```
Last login: Wed Dec  3 10:42:41 on ttys003
carbon:Intervals$ cd v5
carbon:v5$ ls
intInterval.ml  intervals.ml
carbon:v5$ utop
────────────────────────────────────────────────────────────────────────────────
         │ Welcome to utop version 1.14 (using OCaml version 4.01.0)! │
         └────────────────────────────────────────────────────────────┘

Findlib has been successfully loaded. Additional directives:
  #require "package";;      to load a package
  #list;;                   to list the available packages
  #camlp4o;;                to load camlp4 (standard syntax)
  #camlp4r;;                to load camlp4 (revised syntax)
  #predicates "p,q,...";;   to set these predicates
  Topfind.reset();;         to force that packages will be reloaded
  #thread;;                 to enable threads


Type #utop_help for help about using utop.

─( 18:00:00 )─< 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
─( 11:22:49 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "intInterval.ml";;
module Int_interval :
  sig
    type t = Intervals.Make_interval(Core.Std.Int).t
    type endpoint = Intervals.Make_interval(Core.Std.Int).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 22, characters 28-29:
Error: This expression has type int but an expression was expected of type
         Int_interval.endpoint
```

```
─( 11:23:01 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
carbon:v5$ cd ../v6
carbon:v6$ utop
─────────┬──────────────────────────────────────────────────────────────┬───────
         │ Welcome to utop version 1.14 (using OCaml version 4.01.0)! │
         └──────────────────────────────────────────────────────────────┘

Findlib has been successfully loaded. Additional directives:
  #require "package";;      to load a package
  #list;;                   to list the available packages
  #camlp4o;;                to load camlp4 (standard syntax)
  #camlp4r;;                to load camlp4 (revised syntax)
  #predicates "p,q,...";;   to set these predicates
  Topfind.reset();;         to force that packages will be reloaded
  #thread;;                 to enable threads


Type #utop_help for help about using utop.

─( 18:00:00 )─< 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) ->
        sig
          type t
          type endpoint = Endpoint.t
          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
  end
─( 11:29:02 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "intInterval.ml";;
module Int_comparable :
  sig type t = int val compare : t -> t -> t val to_string : t -> string end
module Int_interval :
  sig
```

```
    type t = Intervals.Make_interval(Int_comparable).t
    type endpoint = int
    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
val i : Int_interval.t = <abstr>
─( 11:29:09 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # #use "intInterval.ml";;
module Int_comparable : Intervals.Comparable
module Int_interval :
  sig
    type t = Intervals.Make_interval(Int_comparable).t
    type endpoint = Int_comparable.t
    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 16, characters 28-29:
Error: This expression has type int but an expression was expected of type
         Int_interval.endpoint
─( 11:29:24 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;;
carbon:v6$ cd ../v7
carbon:v7$ utop
─────────┬─────────────────────────────────────────────────────────────┬────────
         │ Welcome to utop version 1.14 (using OCaml version 4.01.0)! │
         └─────────────────────────────────────────────────────────────┘

Findlib has been successfully loaded. Additional directives:
  #require "package";;      to load a package
  #list;;                   to list the available packages
  #camlp4o;;                to load camlp4 (standard syntax)
  #camlp4r;;                to load camlp4 (revised syntax)
  #predicates "p,q,...";;   to set these predicates
  Topfind.reset();;         to force that packages will be reloaded
  #thread;;                 to enable threads


Type #utop_help for help about using utop.

─( 18:00:00 )─< 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) ->
        sig
          type t
          val create : Endpoint.t -> Endpoint.t -> t
          val is_empty : t -> bool
          val contains : t -> Endpoint.t -> bool
          val intersect : t -> t -> t
          val to_string : t -> string
        end
  end
─( 11:41:57 )─< command 1 >───────────────────────────────────────{ counter: 0 }
─top #
utop #
┌───┬────────────┬─────┬───────────┬──────────────┬─────────┬───┬───────┬───────
─( 11:41:57 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
carbon:v7$
```