```
Last login: Fri Jan 26 14:03:32 on ttys004
carbon:$ utop
──────────┬──────────────────────────────────────────────────────────────┬──
          │ Welcome to utop version 2.0.2 (using OCaml version 4.06.0)! │
          └──────────────────────────────────────────────────────────────┘

Type #utop_help for help about using utop.

─( 15:39:39 )─< command 0 >──────────────────────────────────────{ counte
r: 0 }─
utop # #use "simple.ml";;
val inc_v1 : int -> int = <fun>
val inc_v2 : int -> int = <fun>
val square : int -> int = <fun>
val cube : int -> int = <fun>
val add : int -> int -> int = <fun>
val inc_v3 : int -> int = <fun>
val add3 : int -> int -> int -> int = <fun>
val greater : 'a -> 'a -> 'a = <fun>
val circle_area : float -> float = <fun>
val power : int -> float -> float = <fun>
val power_v2 : int -> float -> float = <fun>
val cube : float -> float = <fun>
val foo : float = 13.824
val bar : float = 13.824
val gcd : int -> int -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val even : int -> bool = <fun>
val sum : int list -> int = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val is_empty' : 'a list -> bool = <fun>
val not_empty : 'a list -> bool = <fun>
val not_empty'' : 'a list -> bool = <fun>
val sum : int list -> int = <fun>
val length : 'a list -> int = <fun>
File "simple.ml", line 119, 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>
─( 15:39:39 )─< command 1 >──────────────────────────────────────{ counte
r: 0 }─
utop # head (1::2::[]) ;;
- : int = 1
─( 15:39:42 )─< command 2 >──────────────────────────────────────{ counte
```

```
r: 0 }–
utop # head [ ] ;;
Exception: Match_failure ("simple.ml", 119, 2).
–( 15:39:51 )–< command 3 >–––––––––––––––––––––––––––––––––––––––––––––––––––{ counte
r: 0 }–
utop # #use "simple.ml";;
val inc_v1 : int -> int = <fun>
val inc_v2 : int -> int = <fun>
val square : int -> int = <fun>
val cube : int -> int = <fun>
val add : int -> int -> int = <fun>
val inc_v3 : int -> int = <fun>
val add3 : int -> int -> int -> int = <fun>
val greater : 'a -> 'a -> 'a = <fun>
val circle_area : float -> float = <fun>
val power : int -> float -> float = <fun>
val power_v2 : int -> float -> float = <fun>
val cube : float -> float = <fun>
val foo : float = 13.824
val bar : float = 13.824
val gcd : int -> int -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val even : int -> bool = <fun>
val sum : int list -> int = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val is_empty' : 'a list -> bool = <fun>
val not_empty : 'a list -> bool = <fun>
val not_empty'' : 'a list -> bool = <fun>
val sum : int list -> int = <fun>
val length : 'a list -> int = <fun>
val head : 'a list -> 'a = <fun>
–( 15:40:01 )–< command 4 >–––––––––––––––––––––––––––––––––––––––––––––––––––{ counte
r: 0 }–
utop # head [ ] ;;
Exception: Failure "oh nuts!".
–( 15:42:18 )–< command 5 >–––––––––––––––––––––––––––––––––––––––––––––––––––{ counte
r: 0 }–
utop # head [ ] ;;
Exception: Failure "oh nuts!".
