```
Last login: Mon Dec  1 11:03:58 on ttys006
carbon:code-examples$ cd Intervals/
carbon:Intervals$ ls
ReadMe.md       v2/             v4/             v6/
v1/             v3/             v5/             v7/
carbon:Intervals$ cd ../
carbon:code-examples$ 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 # #use "session_info.ml";;
module type ID =
  sig type t val of_string : string -> t val to_string : t -> string end
module String_id :
  sig type t = string val of_string : 'a -> 'a val to_string : 'a -> 'a end
module Username : ID
module Hostname : ID
type session_info = {
  user : Username.t;
  host : Hostname.t;
  when_started : Time.t;
}
File "session_info.ml", line 31, characters 12-19:
Error: This expression has type Hostname.t
       but an expression was expected of type Username.t
─( 11:31:59 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # #use "session_info.ml";;
module type ID =
  sig type t val of_string : string -> t val to_string : t -> string end
module String_id :
  sig type t = string val of_string : 'a -> 'a val to_string : 'a -> 'a end
module Username : ID
module Hostname : ID
type session_info = {
  user : Username.t;
  host : Hostname.t;
  when_started : Time.t;
}
val sessions_have_same_user : session_info -> session_info -> bool = <fun>
─( 11:32:05 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
```

```
utop # #use "session_info.ml";;
module type ID =
  sig
    type t
    val of_string : string -> t
    val to_string : t -> string
    val append : t -> t -> t
  end
module String_id :
  sig type t = string val of_string : 'a -> 'a val to_string : 'a -> 'a end
File "session_info.ml", line 22, characters 23-32:
Error: Signature mismatch:
       Modules do not match:
         sig
           type t = string
           val of_string : 'a -> 'a
           val to_string : 'a -> 'a
         end
       is not included in
         ID
       The field `append' is required but not provided
─( 11:33:24 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # #use "session_info.ml";;
module type ID =
  sig type t val of_string : string -> t val to_string : t -> string end
module String_id :
  sig
    type t = string
    val of_string : 'a -> 'a
    val to_string : 'a -> 'a
    val append : t -> t -> t
  end
module Username : ID
module Hostname : ID
type session_info = {
  user : Username.t;
  host : Hostname.t;
  when_started : Time.t;
}
val sessions_have_same_user : session_info -> session_info -> bool = <fun>
─( 11:36:14 )─< command 4 >──────────────────────────────────────{ counter: 0 }─
utop # #use "session_info.ml";;
module type ID =
  sig type t val of_string : string -> t val to_string : t -> string end
module String_id :
  sig
    type t = string
    val of_string : 'a -> 'a
    val to_string : 'a -> 'a
    val append : t -> t -> t
  end
module Username : ID
module Hostname : ID
```

```
type session_info = {
  user : Username.t;
  host : Hostname.t;
  when_started : Time.t;
}
val sessions_have_same_user : session_info -> session_info -> bool = <fun>
File "session_info.ml", line 34, characters 42-44:
Error: This expression has type Username.t
       but an expression was expected of type string
─( 11:37:30 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # #use "session_info.ml";;
module type ID =
  sig type t val of_string : string -> t val to_string : t -> string end
module String_id :
  sig
    type t = string
    val of_string : 'a -> 'a
    val to_string : 'a -> 'a
    val append : t -> t -> t
  end
module Username : ID
module Hostname : ID
type session_info = {
  user : Username.t;
  host : Hostname.t;
  when_started : Time.t;
}
val sessions_have_same_user : session_info -> session_info -> bool = <fun>
val app : Username.t -> Username.t -> Username.t = <fun>
─( 11:40:06 )─< command 6 >──────────────────────────────────────{ counter: 0 }─
utop # #quit ;;
carbon:code-examples$ cd Intervals/v3
carbon:v3$ corebuild -clean
carbon:v3$ ls
intervals.ml            intervals.ps            useIntInterval.ml
intervals.ml~           useIntInterval.byte@    useIntInterval.ml~
carbon:v3$ ec
carbon:v3$ rm intervals.ps
remove intervals.ps? y
carbon:v3$ ls
intervals.ml            useIntInterval.byte@    useIntInterval.ml
carbon:v3$ rm useIntInterval.byte
remove useIntInterval.byte? y
carbon:v3$ ls
intervals.ml            useIntInterval.ml
carbon:v3$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 useIntInterval.ml > useIntInterval.ml.depends
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intervals.ml > intervals.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
```

```
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intervals.cmo intervals.ml
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core intervals.cmo useIntInterval.cmo -o useIntInterval.byte
carbon:v3$ ./useIntInterval.byte
An interval: (3, 4)
Another interval: (3, 6)
Their intresection: (3, 4)
carbon:v3$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intervals.ml > intervals.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intervals.cmo intervals.ml
+ ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bi
n-annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package s
explib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package cor
e -o intervals.cmo intervals.ml
File "intervals.ml", line 27, characters 18-29:
Error: Unbound type constructor intInterval
Command exited with code 2.
carbon:v3$ corebuild useIntInterval.byte
ocamlfind ocamldep -syntax camlp4o -package bin_prot.syntax -package sexplib.syn
tax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core -modules
 intervals.ml > intervals.ml.depends
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o intervals.cmo intervals.ml
ocamlfind ocamlc -c -w A-4-33-40-41-42-43-34-44 -strict-sequence -g -annot -bin-
annot -short-paths -thread -syntax camlp4o -package bin_prot.syntax -package sex
plib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -package core
-o useIntInterval.cmo useIntInterval.ml
ocamlfind ocamlc -linkpkg -g -thread -syntax camlp4o -package bin_prot.syntax -p
ackage sexplib.syntax,comparelib.syntax,fieldslib.syntax,variantslib.syntax -pac
kage core intervals.cmo useIntInterval.cmo -o useIntInterval.byte
carbon:v3$ corebuild useIntInterval.byte
carbon:v3$ cd ../v4
carbon:v4$
```