

0 からわかるF#

Part1

中 博俊

F# September 2008 CTP Base



わんくま同盟名古屋勉強会 #07

F#ってなに？

- Visual Studio 2010 (次のバージョン) でリリースされる予定
- 関数型言語
- Ocamlモチーフ

勝手に言語トレンド

- 手続きだけの言語 Cobol, Fortran
↓
- 構造化言語 Cobol78, C
↓
- オブジェクト指向言語 C++, Java, C#
↓
- ダイナミック言語 Python, Ruby
↓
- 関数型言語 Ocaml, Haskell, F#

関数型言語って何？

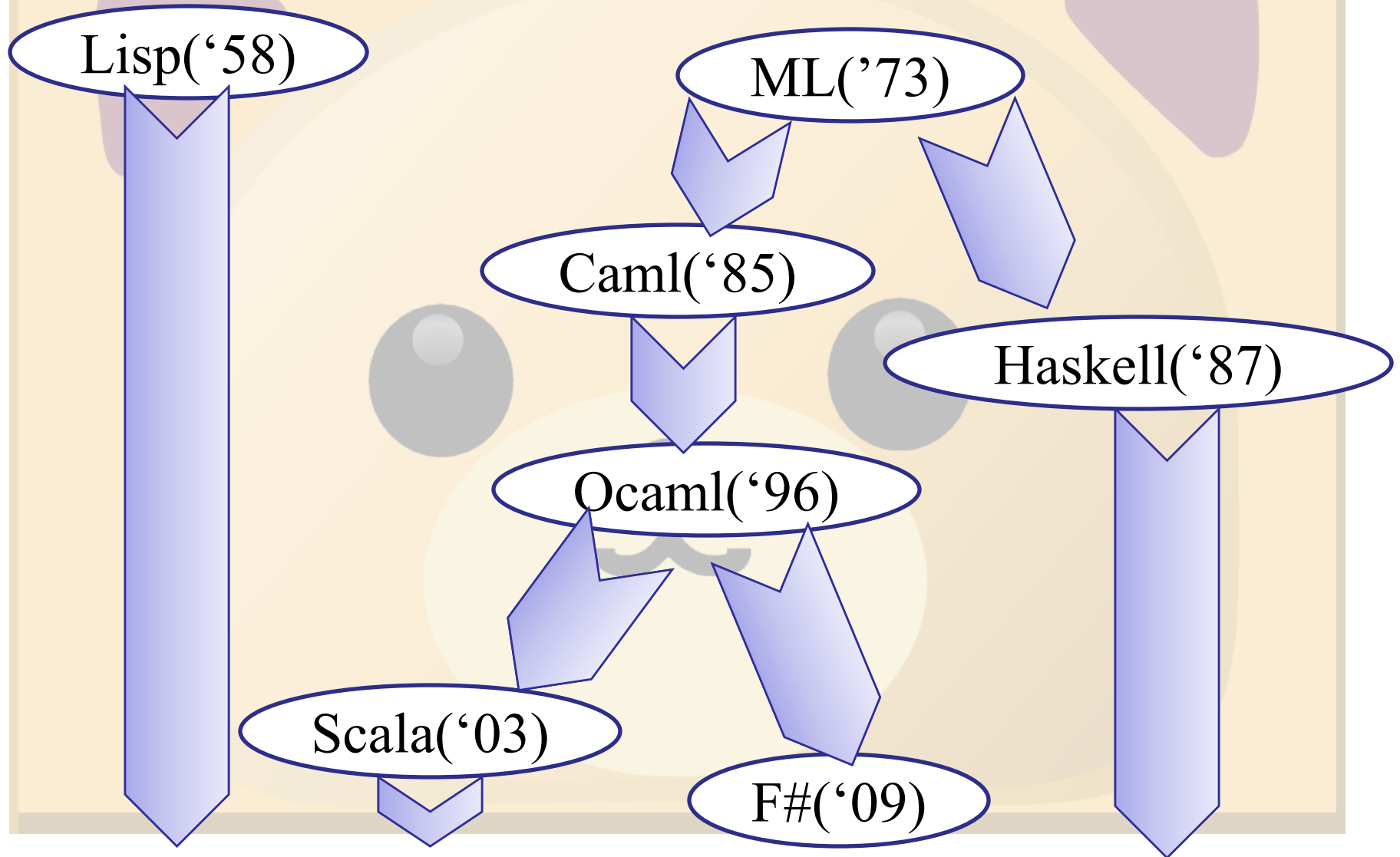
いろいろ複雑な定義がありますが...

