```
Last login: Fri Mar  9 11:15:26 on ttys003
carbon:$ utop
───────────┬──────────────────────────────────────────────────────────────┬───────────
           │ Welcome to utop version 2.0.2 (using OCaml version 4.06.0)! │
           └──────────────────────────────────────────────────────────────┘

Type #utop_help for help about using utop.

─( 14:08:24 )─< command 0 >────────────────────────────────────{ counter: 0 }─
utop # #use "lazy.ml";;
type 'a lazee = 'a hidden ref
and 'a hidden = Value of 'a | Thunk of (unit -> 'a)
val delay : (unit -> 'a) -> 'a lazee = <fun>
val force : 'a lazee -> unit = <fun>
val demand : 'a lazee -> 'a = <fun>
type 'a stream = Cons of 'a * 'a stream lazee
val from : int -> int stream = <fun>
step 1
val nats : int stream = Cons (1, {contents = Thunk <fun>})
─( 14:08:24 )─< command 1 >────────────────────────────────────{ counter: 0 }─
utop # nats ;;
- : int stream = Cons (1, {contents = Thunk <fun>})
─( 14:08:27 )─< command 2 >────────────────────────────────────{ counter: 0 }─
utop # nats ;;
- : int stream = Cons (1, {contents = Thunk <fun>})
─( 14:08:31 )─< command 3 >────────────────────────────────────{ counter: 0 }─
utop # hd nats ;;
Error: Unbound value hd
─( 14:11:14 )─< command 4 >────────────────────────────────────{ counter: 0 }─
utop # #use "lazy.ml";;
type 'a lazee = 'a hidden ref
and 'a hidden = Value of 'a | Thunk of (unit -> 'a)
val delay : (unit -> 'a) -> 'a lazee = <fun>
val force : 'a lazee -> unit = <fun>
val demand : 'a lazee -> 'a = <fun>
type 'a stream = Cons of 'a * 'a stream lazee
val from : int -> int stream = <fun>
step 1
val nats : int stream = Cons (1, {contents = Thunk <fun>})
─( 14:11:20 )─< command 5 >────────────────────────────────────{ counter: 0 }─
utop # hd nats ;;
Error: Unbound value hd
─( 14:11:23 )─< command 6 >────────────────────────────────────{ counter: 0 }─
utop # #use "lazy.ml";;\
type 'a lazee = 'a hidden ref
and 'a hidden = Value of 'a | Thunk of (unit -> 'a)
val delay : (unit -> 'a) -> 'a lazee = <fun>
val force : 'a lazee -> unit = <fun>
val demand : 'a lazee -> 'a = <fun>
type 'a stream = Cons of 'a * 'a stream lazee
val from : int -> int stream = <fun>
step 1
```

```
val nats : int stream = Cons (1, {contents = Thunk <fun>})
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val take : int -> 'a stream -> 'a list = <fun>
─( 14:11:27 )─< command 7 >────────────────────────────────────{ counter: 0 }─
utop # hd nats ;;
Error: Unbound value hd
─( 14:11:37 )─< command 8 >────────────────────────────────────{ counter: 0 }─
utop # #use "lazy.ml";;
type 'a lazee = 'a hidden ref
and 'a hidden = Value of 'a | Thunk of (unit -> 'a)
val delay : (unit -> 'a) -> 'a lazee = <fun>
val force : 'a lazee -> unit = <fun>
val demand : 'a lazee -> 'a = <fun>
type 'a stream = Cons of 'a * 'a stream lazee
val from : int -> int stream = <fun>
step 1
val nats : int stream = Cons (1, {contents = Thunk <fun>})
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val take : int -> 'a stream -> 'a list = <fun>
─( 14:11:39 )─< command 9 >────────────────────────────────────{ counter: 0 }─
utop # hd nats ;;
Error: Unbound value hd
─( 14:11:43 )─< command 10 >───────────────────────────────────{ counter: 0 }─
utop # head nats ;;
- : int = 1
─( 14:11:44 )─< command 11 >───────────────────────────────────{ counter: 0 }─
utop # tail nats ;;
step 2
- : int stream = Cons (2, {contents = Thunk <fun>})
─( 14:11:51 )─< command 12 >───────────────────────────────────{ counter: 0 }─
utop # nats ;;
- : int stream =
Cons (1, {contents = Value (Cons (2, {contents = Thunk <fun>}))})
─( 14:11:56 )─< command 13 >───────────────────────────────────{ counter: 0 }─
utop # take 5 nats ;;
step 3
step 4
step 5
step 6
- : int list = [1; 2; 3; 4; 5]
─( 14:12:09 )─< command 14 >───────────────────────────────────{ counter: 0 }─
utop # nats ;;
- : int stream =
Cons (1,
 {contents =
   Value
    (Cons (2,
      {contents =
        Value
         (Cons (3,
           {contents =
             Value
              (Cons (4,
                {contents =
                  Value
                   (Cons (5,
                     {contents = Value (Cons (6, {contents = Thunk <fun>}))}))
}))}))}))})
─( 14:12:49 )─< command 15 >───────────────────────────────────{ counter: 0 }─
utop # take 5 nats ;;
- : int list = [1; 2; 3; 4; 5]
─( 14:13:25 )─< command 16 >───────────────────────────────────{ counter: 0 }─
utop # #quit ;;
carbon:$
```