```
~ (501\)% sml
Standard ML of New Jersey v110.77 [built: Sat Aug 23 11:18:31 2014]
- 0 ;
val it = 0 : int
- fun f x = x + 1;
val f = fn : int -> int
- f 3 ;
val it = 4 : int
- f 4.3 ;
stdIn:4.1-4.6 Error: operator and operand don't agree [tycon mismatch]
  operator domain: int
  operand:         real
  in expression:
    f 4.3
-  (* Lists are an important data structure *)
- [1, 2, 3, 4];
val it = [1,2,3,4] : int list
- (* all the elements of a list must be of the same type *)
- [[1,2], [3,4], [5,6]] ;
val it = [[1,2],[3,4],[5,6]] : int list list
- [1, 2, [3, 4]] ;
stdIn:8.1-8.15 Error: operator and operand don't agree [literal]
  operator domain: int * int list
  operand:         int * int list list
  in expression:
    2 :: (3 :: 4 :: nil) :: nil
- (* CONS is written as :: *)
- 3 :: [4, 5, 6];
val it = [3,4,5,6] : int list
- (* CAR is written as hd *)
- hd [3,4,5];
val it = 3 : int
- (* CDR is written as tl *)
- tl [3,4,5];
val it = [4,5] : int list
- (* ML provides a nice pattern matching syntax *)
- fun fac 0 = 1
=  |  fac x = x * fac (x - 1) ;
val fac = fn : int -> int
- (* pattern matching for lists! *)
- fun append [] L = L
=  |  append (x::xs) L = x :: append xs L ;
val append = fn : 'a list -> 'a list -> 'a list
- append [1,2,3] [4, 5, 6];
val it = [1,2,3,4,5,6] : int list
- fun add [] = 0
=  |  add (y::ys) = y + add ys
= ;
val add = fn : int list -> int
- [1,2,3] :: [4,5,6];
stdIn:13.1-13.19 Error: operator and operand don't agree [literal]
  operator domain: int list * int list list
  operand:         int list * int list
  in expression:
    (1 :: 2 :: 3 :: nil) :: 4 :: 5 :: 6 :: nil
-
- (* functions are curried *)
- fun f x y = x+y;
val f = fn : int -> int -> int
- fun g (x:real) y = x+y;
```

```
val g = fn : real -> real -> real
- (* length is polymorphic *)
- fun length [] = 0
=  |  length (x::xs) = 1 + length xs ;
val length = fn : 'a list -> int
- length;
val it = fn : 'a list -> int
- length [1,2,3] + length ["hello", "goodbye"] ;
val it = 5 : int
-
```