```
Last login: Fri Jan 26 13:23:35 on ttys005
carbon:$ utop
```

```
        Welcome to utop version 2.0.2 (using OCaml version 4.06.0)!

Type #utop_help for help about using utop.

─( 13:37:11 )─< command 0 >──────────────────────────────────{ counter: 0 }─
utop # #use "simple.ml";;
val power : int -> float -> float = <fun>
val cube : float -> float = <fun>
val gcd : int -> int -> int = <fun>
val sum : int list -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val is_even : int -> bool = <fun>
File "simple.ml", line 47, characters 6-10:
Warning 26: unused variable even.
val even2ways_better : int list -> bool = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val length : 'a list -> int = <fun>
─( 13:37:11 )─< command 1 >──────────────────────────────────{ counter: 0 }─
utop # #use "simple.ml";;
val power : int -> float -> float = <fun>
val cube : float -> float = <fun>
val gcd : int -> int -> int = <fun>
val sum : int list -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val is_even : int -> bool = <fun>
File "simple.ml", line 47, characters 6-10:
Warning 26: unused variable even.
val even2ways_better : int list -> bool = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val length : 'a list -> int = <fun>
File "simple.ml", line 81, characters 2-32:
Warning 8: this pattern-matching is not exhaustive.
Here is an example of a case that is not matched:
[]
val head : 'a list -> 'a = <fun>
─( 13:37:13 )─< command 2 >──────────────────────────────────{ counter: 0 }─
utop # head [1;2;3] ;;
- : int = 1
─( 13:37:32 )─< command 3 >──────────────────────────────────{ counter: 0 }─
utop # head 1::2::3::[] ;;
Error: This expression has type int but an expression was expected of type
         'a list
─( 13:37:37 )─< command 4 >──────────────────────────────────{ counter: 0 }─
utop # head (1::2::3::[]0 ;;
Error: Syntax error: ')' expected, the highlighted '(' might be unmatched
```

```
─( 13:37:44 )─< command 5 >──────────────────────────────────────{ counter: 0 }─
utop # head (1::2::3::[]) ;;
- : int = 1
─( 13:37:49 )─< command 6 >──────────────────────────────────────{ counter: 0 }─
utop # head [] ;;
Exception: Match_failure ("simple.ml", 81, 2).
─( 13:37:53 )─< command 7 >──────────────────────────────────────{ counter: 0 }─
utop # (1, "hello") ;;
- : int * string = (1, "hello")
─( 13:38:02 )─< command 8 >──────────────────────────────────────{ counter: 0 }─
utop # (1,3.14,'c') ;;
- : int * float * char = (1, 3.14, 'c')
─( 13:45:58 )─< command 9 >──────────────────────────────────────{ counter: 0 }─
utop # (3, ["asdf";"qwer";"qwer"]);;
- : int * string list = (3, ["asdf"; "qwer"; "qwer"])
─( 13:46:26 )─< command 10 >─────────────────────────────────────{ counter: 0 }─
utop # (1,2,3.14) ;;
- : int * int * float = (1, 2, 3.14)
─( 13:46:40 )─< command 11 >─────────────────────────────────────{ counter: 0 }─
utop # (1,("c",4.5)) ;;
- : int * (string * float) = (1, ("c", 4.5))
─( 13:47:10 )─< command 12 >─────────────────────────────────────{ counter: 0 }─
utop # (1,"c",4.5) ;;
- : int * string * float = (1, "c", 4.5)
─( 13:47:38 )─< command 13 >─────────────────────────────────────{ counter: 0 }─
utop # let (x,y) = (1,2) ;;
val x : int = 1
val y : int = 2
─( 13:47:52 )─< command 14 >─────────────────────────────────────{ counter: 0 }─
utop # let point = (3.1, 5.7) ;;
val point : float * float = (3.1, 5.7)
─( 13:48:42 )─< command 15 >─────────────────────────────────────{ counter: 0 }─
utop # let (x,y) = point ;;
val x : float = 3.1
val y : float = 5.7
─( 13:49:02 )─< command 16 >─────────────────────────────────────{ counter: 0 }─
utop # () ;;
- : unit = ()
─( 13:49:10 )─< command 17 >─────────────────────────────────────{ counter: 0 }─
utop # (1,2) ;;
- : int * int = (1, 2)
─( 13:49:26 )─< command 18 >─────────────────────────────────────{ counter: 0 }─
utop # (1) ;;
- : int = 1
─( 13:49:50 )─< command 19 >─────────────────────────────────────{ counter: 0 }─
utop # #use "simple.ml";;
val power : int -> float -> float = <fun>
val cube : float -> float = <fun>
val gcd : int -> int -> int = <fun>
val sum : int list -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val is_even : int -> bool = <fun>
```

```
File "simple.ml", line 47, characters 6-10:
Warning 26: unused variable even.
val even2ways_better : int list -> bool = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val length : 'a list -> int = <fun>
File "simple.ml", line 81, characters 2-32:
Warning 8: this pattern-matching is not exhaustive.
Here is an example of a case that is not matched:
[]
val head : 'a list -> 'a = <fun>
val first : 'a * 'b * 'c -> 'a = <fun>
-( 13:50:50 )-< command 20 >-------------------------------------{ counter: 0 }-
utop # first (1,2,3) ;;
- : int = 1
-( 13:53:34 )-< command 21 >-------------------------------------{ counter: 0 }-
utop # List.rev ;;
- : 'a list -> 'a list = <fun>
-( 13:53:40 )-< command 22 >-------------------------------------{ counter: 0 }-
utop # #use "simple.ml";;
val power : int -> float -> float = <fun>
val cube : float -> float = <fun>
val gcd : int -> int -> int = <fun>
val sum : int list -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val is_even : int -> bool = <fun>
File "simple.ml", line 47, characters 6-10:
Warning 26: unused variable even.
val even2ways_better : int list -> bool = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val length : 'a list -> int = <fun>
File "simple.ml", line 81, characters 2-32:
Warning 8: this pattern-matching is not exhaustive.
Here is an example of a case that is not matched:
[]
val head : 'a list -> 'a = <fun>
val first : 'a * 'b * 'c -> 'a = <fun>
val first' : 'a * 'b * 'c -> 'a = <fun>
val first'' : 'a * 'b * 'c -> 'a = <fun>
-( 13:54:57 )-< command 23 >-------------------------------------{ counter: 0 }-
utop #
```

Arg	Array	ArrayLabels	Assert_failure	Bigarray	Buffer	Bytes	BytesLabels	Callbac