

Last login: Mon Feb 27 14:29:22 on ttys004
carbon:SamplePrograms\$ cd Sec_10_3\:35pm/
carbon:Sec_10_3:35pm\$ utop

Welcome to utop version 1.14 (using OCaml version 4.01.0)!

Type #utop_help for help about using utop.

```
-( 18:00:00 )-< command 0 >-----{ counter: 0 }-
utop # let add = fun x -> fun y -> x + y ;;
val add : int -> int -> int = <fun>
-( 15:50:55 )-< command 1 >-----{ counter: 0 }-
utop # add 3 4 ;;
- : int = 7
-( 15:51:23 )-< command 2 >-----{ counter: 0 }-
utop # #use "expr_let.ml";;
type expr =
  Int of int
  | Add of expr * expr
  | Sub of expr * expr
  | Mul of expr * expr
  | Div of expr * expr
  | Let of string * expr * expr
  | Id of string
type environment = (string * int) list
val lookup : string -> environment -> int = <fun>
val eval : environment -> expr -> int = <fun>
val e2 : expr = Let ("x", Int 5, Add (Id "x", Int 4))
-( 15:51:25 )-< command 3 >-----{ counter: 0 }-
utop # eval [("x", Int 3)] (Add (Id "x", Value (Int 1))) ;;
Error: This expression has type expr but an expression was expected of type int
-( 16:00:48 )-< command 4 >-----{ counter: 0 }-
utop #
```

| | | | | | | | | | |
|-----|-----|--------------|-------|-------------|----------------|---------|----------|--------|--|
| Add | Arg | Arith_status | Array | ArrayLabels | Assert_failure | Big_int | Bigarray | Buffer | |
|-----|-----|--------------|-------|-------------|----------------|---------|----------|--------|--|