- 状態を持たないから、副作用を持たない
 - 変数の書き換えが出来ないという意味
- 遅延評価を行う
- リスト構造が基本
- λ 式(ラムダ式)
- カリー化 ←カリー博士にちなんで命名



今は気にしなくていいです！

関数型言語の系譜

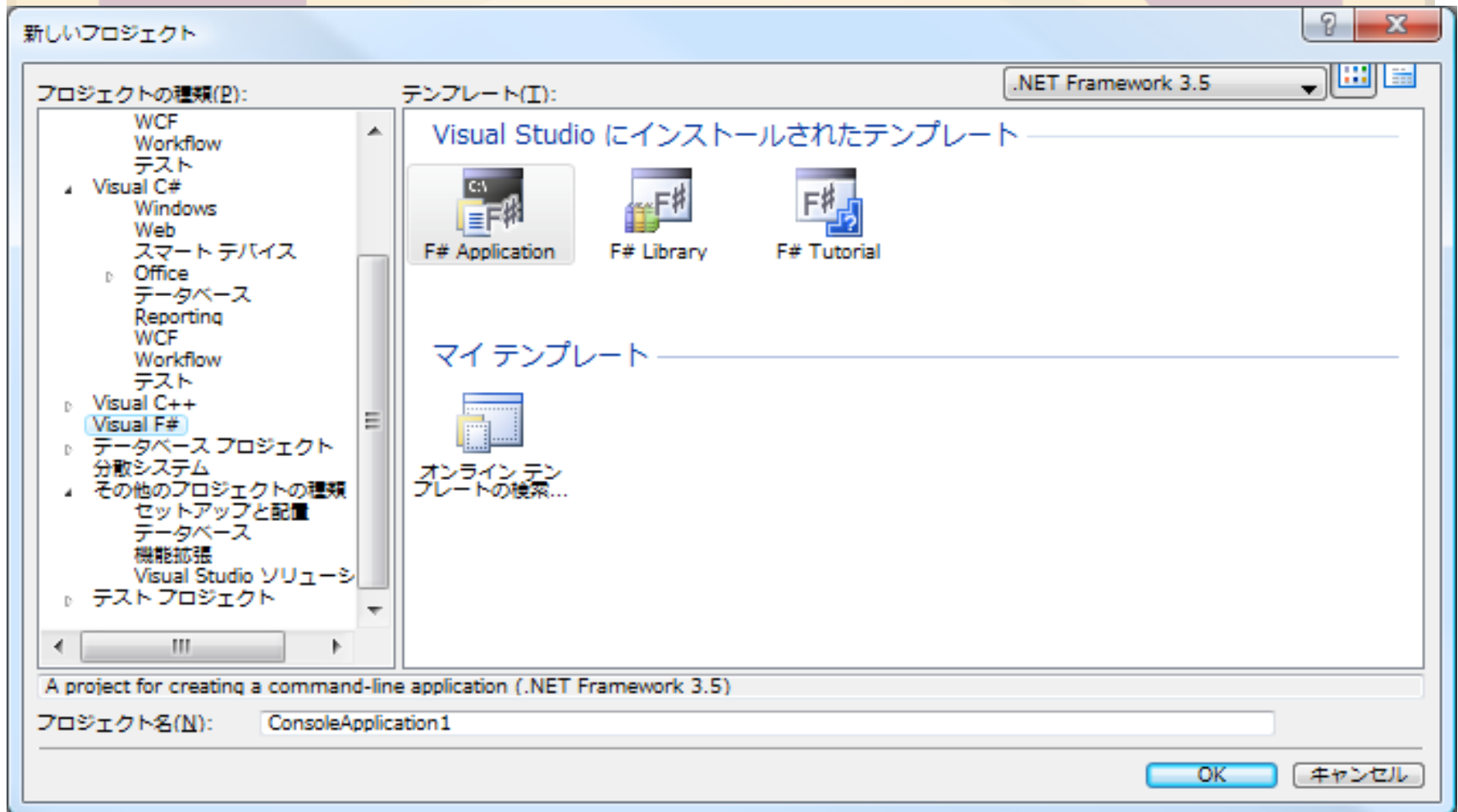


環境作り

- Visual Studio 2008
- Microsoft F#, September 2008 Community Technology Preview
- <http://www.microsoft.com/downloads/details.aspx?FamilyID=61ad6924-93ad-48dc-8c67-60f7e7803d3c&displaylang=en>

