

Ocaml examples

Lecturer:

Disclaimer: *These notes may be distributed outside this class only with the permission of the Instructor.*

1.1 OCaml code examples

1.1.1 Calculate the average

Listing 1: dictionary

```
1 (* calculate the average of a list of integers *)
2 let grades = [80;90;70;60];;
3 let rec fold f l acc =
4   match l with
5   []->acc
6   |h::t-> f h (fold f t acc)
7   ;;
8 let sum l = fold (fun x y ->x+y) l 0;;
9 let s = sum grades;;
10 print_int s;;
11 print_string "\n";;
12
13 let avg l =
14     let s = sum l in
15     let rec length l =
16         match l with
17         []->0
18         |h::t->1 + length t
19     in s/(length l)
20 ;;
21 let v = avg grades;;
22 print_int v;;
23 print_string "\n";;
```

1.1.2 Insertion Sort

Listing 2: insertion sort

```
1 let rec sort = function
2   | [] -> []
3   | x :: l -> insert x (sort l)
4   and insert elem = function
5   | [] -> [elem]
```

```

6 | x :: l -> if elem < x then elem :: x :: l
7 |           else x :: insert elem l;;

```

Listing 3: insertion sort 2

```

1 let rec sort lst =
2     match lst with
3     | [] -> []
4     | x :: l -> insert x (sort l);;
5 let rec insert elem lst = match lst with
6 | [] -> [elem]
7 | x :: l -> if elem < x then elem :: x :: l
8 |           else x :: insert elem l;;

```

1.1.3 List of functions

Listing 4: apply list of functions to a list

```

1 \label{mapmap}
2 (* in this example, we will apply a list of functions to a list
3    and return the result as a list of list *)
4 let list= [2;3;6;9];;
5 let double x = x *2;;
6 let halve x = x/2;;
7 let self x = x;;
8 let square x = x * x;;
9 let flist=[double; halve; self; square];;
10 let rec map_map f1 l1 =
11     match f1 with
12     | [] -> []
13     | h1::t1->
14         (let rec map f l=
15             match l with
16             | [] -> []
17             | h::t->f h::map f t
18             in map h1 l1
19             )::map_map t1 l1
20 ;;
21
22 map_map flist list;;

```

Listing 5: Result

```

1 Result:
2 [[4; 6; 12; 18]; [1; 1; 3; 4]; [2; 3; 6; 9]; [4; 9; 36; 81]]

```

In the example in Listing ??, if we add following two functions to the function list.

Listing 6: more functions

```

1 let is_even x = if x mod 2 = 0 then true else false;;

```

```

2 let to_str x = string_of_int x;;
3 let flist=[double;halve;self;square;is_eve;to_str];;

```

Does it work? Why?

1.1.4 Explode: String to list

Listing 7: Result

```

1 let explode s =
2   let rec expl i l =
3     if i < 0 then l else
4     expl (i - 1) (s.[i] :: l) in
5   expl (String.length s - 1) [];;

```

1.1.5 Implode: Character list to string

Listing 8: Result

```

1 let implode l =
2   let result = String.create (List.length l) in
3   let rec imp i = function
4     | [] -> result
5     | c :: l -> result.[i] <- c; imp (i + 1) l in
6   imp 0 l;;

```

1.1.6 Read a File

Listing 9: Read a file

```

1 (* Read lines from a text file. Use regular expression to
2  replace the first letter of each line with A *)
3
4 let change name =
5   Str.replace_first (Str.regexp "[A-Z]") "ABC" name
6 ;;
7
8 let read_file file_name =
9   let in_file = open_in file_name in
10  try
11    while true do
12      let line = input_line in_file in
13      print_endline (change line)
14    done
15  with End_of_file ->
16    close_in in_file
17
18 ;;

```

```
19 |  
20 |  
21 | read_file "names.txt";;
```

References

[OCaml from the very beginning] JOHN WHITTINGTON *Coherent Press*