–( 15:42:25 )–< command 6 >–––––––––––––––––––––––––––––––––––––––––––––––––––{ counte
r: 0 }–
utop # #use "simple.ml";;
File "simple.ml", line 112, characters 0-3:
Error: Syntax error
–( 15:58:35 )–< command 7 >–––––––––––––––––––––––––––––––––––––––––––––––––––{ counte
r: 0 }–
utop # #use "simple.ml";;
val inc_v1 : int -> int = <fun>
```

```
val inc_v2 : int -> int = <fun>
val square : int -> int = <fun>
val cube : int -> int = <fun>
val add : int -> int -> int = <fun>
val inc_v3 : int -> int = <fun>
val add3 : int -> int -> int -> int = <fun>
val greater : 'a -> 'a -> 'a = <fun>
val circle_area : float -> float = <fun>
val power : int -> float -> float = <fun>
val power_v2 : int -> float -> float = <fun>
val cube : float -> float = <fun>
val foo : float = 13.824
val bar : float = 13.824
val gcd : int -> int -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val even : int -> bool = <fun>
val sum : int list -> int = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val is_empty' : 'a list -> bool = <fun>
val not_empty : 'a list -> bool = <fun>
val not_empty'' : 'a list -> bool = <fun>
val sum : int list -> int = <fun>
val length : 'a list -> int = <fun>
val head : 'a list -> 'a = <fun>
File "simple.ml", line 125, characters 19-25:
Error: Unbound value sum_v2
-( 15:58:37 )-< command 8 >-------------------------------------------{ counte
r: 0 }-
utop # (1, 3.4) ;;
- : int * float = (1, 3.4)
-( 15:58:48 )-< command 9 >-------------------------------------------{ counte
r: 0 -( 16:00:25 )-< command 9 >-------------------------------------------{ cou
-( 16:00:25 )-< command 9 >-----------------------------------{ counter: 0 }-
utop # (1, 3.4) ;;
- : int * float = (1, 3.4)
-( 16:00:25 )-< command 10 >----------------------------------{ counter: 0 }-
utop # (1, 'c', 3.1415) ;;
- : int * char * float = (1, 'c', 3.1415)
-( 16:00:46 )-< command 11 >----------------------------------{ counter: 0 }-
utop # let t3 = (1, 'c', 3.1415) ;;
val t3 : int * char * float = (1, 'c', 3.1415)
-( 16:01:05 )-< command 12 >----------------------------------{ counter: 0 }-
utop # match t3 with (x,y,z) -> x ;;
- : int = 1
-( 16:01:31 )-< command 13 >----------------------------------{ counter: 0 }-
utop # let (x,y,z) = t3 ;;
val x : int = 1
val y : char = 'c'
```

```
val z : float = 3.1415
-( 16:01:49 )-< command 14 >-------------------------------------{ counter: 0 }-
utop # let (x,_,_) = t3 ;;
val x : int = 1
-( 16:01:57 )-< command 15 >-------------------------------------{ counter: 0 }-
utop # #use "simple.ml";;
val inc_v1 : int -> int = <fun>
val inc_v2 : int -> int = <fun>
val square : int -> int = <fun>
val cube : int -> int = <fun>
val add : int -> int -> int = <fun>
val inc_v3 : int -> int = <fun>
val add3 : int -> int -> int -> int = <fun>
val greater : 'a -> 'a -> 'a = <fun>
val circle_area : float -> float = <fun>
val power : int -> float -> float = <fun>
val power_v2 : int -> float -> float = <fun>
val cube : float -> float = <fun>
val foo : float = 13.824
val bar : float = 13.824
val gcd : int -> int -> int = <fun>
val all : bool list -> bool = <fun>
val even2ways : int list -> bool = <fun>
val even : int -> bool = <fun>
val sum : int list -> int = <fun>
val string_concat : string -> string list -> string = <fun>
val is_empty : 'a list -> bool = <fun>
val is_empty' : 'a list -> bool = <fun>
val not_empty : 'a list -> bool = <fun>
val not_empty'' : 'a list -> bool = <fun>
val sum : int list -> int = <fun>
val length : 'a list -> int = <fun>
val head : 'a list -> 'a = <fun>
val sum_v2 : int list -> int = <fun>
val sum : int list -> int = <fun>
val first : 'a * 'b * 'c -> 'a = <fun>
-( 16:02:29 )-< command 16 >-------------------------------------{ counter: 0 }-
utop #
```

Arg	Array	ArrayLabels	Assert_failure	Bigarray	Buffer	Bytes	BytesLabels	