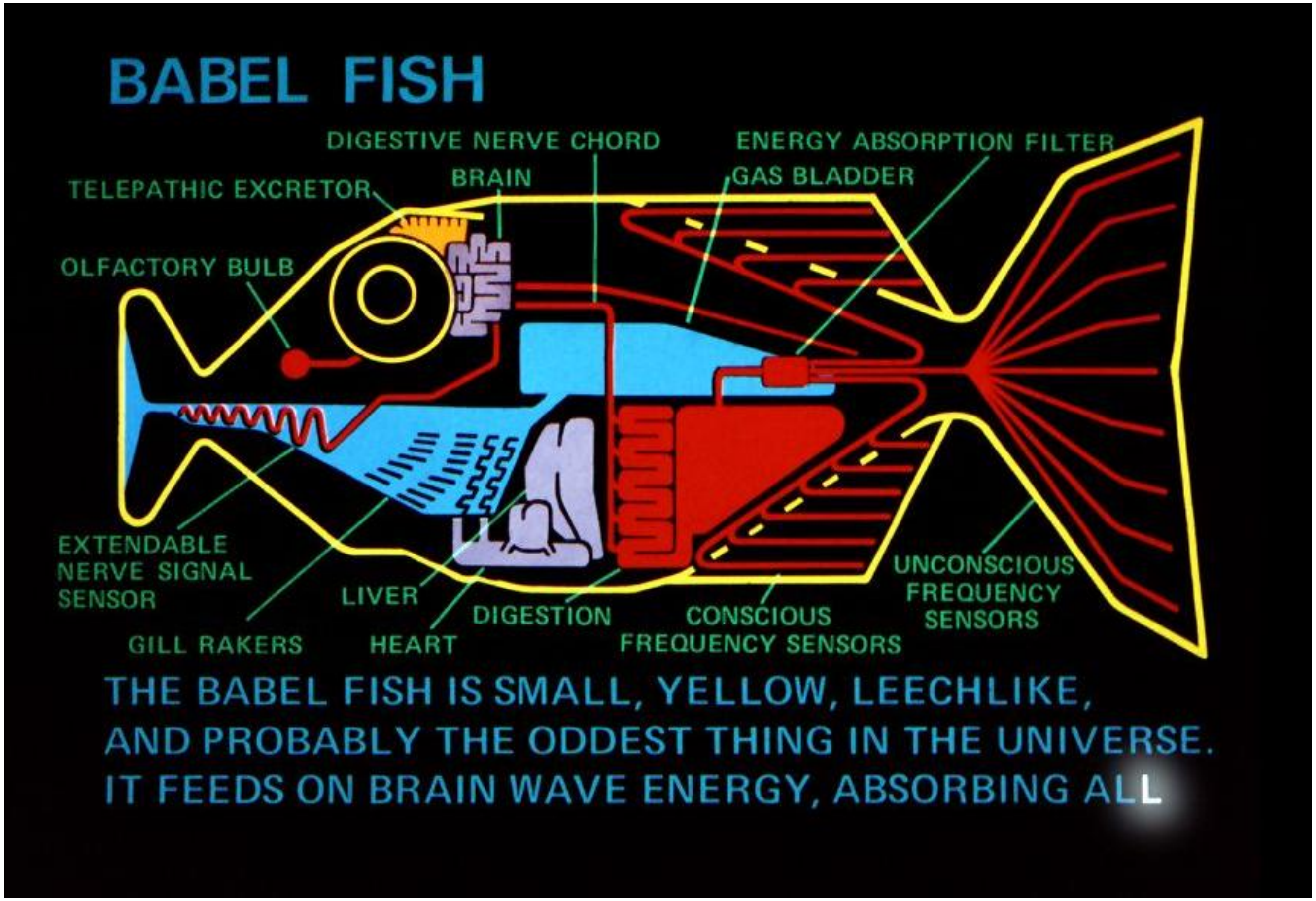
700+
Surveys in Microdata >>

Microsoft Research

## Languages and Data

- Mismatch between two worlds!
- Structure in the **language**
  - Classes with properties
  - Functional data types
- Structure in the **data source**
  - Database, XML, OData, Web services
  - CSV file, REST service or JSON file

