Some F# Practicalities

Björn Lisper
School of Innovation, Design, and Engineering
Mälardalen University

bjorn.lisper@mdh.se
http://www.idt.mdh.se/~blr/

How to Develop F# Programs

F# programs are just text files

You can create and edit them with the text editor of your choice

F# files should end with “`.fs`”

You can either

Batch compile into a “`.exe`” file with `fsc`, and run:

```
>fsc file.fs   ("fsharpc file.fs" in F# 3.1)
>file.exe
```

Or, use the the F# interactive compiler, `fsi` (`fsharpi` in F# 3.1)

The F# Interactive Compiler

```
>fsi

Microsoft (R) F# 2.0 Interactive build 2.0.0.0
Copyright (c) Microsoft Corporation. All Rights Reserved.

For help type #help;;

>
```

Gives you an environment where you can type F# expressions to the prompt, and have them evaluated. End every expression with “`;;`”

```
> 5 + 6 ;;
val it : int = 11
>
```

So `fsi` can be used as a simple calculator

The F# Interactive Compiler (2)

Any F# expression can be evaluated:

```
> let x = 17.0 in x*(3.0 + 7.0/x);;
val it : float = 58.0
```

You can also make declarations with `let`. These are visible from then on:

```
> let x = 17.0;;
val x : float = 17.0
> x + 33.5;;
val it : float = 50.5
```

The F# Interactive Compiler (3)

You will want to use `fsi` for interactive testing. To get your code into `fsi`, use the `#load` command:

```
#load "file.fs";;
```

This will compile the code in `file.fs` and load it into `fsi`

`fsi` will create a *module* named `File`, where the declared entities in `file.fs` reside (more on modules later)

The F# Interactive Compiler (4)

A function `f`, declared in `file.fs`, can be accessed by prefixing its name with the module name:

```
> File.f 2;;
val it : int = 47
```

To avoid the prefix, you can first *open* the module:

```
> open File;;
> f 2;;
val it : int = 47
```

Visual Studio

Windows users can use Visual Studio 2008/10/12

From Visual Studio 2010 there is full support for F#