

# Intro to OCaml.

EFC // Lecture 3  
2017

# History

- Alonzo Church : Lambda Calculus (1930s)
- Peter Landin : ISWIM (1960s)
- Robin Milner : ML (1973)
- Gerard Huet : Caml (1985)
- Xavier Leroy : **OCaml** (1996)
- Don Syme : F# (2005)

# Influenced

- Haskell (independent)
- Clojure (Lisp)
- Erlang (Eriksson)
- F# (.NET)
- Scala (Java)
- Rust (Mozilla)

# Users / promoters

- Jane Street : proprietary financial trading
- Facebook : code analysis and management (**Infer**)
- Microsoft : programming language tools (F#)
- Citrix : server virtualisation
- Galois : cryptographic algorithms
- OCamlPro : consulting services
- Bloomberg : financial information

# Set up

- UG Lab: **module load spam**
  - add this to the **.profile** file
- Home: pointers on Canvas

Online tutorial <http://try.ocamlpro.com/>

Online books  
(a bit advanced) <https://realworldocaml.org/> ... advanced  
<http://www.cl.cam.ac.uk/~lp15/MLbook/pub-details.html> ... SML

(REPL demo)

Understanding  
*evaluation*: rewriting

global  
definition

```
let x = 1;;
```

```
x + x;;
```

“local”  
definition

```
let x = 2 in x + x;;
```

```
x + x;;
```



```
let x = 1;;
```

```
x + x;;
```

```
let x = 2 in x + x;;
```

```
x + x;;
```

x=1

x + x;;

let x = 2 in x + x;;

x + x;;

x=1

x + x;;

let x = 2 in x + x;;

x + x;;

x=1

1 + x;;

let x = 2 in x + x;;

x + x;;

x=1

1 + x;;

let x = 2 in x + x;;

x + x;;

x=1

1 + 1;;

let x = 2 in x + x;;

x + x;;

x=1

1 + 1;;

let x = 2 in x + x;;

x + x;;

x=1

2;;

let x = 2 in x + x;;

x + x;;



x=1

```
let x = 2 in x + x;;
```

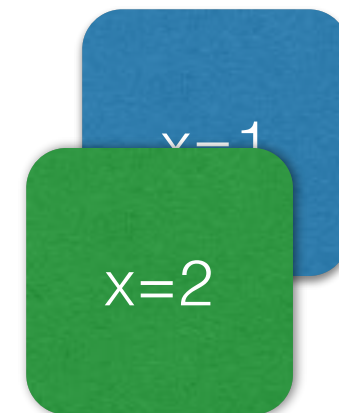
```
x + x;;
```

x=1

“local”  
definition

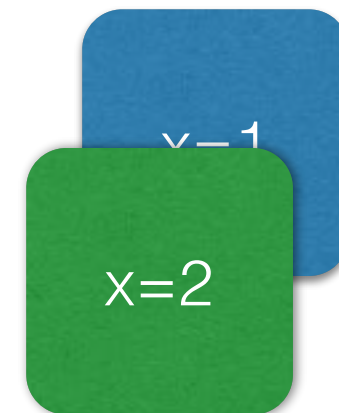
let x = 2 in x + x;;

x + x;;



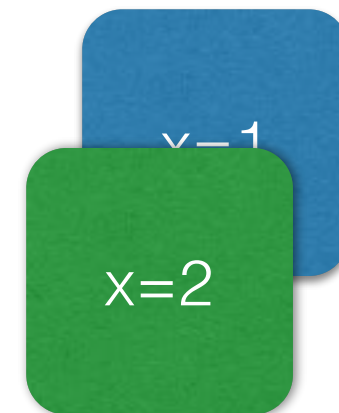
$X + X ; ;$

$X + X ; ;$



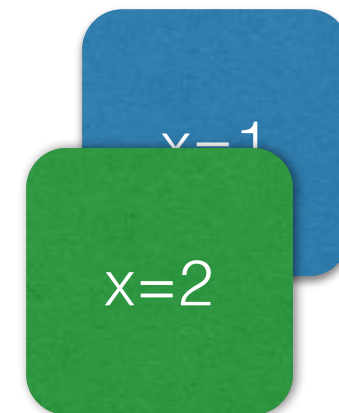
~~X~~ + X ; ;

X + X ; ;



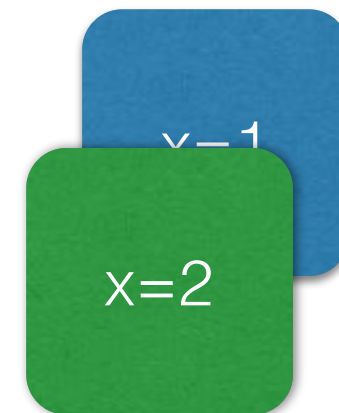
2 + x ; ;

x + x ; ;



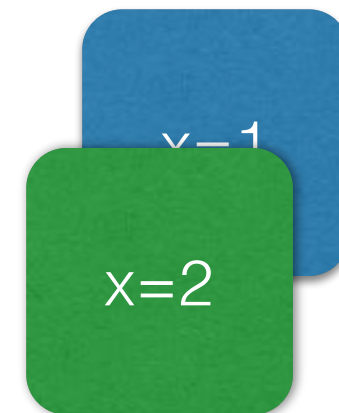
2 + x ; ;

x + x ; ;



2 + 2 ; ;

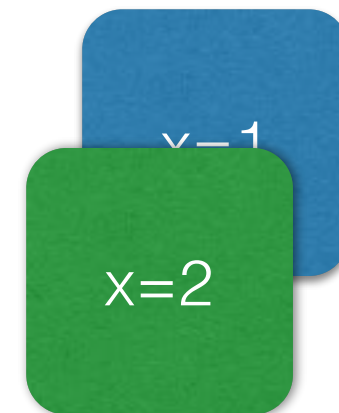
x + x ; ;



2 + 2 ; ;

x + x ; ;





4 ::

x + x ::

x=1

x + x ; ;

x=1

x + x ; ;



1 + x;;

x=1

1 + x ; ;

x=1

1 + 1 ; ;


x=1

1 + 1::



2 ::




$$x=1$$

Test your understanding!

<http://bit.ly/focs03a>



Answers

“local”  
definition

let x = 1 in

(let x = 2 in x + x) + x;;

“local”  
definition

```
let x = 1 in
```

```
(let x = 2 in x + x) + x;;
```

x=1

`(let x = 2 in x + x) + x;;`

x=1

(let x = 2 in x + x) + x;;



$(x + x) + x;;$



$(x + x) + x ; ;$





$(2 + x) + x;;$



$(2 + x) + x;;$



(2 + 2) + x;;



$(2 + 2) + x;;$



(4) + x ; ;

x=1

4 + x ; ;

4 + 1 ; ;

x=1


x=1

4 + 1;;



5 ; ;

$x=1$


$$x=1$$

Test your understanding

<http://bit.ly/focs03b>



Answers

# The Birmingham musical education

<https://www.youtube.com/watch?v=aQUIA8Hcv4s>