# F# Type Providers


BABEL FISH
DIGESTIVE NERVE CHORD
ENERGY ABSORPTION FILTER
TELEPATHIC EXCRETOR
BRAIN
GAS BLADDER
OLFACTORY BULB
EXTENDABLE NERVE SIGNAL SENSOR
LIVER
DIGESTION
CONSCIOUS FREQUENCY SENSORS
UNCONSCIOUS FREQUENCY SENSORS
GILL RAKERS
HEART
THE BABEL FISH IS SMALL, YELLOW, LEECHLIKE,
AND PROBABLY THE ODDEST THING IN THE UNIVERSE.
IT FEEDS ON BRAIN WAVE ENERGY, ABSORBING ALL

# Processing CSV files in F#

## Stock prices from Yahoo Finance

Column names

```
Open,  High,  Low,   Close, Volume,   Adj
31.25, 31.73, 30.55, 30.77, 14122000, 30.77
31.92, 31.99, 30.76, 31.10, 19526900, 31.10
31.96, 32.19, 30.90, 31.36, 17713300, 31.36
(...)
```

Data set

## What we want to write

Column names

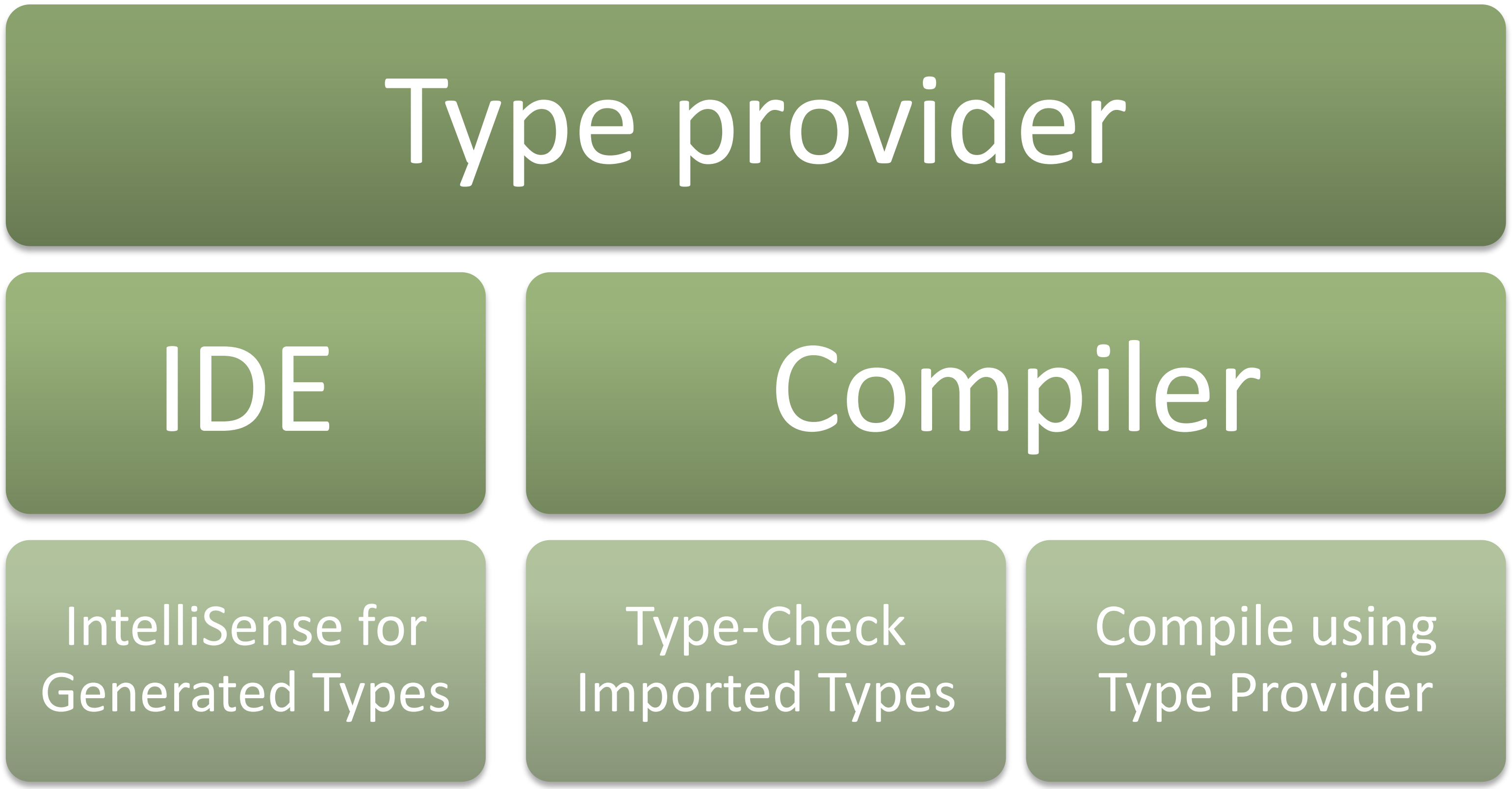
```
let fbstocks = (...)
for row in fbstocks.Data do
  printfn "%f" (row.Close - row.Open)
```

