

Objective Caml version 3.06

```

# type a = int;;
type a = int

# let x: a = 1;;
val x : a = 1

# type b = int;;
type b = int

# let y: b = 2;;
val y : b = 2

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

# x = y;;
- : bool = false

# let (f: a -> b) = fun x -> x;;
val f : a -> b = <fun>

# let x = {u = 1; v = 2};;
type r = {u: int; v: int};;
Characters 8-22:
  let x = {u = 1; v = 2};;
          ^^^^^^^^^^^^^^^^^
Unbound record field label u

# type r = { u : int; v : int; }

# let x: r = {u = 1; v = 2};;
val x : r = {u = 1; v = 2}

# type s = {u: int; v: int};;
type s = { u : int; v : int; }

# let y: s = {u = 1; v = 2};;
val y : s = {u = 1; v = 2}

# x = y;;
Characters 4-5:
  x = y;;
  ^
This expression has type s but is here used with type r

# y.u;;
- : int = 1

# x.u;;
Characters 0-1:
  x.u;;
  ^
This expression has type r but is here used with type s

# let f x = x.u;;
val f : s -> int = <fun>

# type r = A of int;;
type r = A of int

# let x: r = A 1;;
val x : r = A 1

# type s = A of int;;
type s = A of int

# let y: s = A 2;;
val y : s = A 2

# x = y;;
Characters 4-5:
  x = y;;
  ^
This expression has type s but is here used with type r

```