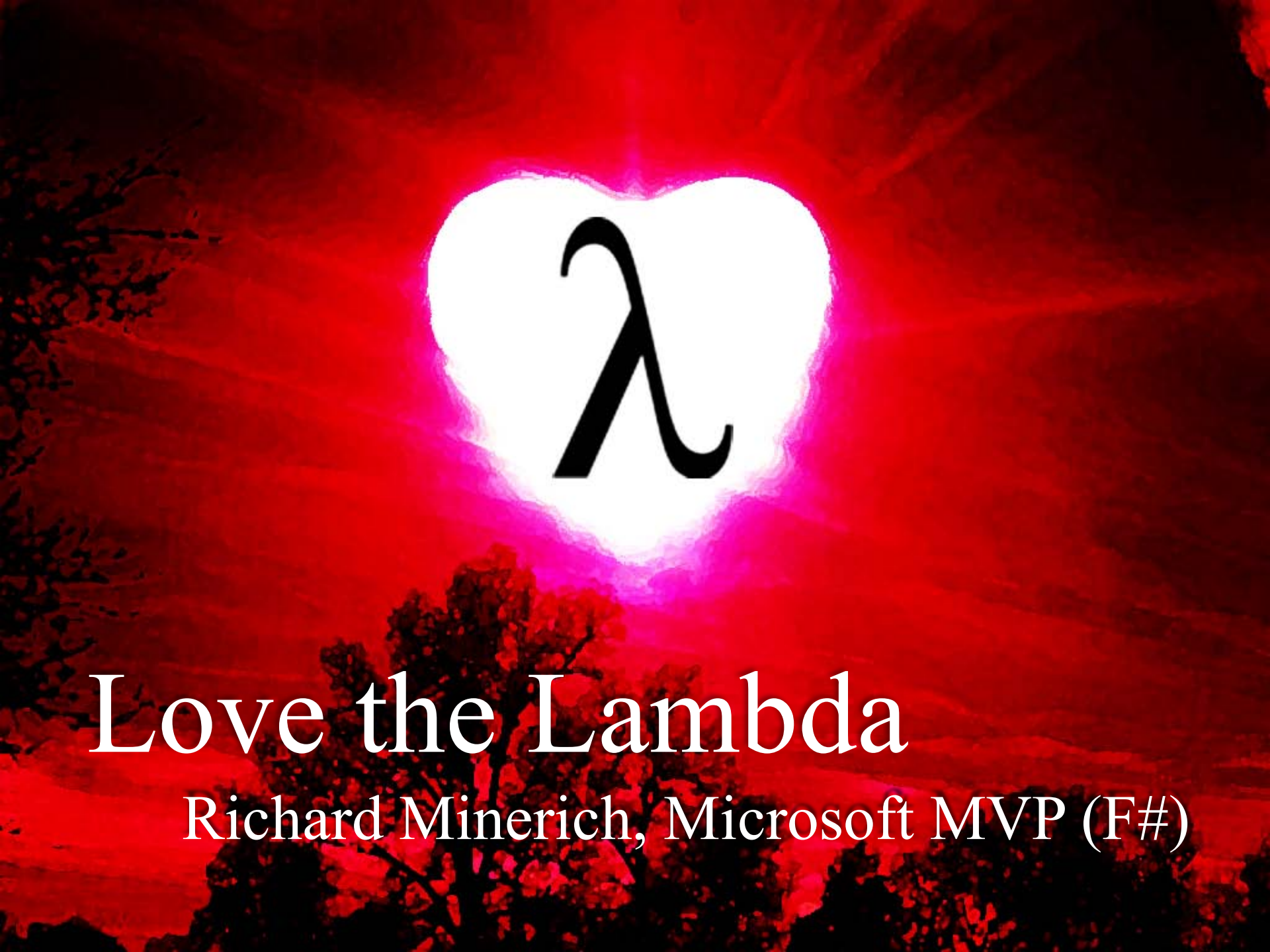

Love the Lambda

Richard Minerich, Microsoft MVP (F#)

The tools we use have a profound (and devious!) influence on our thinking habits, and, therefore, on our thinking abilities.

-- Edsger Dijkstra

C#

```
int one = 1;
int two = 2;
Func<int, int, int> adder =
    (num1, num2) =>
    {
        return num1 + num2;
    };
int three = adder(one, two);
```

C#

```
var one = 1;
var two = 2;
Func<int, int, int> adder =
    (num1, num2) => num1 + num2;
var three = adder(one, two);
```

F#

```
let one = 1
let two = 2
let adder num1 num2 =
    num1 + num2
let three = adder one two
```

Functions as Variables

We can pass them.
We can return them.
We can assign them.
We can replace them.

C#

```
var one = 1;
Func<int, int> addOne =
    (num) => num + one;
var two = 2;
var three = addOne(two);
```

F#

```
let one = 1
let addOne num =
    num + one
let two = 2
let three = addOne two
```

Examples of Closures

What else can we do?

Maybe make language features?

Controlling complexity is the essence of computer programming.

-- Brian Kernigan

Higher Order Functions

Filter – Find a subset of a collection

Map – 1:1 mapping from old to new

Reduce – Many to one of the same type

Fold – Many to one of any type

Taking it further in F#

- Partial Application
- Pipelining
- Composition
- Inlining

New England F# User's Group

Next Meeting: July 5th

Daniel Mohl - F# for High Performance, Readable, and Efficient Code.

Check us out at fsug.org

Thank You for coming!

For more on F#, check out the MSDN site:

msdn.microsoft.com/en-us/fsharp/

Questions/Comments? Contact me

In person: (I'm here all day)

On twitter: twitter.com/rickasaurus

Or via my site: RichardMinerich.com