```
Last login: Mon Feb  5 13:23:55 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.

─( 15:46:55 )─< command 0 >────────────────────────────────────{ counter: 0 }─
utop # List.map ;;
- : ('a -> 'b) -> 'a list -> 'b list = <fun>
─( 15:46:55 )─< command 1 >────────────────────────────────────{ counter: 0 }─
utop # List.map (fun x -> x + 1) [1;2;3] ;;
- : int list = [2; 3; 4]
─( 15:46:59 )─< command 2 >────────────────────────────────────{ counter: 0 }─
utop # List.map Char.code ['e'; 'T' ] ;;
- : int list = [101; 84]
─( 15:47:15 )─< command 3 >────────────────────────────────────{ counter: 0 }─
utop # List.filter ;;
- : ('a -> bool) -> 'a list -> 'a list = <fun>
─( 15:47:30 )─< command 4 >────────────────────────────────────{ counter: 0 }─
utop # 'A' < 'D' ;;
- : bool = true
─( 15:53:35 )─< command 5 >────────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
─( 16:01:25 )─< command 6 >────────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
val fold : ('a -> 'b -> 'b) -> 'b -> 'a list -> 'b = <fun>
─( 16:02:37 )─< command 7 >────────────────────────────────────{ counter: 0 }─
utop # fold (+) 0 [1;2;3;4] ;;
- : int = 10
─( 16:10:12 )─< command 8 >────────────────────────────────────{ counter: 0 }─
utop # fold ( * ) 1 [1;2;3;4] ;;
- : int = 24
─( 16:10:21 )─< command 9 >────────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
val foldr : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
─( 16:10:41 )─< command 10 >───────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
val foldr : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
val length : 'a list -> int = <fun>
─( 16:14:18 )─< command 11 >───────────────────────────────────{ counter: 0 }─
utop # length [1;2;3;4;5]  ;;
- : int = 5
─( 16:21:04 )─< command 12 >───────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
val foldr : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
val foldl' : ('a -> 'a list -> 'a list) -> 'a list -> 'a list -> 'a list =
  <fun>
val length : 'a list -> int = <fun>
─( 16:21:15 )─< command 13 >───────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
val foldr : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
val foldl' : ('a -> 'b -> 'a) -> 'b list -> 'a -> 'a = <fun>
val length : 'a list -> int = <fun>
─( 16:31:04 )─< command 14 >───────────────────────────────────{ counter: 0 }─
utop # #use "higher_order.ml";;
val implode : char list -> string = <fun>
val explode : string -> char list = <fun>
val map : ('a -> 'b) -> 'a list -> 'b list = <fun>
val filter : ('a -> bool) -> 'a list -> 'a list = <fun>
val removeABCD : char list -> char list = <fun>
val foldr : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val foldl : ('a -> 'b -> 'a) -> 'a -> 'b list -> 'a = <fun>
val foldl' : ('a -> 'b -> 'b) -> 'a list -> 'b -> 'b = <fun>
val length : 'a list -> int = <fun>
─( 16:31:31 )─< command 15 >───────────────────────────────────{ counter: 0 }─
utop #
 │Arg│Array│ArrayLabels│Assert_failure│Bigarray│Buffer│Bytes│BytesLabels│Callb│
```