たったこれだけ

プロジェクトが追加されている



まとりあえずHello World.

DEMO #1

#light

```
printf "Hello F# World"  
printf "Hello F# World"
```



わんくま同盟名古屋勉強会 #07



DEMO #2

System.Windows.Forms.dllを参照
System.Windows.Forms.MessageBox.Show "Hello World"



わんくま同盟名古屋勉強会 #07

コンソールモードでやるよ！

- C:¥Users¥vaioz¥Desktop¥FSharp-1.9.6.2¥bin¥fsi.exe
- これでコンソールモードです。
- 日本語対応してないから本格的には使えない。。。。

1～10までの二乗を表示するプログラムC#1.0版

```
ArrayList numbers = new ArrayList();  
for (int i = 1; i <= 10; i++) {  
    numbers.Add(i); }  
}
```

```
ArrayList squares = new ArrayList();  
for (int i = 0; i < numbers.Count; i++) {  
    squares.Add((int)numbers[i] * (int)numbers[i]); }  
}
```

```
Console.WriteLine("N^2 = {");  
for (int i = 0; i < squares.Count; i++) {  
    Console.WriteLine(squares[i] + "; "); }  
Console.WriteLine("}");  
Console.ReadKey(true);  
}
```



1~10までの二乗を表示するプログラムC#2.0版

```
List<int> numbers = new List<int>();  
for ( int i = 1; i <= 10; i++ ) {  
    numbers.Add(i);}  
List<int> squares = numbers.ConvertAll(  
    delegate(int x ) { return x*x; });  
Console.Write("N^2 = {");  
squares.ForEach(delegate(int x )  
    { Console.Write(x + "; "); });  
Console.Write("}");  
  
Console.ReadKey(true);
```



1~10までの二乗を表示するプログラムC#3.0版

```
var numbers = Enumerable.Range(1, 10);  
var squares = (from x in numbers  
               let square = x*x  
               select square).ToList();  
Console.Write("N^2 = {");  
squares.ForEach(x => Console.Write(x + "  
"));  
Console.Write("}");  
  
Console.ReadKey(true);
```



1～10までの二乗を表示するプログラムF#版

```
#light
```

```
let numbers = [1 .. 10]
```

```
let square x = x * x
```

