

Objective Caml version 3.06

```

# ref;;
- : 'a -> 'a ref = <fun>

# let x = ref 0;;
val x : int ref = {contents = 0}

# x;;
- : int ref = {contents = 0}

# (!);;
- : 'a ref -> 'a = <fun>

# ! x;;
- : int = 0

# x.contents;;
- : int = 0

# (:=);;
- : 'a ref -> 'a -> unit = <fun>

# x := 2;;
- : unit = ()

# ! x;;
- : int = 2

# x := x + 3;;
Characters 5-6:
  x := x + 3;;
      ^

This expression has type int ref but is here used with type int

# x := !x + 3;;
- : unit = ()

# x := 0;
for i = 1 to 100 do x := !x + i done;
!x;;
- : int = 5050

# let y = ref 0;;
val y : int ref = {contents = 0}

# x := 1;
y := 0;
while !x < 1000 do x := !x + !x; y := !y + 1 done;
!y;;
- : int = 10

# let (inc, get) =
  let x = ref 0
  in ((fun () -> x := !x + 1),
      (fun () -> !x));;
  val inc : unit -> unit = <fun>
val get : unit -> int = <fun>

# inc ();;
- : unit = ()

# get ();;
- : int = 1

#

```