```
Last login: Fri Mar 24 13:42:49 on ttys006
carbon:SamplePrograms$ uto[
>
> ;
> -bash: unexpected EOF while looking for matching `]'
-bash: syntax error: unexpected end of file
carbon:SamplePrograms$ utop
```

```
        Welcome to utop version 1.14 (using OCaml version 4.01.0)!

Type #utop_help for help about using utop.

─( 18:00:00 )─< command 0 >──────────────────────────────────────{ counter: 0 }─
utop # #use "streams.ml";;
type 'a stream = Cons of 'a * (unit -> 'a stream)
val ones : int stream = Cons (1, <fun>)
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val from : int -> int stream = <fun>
val nats : int stream = Cons (1, <fun>)
val take : int -> 'a stream -> 'a list = <fun>
val filter : ('a -> bool) -> 'a stream -> 'a stream = <fun>
val filter : ('a -> bool) -> 'a stream -> 'a stream = <fun>
val even : int -> bool = <fun>
val squares_from : int -> int stream = <fun>
val t1 : int list = [1; 4; 9; 16; 25; 36; 49; 64; 81; 100]
val squares : int stream = Cons (1, <fun>)
val zip : ('a -> 'b -> 'c) -> 'a stream -> 'b stream -> 'c stream = <fun>
val nats2 : int stream = Cons (1, <fun>)
val factorials : int stream = Cons (1, <fun>)
val non : ('a -> bool) -> 'a -> bool = <fun>
val multiple_of : int -> int -> bool = <fun>
val sift : int -> int stream -> int stream = <fun>
val sieve : int stream -> int stream = <fun>
val primes : int stream = Cons (2, <fun>)
─( 15:52:43 )─< command 1 >──────────────────────────────────────{ counter: 0 }─
utop # ones ;;
- : int stream = Cons (1, <fun>)
─( 15:53:02 )─< command 2 >──────────────────────────────────────{ counter: 0 }─
utop # match ones with Cons(_, f) -> f () ;;
- : int stream = Cons (1, <fun>)
─( 15:53:53 )─< command 3 >──────────────────────────────────────{ counter: 0 }─
utop # head ones ;;
- : int = 1
─( 15:53:53 )─< command 4 >──────────────────────────────────────{ counter: 0 }─
utop # head (tail (tail ones)) ;;
- : int = 1
─( 15:55:29 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # from 5 ;;
- : int stream = Cons (5, <fun>)
─( 15:55:38 )─< command 6 >──────────────────────────────────────{ counter: 0 }─
utop # head (tail (tail from 5)) ;;
```

```
Error: This function has type 'a stream -> 'a stream
       It is applied to too many arguments; maybe you forgot a `;'.
─( 15:57:02 )─< command 7 >──────────────────────────────────────{ counter: 0 }─
utop # head (tail (tail (from 5))) ;;
step 6
step 7
- : int = 7
─( 15:57:26 )─< command 8 >──────────────────────────────────────{ counter: 0 }─
utop # nats ;;
- : int stream = Cons (1, <fun>)
─( 15:57:34 )─< command 9 >──────────────────────────────────────{ counter: 0 }─
utop # head (tail (tail (tail nats))) ;;
step 2
step 3
step 4
- : int = 4
─( 15:58:46 )─< command 10 >─────────────────────────────────────{ counter: 0 }─
utop # take 10 nats ;;
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
- : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]
─( 15:58:55 )─< command 11 >─────────────────────────────────────{ counter: 0 }─
utop # #use "streams.ml";;
type 'a stream = Cons of 'a * (unit -> 'a stream)
val ones : int stream = Cons (1, <fun>)
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val from : int -> int stream = <fun>
val nats : int stream = Cons (1, <fun>)
val take : int -> 'a stream -> 'a list = <fun>
File "streams.ml", line 48, characters 49-51:
Error: This expression has type unit -> 'a stream
       but an expression was expected of type 'a stream
─( 16:08:40 )─< command 12 >─────────────────────────────────────{ counter: 0 }─
utop # #use "streams.ml";;
type 'a stream = Cons of 'a * (unit -> 'a stream)
val ones : int stream = Cons (1, <fun>)
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val from : int -> int stream = <fun>
val nats : int stream = Cons (1, <fun>)
val take : int -> 'a stream -> 'a list = <fun>
File "streams.ml", line 48, characters 40-56:
Error: This expression has type 'b stream
       but an expression was expected of type unit -> 'a stream
```

```
─( 16:08:40 )─< command 13 >──────────────────────────────────{ counter: 0 }─
utop # #use "streams.ml";;
type 'a stream = Cons of 'a * (unit -> 'a stream)
val ones : int stream = Cons (1, <fun>)
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val from : int -> int stream = <fun>
val nats : int stream = Cons (1, <fun>)
val take : int -> 'a stream -> 'a list = <fun>
