```
Last login: Fri Mar 23 13:47:19 on ttys006
carbon:$ pwd
/project/evw/Teaching/18_Spring_2041/carbon-repos/public-class-repo/Sam
ple Programs/Sec_01_1-25pm/Intervals
carbon:$ ls
README.md       v2/             v4/             v6/
v1/             v3/             v5/             v7/
carbon:$ cd v4
carbon:$ ocamlbuild useInterval.byte
Finished, 9 targets (9 cached) in 00:00:00.
carbon:$ ./useInterval.byte
An interval: (3, 4)
Another interval: (3, 6)
Their intresection: (3, 4)
A string interval: (a, d)
carbon:$ up
/project/evw/Teaching/18_Spring_2041/carbon-repos/public-class-repo/Sam
ple Programs/Sec_01_1-25pm/Intervals
carbon:$ cd v5
carbon:$ ls
_build/         intInterval.ml  intervals.ml
carbon:$ ocamlbuild intInterval.byte
+ /Users/evw/.opam/4.06.0/bin/ocamlc.opt -c -o intInterval.cmo intInter
val.ml
File "intInterval.ml", line 35, characters 28-29:
Error: This expression has type int but an expression was expected of t
ype
         Int_interval.endpoint =
           Intervals.Make_interval(Int_comparable).endpoint
Command exited with code 2.
Hint: Recursive traversal of subdirectories was not enabled for this bu
ild,
  as the working directory does not look like an ocamlbuild project (no
  '_tags' or 'myocamlbuild.ml' file). If you have modules in subdirecto
ries,
  you should add the option "-r" or create an empty '_tags' file.

  To enable recursive traversal for some subdirectories only, you can u
se the
  following '_tags' file:

      true: -traverse
      <dir1> or <dir2>: traverse

Compilation unsuccessful after building 4 targets (3 cached) in 00:00:0
0.
carbon:$
```

```
carbon:$ utop
──────┬─────────────────────────────────────────────────────────────┬──────
      │ Welcome to utop version 2.0.2 (using OCaml version 4.06.0)! │
      └─────────────────────────────────────────────────────────────┘

Type #utop_help for help about using utop.

─( 13:53:22 )─< command 0 >──────────────────────────────{ counter: 0 }─
utop # #mod_use "intervals.ml";;
module Intervals :
  sig
    module type Comparable =
      sig
        type t
        val compare : t -> t -> int
        val to_string : t -> string
      end
    module type Interval_intf =
      sig
        type t
        type endpoint
        val create : endpoint -> endpoint -> t
        val is_empty : t -> bool
        val contains : t -> endpoint -> bool
        val intersect : t -> t -> t
        val to_string : t -> string
      end
    module Make_interval :
      functor (Endpoint : Comparable) -> Interval_intf
  end
─( 13:53:29 )─< command 1 >──────────────────────────────{ counter: 0 }─
utop # #use "intInterval.ml";;
module Int_comparable :
  sig
    type t = int
    val compare : 'a -> 'a -> t
    val to_string : t -> string
  end
module Int_interval :
  sig
    type t = Intervals.Make_interval(Int_comparable).t
    type endpoint = Intervals.Make_interval(Int_comparable).endpoint
    val create : endpoint -> endpoint -> t
    val is_empty : t -> bool
    val contains : t -> endpoint -> bool
    val intersect : t -> t -> t
    val to_string : t -> string
  end
File "intInterval.ml", line 35, characters 28-29:
Error: This expression has type int
       but an expression was expected of type Int_interval.endpoint
─( 13:53:29 )─< command 2 >──────────────────────────────{ counter: 0 }─
utop #
 │Arg│Array│ArrayLabels│Assert_failure│Bigarray│Buffer│Bytes│BytesLabel│
```