Microsoft Research

Analyzing WorldBank data

## DEMO #1

# What type providers do?


Type provider
IDE
Compiler
IntelliSense for Generated Types
Type-Check Imported Types
Compile using Type Provider

## Agenda

*Why F# matters*

*Accessing data with type providers*

***Functional & explorative programming***

*Scientific computing with F#*

## Functional programming

$$x = x + 1$$

Microsoft Research

# Functional programming

*Functional programming emphasizes the **evaluation of expressions**, rather than **execution of commands**. The expressions in these languages are formed by using functions to combine basic values.*

- This is how mathematics works!
  - Roots of a quadratic equation

$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

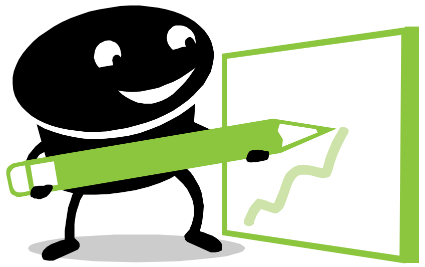
# Functional programming

- Program as an equation
  - Equations can get long and ugly
  - How to break them into pieces?

- More ideas from mathematics

$$Let\ discriminant\ \boldsymbol{D}\ be: b^2 - 4ac$$

$$Roots\ of\ quadratic\ equation: \frac{-b \pm \sqrt{D}}{2a}$$



# The `let` keyword

## Immutable values

```
> let a = 3.0
val a : float = 3.0

> a = 2.0
val it : bool = false
```

Type *float* is inferred (efficiency & correctness)

Normal value cannot be mutated

## Function declarations

Add function parameters

```
> let twice a = a * 2.0
val twice : float -> float
```

Still no need to write the type!

Microsoft Research

## Expressions at a larger scale

- Calculating with trades instead of numbers

```
let itcontract =
   sellOn
     (DateTime(2012, 4, 30)) ("MSFT", 23.0) $
   purchaseRepeatedly
     (DateTime(2012, 4, 23))
     (TimeSpan.FromDays(7.0))
     10 ("AAPL", 220.0)
```

- Declarative specification, multiple uses
  – What can happen on a given date?
  – Valuation and risk assessment

Visualizing stock options

## DEMO #2

## Agenda

*Why F# matters*

*Accessing data with type providers*

*Functional & explorative programming*

***Scientific computing with F#***

## Why write calculations in F#?

- Efficient
  - Interoperability & good libraries
  - Compiled to native code by JIT
- Generic code
  - Better than C#, simplified by type inference
- Correctness
  - Types help avoid mistakes
  - Advanced checking with units of measure

# F# Technologies

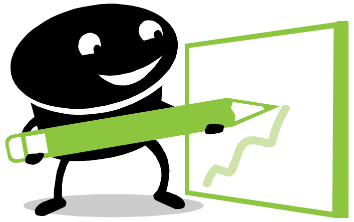
- Standard F# tools
  - F# language and core libraries

- Math libraries
  - Math.NET Numerics (open source)
  - StatFactory FCore, QuantAlea GPGPU

- Machine-learning packages
  - Microsoft Solver Foundation, Infer .NET

## Working with stock prices

- Obtaining stock prices from Yahoo
  - Using CSV type provider
- Using units of measure
  - Avoiding common mistakes
- Working with collections in F#
  - Functions and sequence expressions