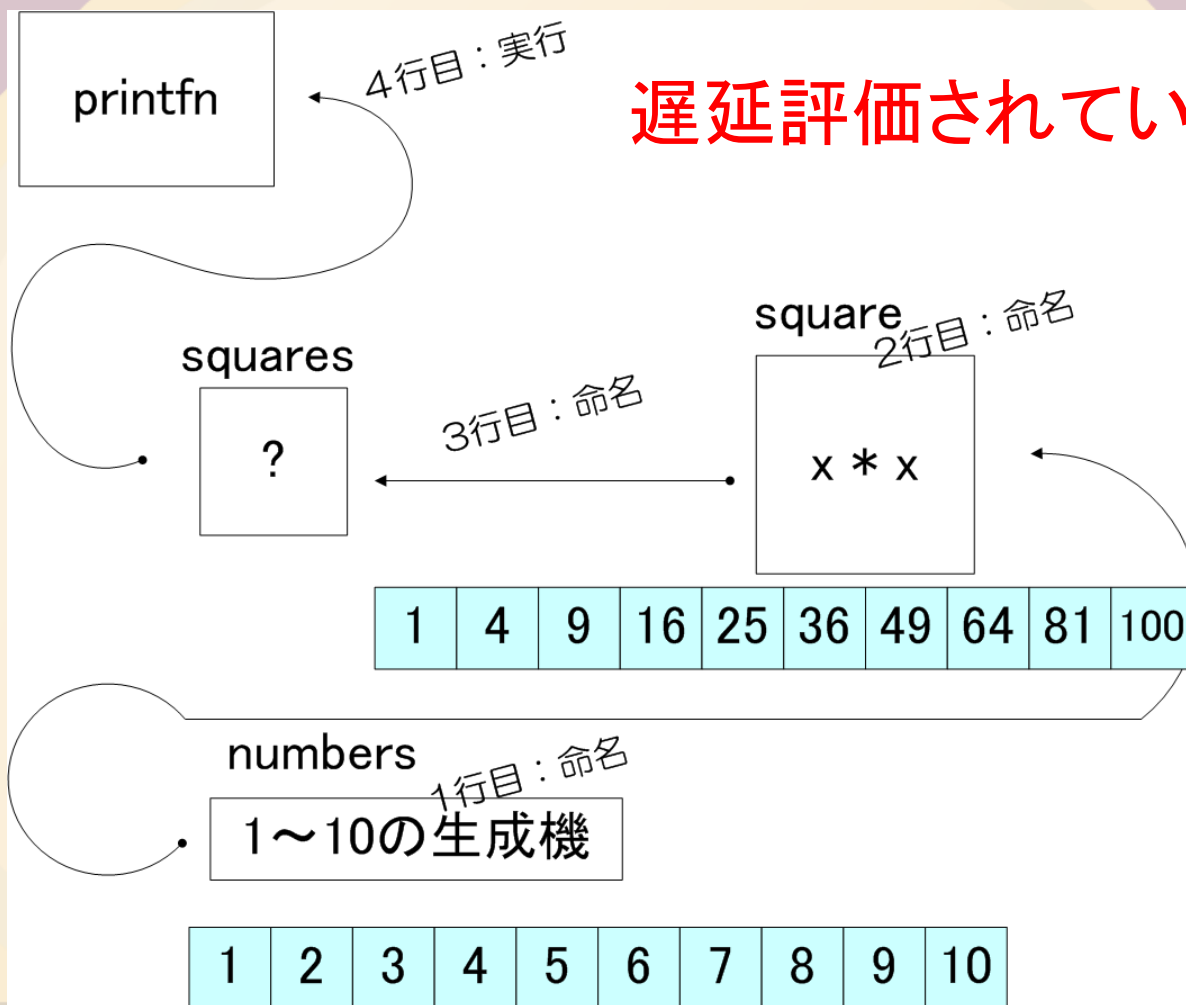
```
let squares = List.map square  
  numbers
```

```
printfn "N^2 = %A" squares
```