File "streams.ml", line 49, characters 25-51:
Error: This expression should not be a function, the expected type is
'a stream
─( 16:10:05 )─< command 14 >──────────────────────────────────{ counter: 0 }─
utop # #use "streams.ml";;
type 'a stream = Cons of 'a * (unit -> 'a stream)
val ones : int stream = Cons (1, <fun>)
val head : 'a stream -> 'a = <fun>
val tail : 'a stream -> 'a stream = <fun>
val from : int -> int stream = <fun>
val nats : int stream = Cons (1, <fun>)
val take : int -> 'a stream -> 'a list = <fun>
val filter : ('a -> bool) -> 'a stream -> 'a stream = <fun>
val filter : ('a -> bool) -> 'a stream -> 'a stream = <fun>
val even : int -> bool = <fun>
val squares_from : int -> int stream = <fun>
val t1 : int list = [1; 4; 9; 16; 25; 36; 49; 64; 81; 100]
val squares : int stream = Cons (1, <fun>)
val zip : ('a -> 'b -> 'c) -> 'a stream -> 'b stream -> 'c stream = <fun>
val nats2 : int stream = Cons (1, <fun>)
val factorials : int stream = Cons (1, <fun>)
val non : ('a -> bool) -> 'a -> bool = <fun>
val multiple_of : int -> int -> bool = <fun>
val sift : int -> int stream -> int stream = <fun>
val sieve : int stream -> int stream = <fun>
val primes : int stream = Cons (2, <fun>)
─( 16:13:46 )─< command 15 >──────────────────────────────────{ counter: 0 }─
utop # even ;;
- : int -> bool = <fun>
─( 16:14:18 )─< command 16 >──────────────────────────────────{ counter: 0 }─
utop # even 2 ;;
- : bool = true
─( 16:15:04 )─< command 17 >──────────────────────────────────{ counter: 0 }─
utop # take 10 (filter even nats) ;;
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
```

```
step 12
step 13
step 14
step 15
step 16
step 17
step 18
step 19
step 20
step 21
step 22
- : int list = [2; 4; 6; 8; 10; 12; 14; 16; 18; 20]
─( 16:15:06 )─< command 18 >──────────────────────────────────{ counter: 0 }─
utop # take 10 (filter even ones) ;;
^CInterrupted.
─( 16:15:16 )─< command 19 >──────────────────────────────────{ counter: 0 }─
utop # take 10 (squares_from 4) ;;
- : int list = [16; 25; 36; 49; 64; 81; 100; 121; 144; 169]
─( 16:16:18 )─< command 20 >──────────────────────────────────{ counter: 0 }─
utop # take 10 (zip (fun x y -> x + y) nats nats) ;;
step 2
step 2
step 3
step 3
step 4
step 4
step 5
step 5
step 6
step 6
step 7
step 7
step 8
step 8
step 9
step 9
step 10
step 10
step 11
step 11
- : int list = [2; 4; 6; 8; 10; 12; 14; 16; 18; 20]
─( 16:17:29 )─< command 21 >──────────────────────────────────{ counter: 0 }─
utop # take 10 (zip (fun x y -> x + y) nats squares) ;;
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
```

```
- : int list = [2; 6; 12; 20; 30; 42; 56; 72; 90; 110]
-( 16:19:35 )-< command 22 >-------------------------------------{ counter: 0 }-
utop # take 5 nats2 ;;
- : int list = [1; 2; 3; 4; 5]
-( 16:19:47 )-< command 23 >-------------------------------------{ counter: 0 }-
utop # take 10 factorials ;;
step 2
step 2
step 3
step 2
step 3
step 4
step 2
step 3
step 4
step 5
step 2
step 3
step 4
step 5
step 6
step 2
step 3
step 4
step 5
step 6
step 7
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
- : int list = [1; 1; 2; 6; 24; 120; 720; 5040; 40320; 362880]
-( 16:21:15 )-< command 24 >-------------------------------------{ counter: 0 }-
```

```
utop # take 10 primes ;;
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
step 12
step 13
step 14
step 15
step 16
step 17
step 18
step 19
step 20
step 21
step 22
step 23
step 24
step 25
step 26
step 27
step 28
step 29
step 30
step 31
- : int list = [2; 3; 5; 7; 11; 13; 17; 19; 23; 29]
─( 16:23:49 )─< command 25 >──────────────────────────────────────{ counter: 0 }─
utop # take 10 nats  ;;
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
- : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]
─( 16:25:48 )─< command 26 >──────────────────────────────────────{ counter: 0 }─
utop # take 10 nats  ;;
step 2
step 3
step 4
step 5
step 6
step 7
step 8
```

```
step 9
step 10
step 11
- : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]
-( 16:26:08 )-< command 27 >-------------------------------------{ counter: 0 }-
utop # take 10 nats  ;;
step 2
step 3
step 4
step 5
step 6
step 7
step 8
step 9
step 10
step 11
- : int list = [1; 2; 3; 4; 5; 6; 7; 8; 9; 10]
-( 16:26:09 )-< command 28 >-------------------------------------{ counter: 0 }-
utop #
```

Arg	Arith_status	Array	ArrayLabels	Assert_failure	Big_int	Bigarray	Buffer	Ca