Statistics reminder
(standard deviation)

$$\sqrt{\frac{\sum(v_i - avg)^2}{count}}$$

Analyzing Facebook stock price

## DEMO #3

# How functional programmers think

- Data-centric design
  - What data does the program use?
  - How to represent the data?
  - How to transform between representations?
- Implementing design
  - Domain on a single page
  - Transformations are functions

## Data Structures

- Define data structures first
  - Model data by composing primitives
- Three basic functional types

| Tuples and Records | Discriminated Unions | Collections |
|---|---|---|
| Combine values of different types | Represent one of alternative options | Zero or more values of the same type |

Parallelizing Black-Scholes pricing

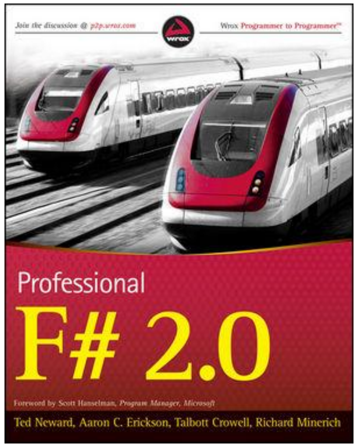
## DEMO #4

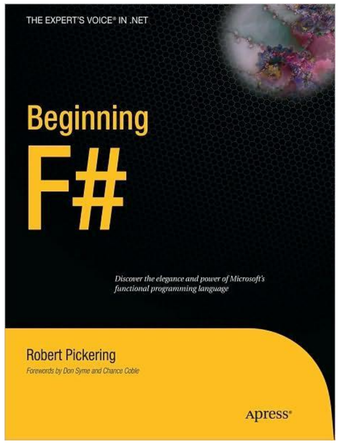
# Parallel computing etc.

- Parallel programming
  - Run faster on multi-core
- Asynchronous programming
  - Avoid blocking I/O
- Concurrent programming
  - Computations that communicate

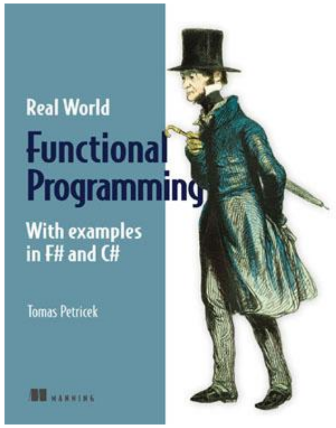
## SUMMARY

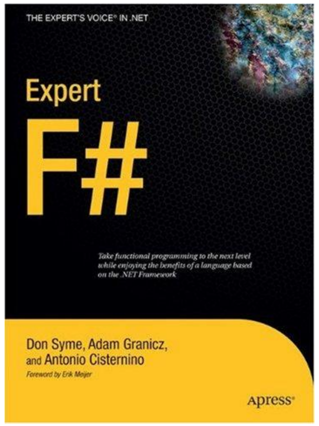
# F# Books

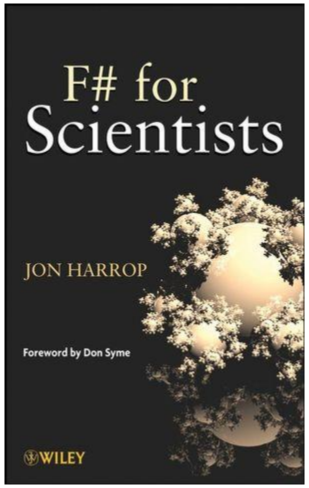

Join the discussion @ p2p.wrox.com
Wrox Programmer to Programmer™
Professional
F# 2.0
Foreword by Scott Hanselman, Program Manager, Microsoft
Ted Neward, Aaron C. Erickson, Talbott Crowell, Richard Minerich


THE EXPERT'S VOICE® IN .NET
Beginning
F#
Discover the elegance and power of Microsoft's functional programming language
Robert Pickering
Forewords by Don Syme and Chance Coble
Apress®

A Comprehensive Guide for Writing Simple Code to Solve Complex Problems
Programming
F#
O'REILLY®
Chris Smith
Foreword by Don Syme


Real World
Functional
Programming
With examples in F# and C#
Tomas Petricek
MANNING

