



*„Programs should be written for humans to read,
and only incidentally for machines to execute.“
– Hal Abelson & Jay Sussman*

Bevor Sie mit den Aufgaben anfangen, schauen Sie sich bitte die OCaml-Richtlinien an:
<https://ocaml.org/learn/tutorials/guidelines.html>

Aufgabe 0.

- OCaml installieren: <https://ocaml.org/docs/install.html> (Debian/Ubuntu: `(sudo) apt-get install ocaml`)
- Camlp5 installieren: <http://camlp5.gforge.inria.fr/> (Debian/Ubuntu: `(sudo) apt-get install camlp5`)
- Einen Editor für Ihre OCaml Dateien wählen und installieren. Das könnte ein einfacher Text-Editor sein, wie z.B. „Leafpad“ oder „Notebook“, oder etwas, das OCaml evaluieren und kompilieren kann, wie z.B. „Emacs“ + „tuareg-mode“. (Debian/Ubuntu: `(sudo) apt-get emacs tuareg-mode`)

Aufgabe 1.

Schreiben Sie für jede der folgenden Funktionen eine OCaml-Aussage, die das Ergebnis `- : int = 9` liefert:

```
let f1 = fun x -> x * x
and f2 = fun x y -> x / y
and f3 = fun x -> (fun y -> x / y)
and f4 x = x 3
and f5 x = if x=2
            then (fun () -> x)
            else (fun () -> x * 3);;
```

Aufgabe 2.

Schreiben Sie für jede der folgenden Beispiele mindestens eine, nicht-triviale Funktion, die das gleiche Ergebnis liefert:

- `append_to_end_int_list 0 [1; 2; 3; 4] = [1; 2; 3; 4; 0]`
- `append_to_end_string 's' "hacker" = "hackers"`
- `order "theorem prover" = "eeehmooprtrtv"` (Leerzeichen entfernen)
- `reverse_int [1; 2; 3] = [3; 2; 1]`
- `reverse_string "abc" = "cba"`
- `palindrome_p "never odd or even" = true` (Leerzeichen ignorieren)

Sie können die Ressourcen unten, das Internet, und folgende Funktionen dafür benutzen:

```
let explode string =
  let rec expl a b =
    if a < 0 then b
    else expl (a - 1) (string.[a] :: b)
  in
  expl (String.length string - 1) [];;

let implode list =
  let result = String.create (List.length list)
  in
  let rec impl n = function
    | [] -> result
    | a :: list -> result.[n] <- a;
                    impl (n + 1) list
  in
  impl 0 list;;
```

Ressourcen z.B.: <http://caml.inria.fr/pub/docs/manual-ocaml/>