

Objective Caml version 3.06

```
(* Listen als rein funktionale Objekte:
   Da man 2 (sehr unterschiedliche) Konstruktoren braucht,
   naemlich einen fuer die leere Liste und einen fuer nichtleere Listen,
   definiert man sich am besten 2 Klassen, die Objekte vom gleichen Typ liefern. *)

# type 'a list_object =
< hd: 'a;
  tl: 'a list_object;
  is_empty: bool >;
type 'a list_object = < hd : 'a; is_empty : bool; tl : 'a list_object >

# exception Empty;;
exception Empty

# class ['a] empty_list =
object (self: 'b)
  method hd: 'a = raise Empty
  method tl: 'b = raise Empty
  method is_empty = true
end;;
class ['a] empty_list :
  object ('b) method hd : 'a method is_empty : bool method tl : 'b end

# let (l: int list_object) = new empty_list;;
val l : int list_object = <obj>

# class ['a] cons (x: 'a) (l: 'a list_object) =
object
  method hd = x
  method tl = l
  method is_empty = false
end;;
class ['a] cons :
  'a ->
  'a list_object ->
  object method hd : 'a method is_empty : bool method tl : 'a list_object end

# let (l1: int list_object) = new cons 2 l;;
val l1 : int list_object = <obj>

# l1#hd;;
- : int = 2

# l1#tl;;
- : int list_object = <obj>

# l1#tl#hd;;
Exception: Empty.

# l1#is_empty;;
- : bool = false

# l1#tl#is_empty;;
- : bool = true

#
```