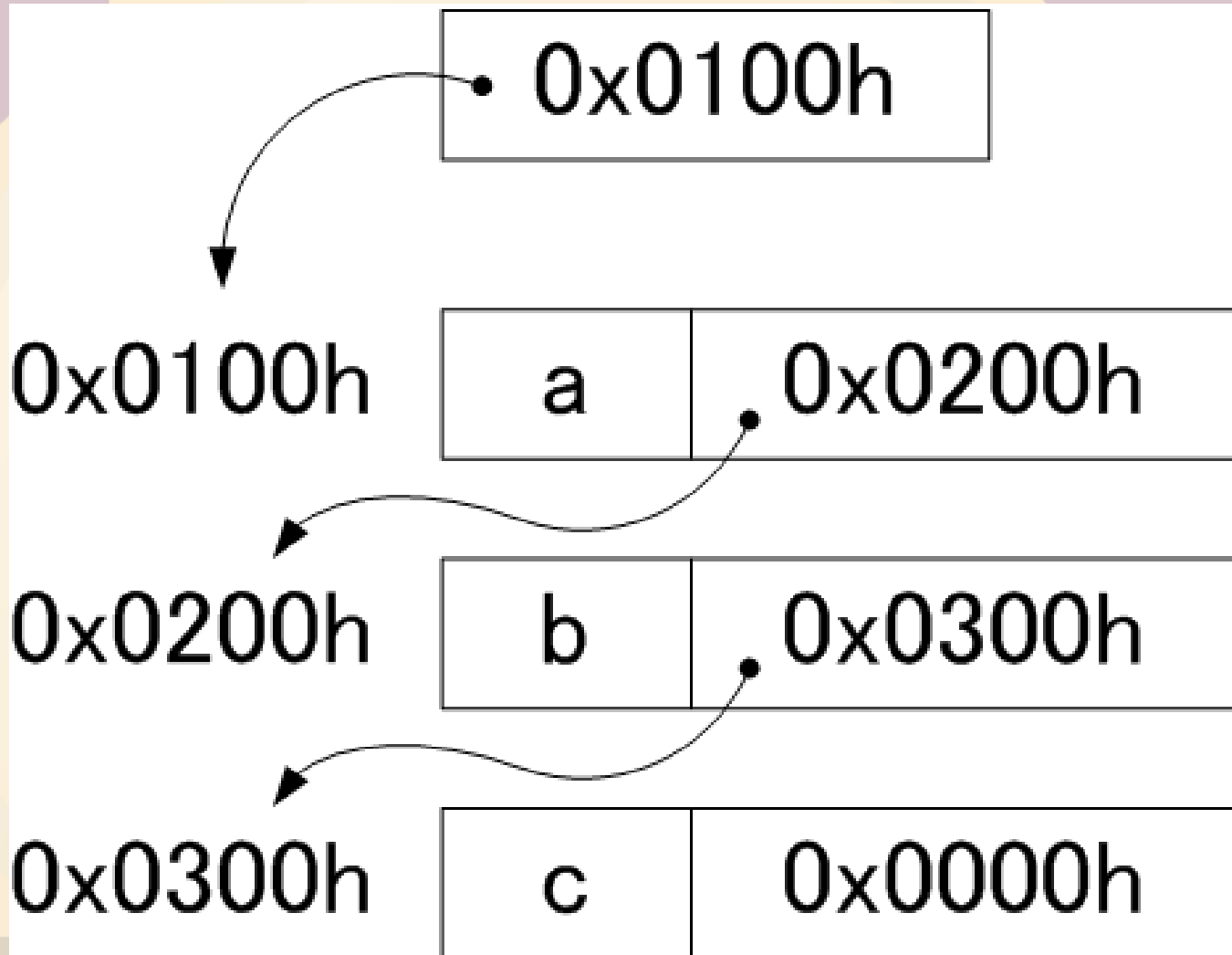
```
System.Console.ReadKey(true)
```



絵にしてみた



関数型言語の特徴 リスト構造をおさらい



無名関数

いちいち関数を

```
let square x = x * x
```

って書くのは面倒ですよ？

その場合無名関数を使います。

```
List.map square numbers
```

を

```
List.map (fun x -> x*x) numbers;
```

と記述することが出来ます。

カーリー化

別に何も難しいことはありません

```
> let add a b = a + b;;
```

```
val add : int -> int -> int
```

```
> add 1 2;;
```

```
val it : int = 3
```

```
> let add1 a = add 1 a;;
```

```
val add1 : int -> int
```

```
> add1 3;;
```

```
val it : int = 4
```



Tuple

```
> [1,2,3];;
```

```
val it : (int * int * int) list = [(1, 2, 3)]
```

```
> [1,"abc"];;
```

```
val it : (int * string) list = [(1, "abc")]
```

```
> [[1,'a'];[2,'b']];;
```

```
val it : (int * char) list list = [[(1, 'a')]; [(2, 'b')]]
```



List

- IntのList

```
> [1;2;3];;
```

```
val it : int list = [1; 2; 3]
```

```
> [1..3];;
```

```
val it : int list = [1; 2; 3]
```

```
> [1..2..10];;
```

```
val it : int list = [1; 3; 5; 7; 9]
```

List.map fun list

すべてのList要素に関数を適用して、新しいリストを定義する。

List.length list

List.max

List.sum



Array

```
> [|1;2;3|];;
```

```
val it : int array = [|1; 2; 3|]
```

```
>let arr = Array.create 2 ""
```

```
>arr.[0] <- "Hello, "
```

```
>arr.[1] <- "F# world"
```

```
> Array.map (fun x -> printfn "%A" x) arr;;
```

```
"Hello, "
```

```
"F# world"
```

```
val it : unit [] = [|null; null|]
```



for

- シンプルループ

```
for i = 0 to 100 do
```

```
    printfn "%d" i
```

- 列挙ループ

```
[for i in 0..100 -> i*i]
```

パイプ

```
let pow x = x*x;;  
val pow : int -> int  
> pow 3;;  
val it : int = 9  
  
> 4 |> pow;;  
val it : int = 16  
  
> pow <| 5;;  
val it : int = 25
```

