

# TryF# – building a system for multi-platform access to a managed language



Nigel Horspool  
University of Victoria, Canada  
[nigelh@cs.uvic.ca](mailto:nigelh@cs.uvic.ca)

- Designed by Don Syme,  
MSR Cambridge
- Mostly functional language,  
in spirit of Objective Caml

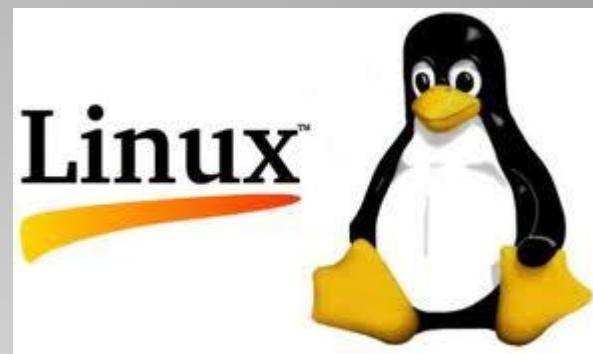
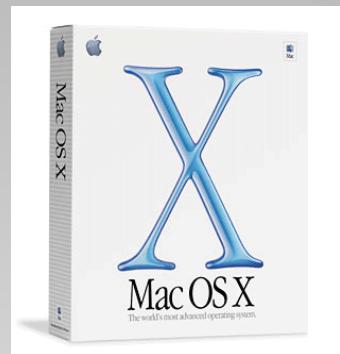


- Has strong inferred static typing
- A full member of the .NET family
- Integrated with Visual Studio 2010

## What is F#?



- as .NET plus a standalone F# package
  - or
- with **Visual Studio 2010**



- as **mono** plus a standalone F# package
  - or
- as **monodevelop** plus the F# plugin

## Options for Installing F#

<http://www.trypython.org>

**Python** in a browser

<http://tryruby.org/>

**Ruby** in a browser

<http://tryhaskell.org/>

**Haskell** in a browser

<http://tryscheme.sourceforge.net/>

**Scheme** in a browser

**Help for Beginners in Various  
Programming Languages**

**On Server**



**On Client**

**Where is the Code Evaluated?**



[www.tryfsharp.org](http://www.tryfsharp.org)

The **TryF#** website is intended to give

- a good feel for the language
- a brief introduction to F# via on-line tutorial
- the ability to edit and run modest programs
- provide links to further resources

**And now ... a Try F# Website**

We want ...

- same experience for everyone, regardless of platform and browser
- scalability to many users
- security
- no major investment in servers

Implies that TryF# must work as a browser plugin with all or most computing on the client side

## TryF# Design Considerations

# Silverlight

- is “an application framework for rich internet applications” (Wikipedia)
- usable as a browser plug-in
- includes a large subset of .NET Framework – actions are programmed in any .NET language
- UIs scripted in XAML
- runs on Windows and Mac OS X
- **Moonlight** is the Linux version



## Use of Silverlight & Moonlight

1. Determines the OS and browser
2. Checks if Silverlight/Moonlight already installed; if not
  - Windows: provides necessary components
  - Mac OS X: provides necessary components
  - Linux: tells user what needs to be installed
3. Downloads the TryF# plugin

## Initial Actions of TryF#

- *Begin* button initiates checking and the downloads
  - Three different webpage layouts
  - Tutorial includes sample F# code
  - Easy to run the sample code
- Top half of code window contains editable input
    - Right-click options
    - Bottom half shows output from `fsi` (F# Interactive)
    - Optional extra graphics window

## Demo Time for TryF# Features

- Easy way for students to run F# code on their own computers
- Windows, Mac and Linux\* users are given identical environments
- Includes simple tutorial introduction to F# aimed at beginners
- F# code can be loaded from local files
- Contents of code window can be saved as local file for submission to the instructor

\* coming soon

## Use of TryF# in Courses

- The TryF# website can be hosted on a local webserver
- Tutorial pages can be edited or replaced with new content
  - Format is HTML with extra tag attributes to support code examples
  - Tutorial sections can be written in *Markdown* format and translated to HTML (with some minimal touch-ups later)
- The XAML and CSS specifications can be changed to give a different look-and-feel

## Local Modifications?

O.S.	IE8 IE9	Firefox 3+	Safari 3+	Chrome 4+
Vista, Win7 + Silverlight 4.0	YES	YES	NO	YES
Mac OS X + Silverlight 4.0	n/a	YES	YES	YES
Linux + Mono 2.8 + Moonlight 3.0	n/a	YES	n/a	n/a

## Compatibility of TryF# Plugin

- A more complete language tutorial
- Access to more class libraries than those included in Silverlight
- 'Intellisense' when entering code
- Inclusion of quizzes and tests (as uploaded for local use by instructors?)
  - With automatic scoring? Mark submission?
- Uploading of additional code examples from users

## Future Directions for TryF#?

tryF#

<http://www.tryfsharp.org>

**That Web Address Again**