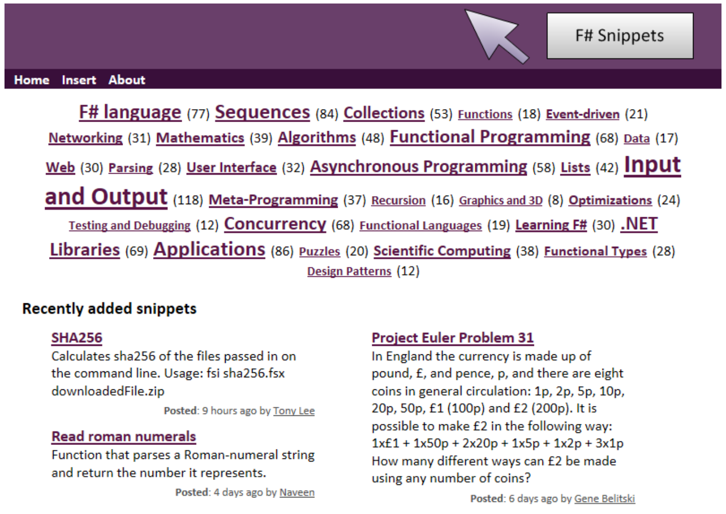

THE EXPERT'S VOICE® IN .NET
Expert
F#
Take functional programming to the next level while enjoying the benefits of a language based on the .NET Framework
Don Syme, Adam Granicz, and Antonio Cisternino
Foreword by Erik Meijer
Apress®


F# for
Scientists
JON HARROP
Foreword by Don Syme
WILEY

Visual F# 2010 for
Technical Computing
Jon D. Harrop

## F# on the internet

- F# Foundation
  - http://fsharp.org
- F# Snippets
  - http://fssnip.net
- MSDN Articles
  - http://functional-programming.net/msdn
- StackOverflow
  - http://stackoverflow.com/



F# language (77) Sequences (84) Collections (53) Functions (18) Event-driven (21) Networking (31) Mathematics (39) Algorithms (48) Functional Programming (68) Data (17) Web (30) Parsing (28) User Interface (32) Asynchronous Programming (58) Lists (42) Input and Output (118) Meta-Programming (37) Recursion (16) Graphics and 3D (8) Optimizations (24) Testing and Debugging (12) Concurrency (68) Functional Languages (19) Learning F# (30) .NET Libraries (69) Applications (86) Puzzles (20) Scientific Computing (38) Functional Types (28) Design Patterns (12)

**Recently added snippets**

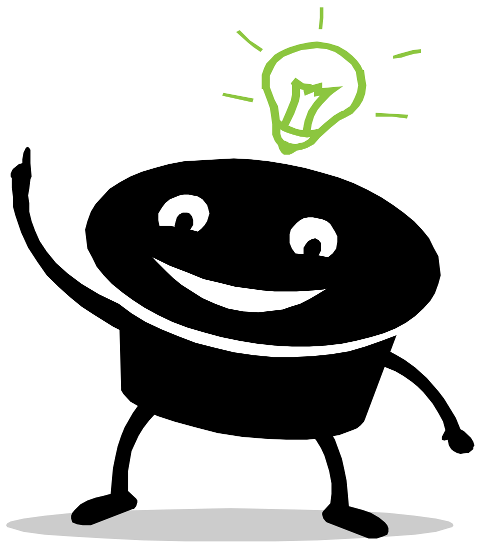
**SHA256**
Calculates sha256 of the files passed in on the command line. Usage: fsi sha256.fsx downloadedFile.zip
**Posted**: 9 hours ago by Tony Lee

**Read roman numerals**
Function that parses a Roman-numeral string and return the number it represents.
**Posted**: 4 days ago by Naveen

**Project Euler Problem 31**
In England the currency is made up of pound, £, and pence, p, and there are eight coins in general circulation: 1p, 2p, 5p, 10p, 20p, 50p, £1 (100p) and £2 (200p). It is possible to make £2 in the following way: 1x£1 + 1x50p + 2x20p + 1x5p + 1x2p + 3x1p How many different ways can £2 be made using any number of coins?
**Posted**: 6 days ago by Gene Belitski

## Thanks!

Questions?



Get in touch!

| | |
|---|---|
| Email: | tomas@tomasp.net |
| Web site: | http://tomasp.net |
| Twitter: | @tomaspetricek |