各種ライブラリ

名前	更新日時	種類	サイズ
math	2009/03/14 1:00	ファイル フォルダ	
array.fs	2008/09/04 13:47	Visual F# Source file	15 KB
array.fsi	2008/09/04 13:47	Visual F# Signatur...	15 KB
array2.fs	2008/09/04 13:46	Visual F# Source file	5 KB
array2.fsi	2008/09/04 13:46	Visual F# Signatur...	5 KB
array3.fs	2008/09/04 13:47	Visual F# Source file	3 KB
array3.fsi	2008/09/04 13:47	Visual F# Signatur...	3 KB
collections.fs	2008/09/04 13:46	Visual F# Source file	6 KB
collections.fsi	2008/09/04 13:46	Visual F# Signatur...	3 KB
control.fs	2008/09/04 13:47	Visual F# Source file	43 KB
control.fsi	2008/09/04 13:47	Visual F# Signatur...	28 KB
event.fs	2008/09/04 13:47	Visual F# Source file	6 KB
event.fsi	2008/09/04 13:47	Visual F# Signatur...	7 KB
fslib-extra-pervasives.fs	2008/09/04 13:47	Visual F# Source file	7 KB
fslib-extra-pervasives.fsi	2008/09/04 13:47	Visual F# Signatur...	6 KB
list.fs	2008/09/04 13:46	Visual F# Source file	13 KB
list.fsi	2008/09/04 13:46	Visual F# Signatur...	16 KB
local.fs	2008/09/04 13:46	Visual F# Source file	19 KB
local.fsi	2008/09/04 13:46	Visual F# Signatur...	3 KB
map.fs	2008/09/04 13:47	Visual F# Source file	23 KB
map.fsi	2008/09/04 13:47	Visual F# Signatur...	11 KB
nativeptr.fs	2008/09/04 13:47	Visual F# Source file	2 KB
nativeptr.fsi	2008/09/04 13:47	Visual F# Signatur...	2 KB
option.fs	2008/09/04 13:46	Visual F# Source file	2 KB

array.fs 更新日時: 2008/09/04 13:47
Visual F# Source file サイズ: 14.5 KB
作成日時: 2009/03/14 1:00

.NET使おうよね

```
> 4u.ToString();  
val it : string = "4"  
  
> open System.Collections.Generic  
> let lst = List<string>();  
val lst : List<string>  
> lst.Add("ABC");  
val it : unit = ()  
> let printfn2 x = printfn "%A" x;  
  
> Seq.map (fun x -> printfn "%A" x) lst;  
"ABC"  
"DEF"  
"GHI"  
val it : seq<unit> = seq [null; null; null]
```



.NET使おうよね2

```
#light
open System.IO
let stream filename =
    System.IO.File.Open(filename,
        FileMode.Open)
let reader = new StreamReader(stream
    "C:¥¥temp¥¥a.txt")
let filevalue = reader.ReadToEnd()
(fun x -> printfn "%A" x) filevalue
reader.Close()
```



まとめ+次回予告

- 今回は難しいのであえてはずした
- 型作成
- パターンマッチング
- 非同期

- F# Developer Center
- <http://msdn.microsoft.com/en-us/fsharp/default.aspx>
- Spec
- <http://research.microsoft.com/en-us/um/cambridge/projects/fsharp/manual/spec2.aspx>
- Library
- <http://research.microsoft.com/en-us/um/cambridge/projects/fsharp/manual/namespaces.html>



- Blogなど
- <http://msdn.microsoft.com/ja-jp/magazine/cc164244.aspx>
- http://blogs.msdn.com/dd_jpn/archive/2008/07/03/8684353.aspx
- Wikipedia
- http://ja.wikipedia.org/wiki/F_Sharp

