

課題 6.1

1.

```
let deriv f x d = (f(x +. d) -. f(x)) /. d;;
deriv (fun x -> x *. x) 2.0 0.1;;
deriv (fun x -> x *. x *. x) 2.0 0.1;;
```

実行結果

```
val deriv : (float -> float) -> float -> float -> float = <fun>
- : float = 4.10000000000000142
- : float = 12.6100000000000101
```

2.

```
let rec applyn f n x =
  if n = 0 then x
  else applyn f (n - 1) (f x);;
applyn (fun x -> x + 2) 4 3;;
applyn (fun x -> x * x) 3 2;;
```

実行結果

```
val applyn : ('a -> 'a) -> int -> 'a -> 'a = <fun>
- : int = 11
- : int = 256
```

課題 6.2

1.

```
let rec split f l =
  match l with
  [] -> ([], [])
  | x::xs ->
    let (l1, l2) = split f xs in
    if f(x) then (x::l1, l2)
    else (l1, x::l2);;
split (fun x -> x mod 2 = 0) [3; 1; 2];;
split (fun x -> x mod 2 = 0) [];;
```

実行結果

```
val split : ('a -> bool) -> 'a list -> 'a list * 'a list = <fun>
- : int list * int list = ([2], [3; 1])
- : int list * int list = ([], [])
```

2.

```
let rec apply_list lf v =
  match lf with
  [] -> []
  | f::fs -> f v :: apply_list fs v;;
apply_list [(fun x -> x + x); (fun x -> x * x); (fun x -> -x)] 5;;
```

実行結果

```
val apply_list : ('a -> 'b) list -> 'a -> 'b list = <fun>
- : int list = [10; 25; -5]
```

課題 6.3

1.

```
let exists f l = List.fold_right (fun x y -> f x || y) l false;;
exists (fun x -> x > 1) [0; 3];;
```

実行結果

```
val exists : ('a -> bool) -> 'a list -> bool = <fun>
- : bool = true
```

2.

```
let flatten l = List.fold_right (fun x y -> x@y) l [];;
flatten [[1; 2]; []; [3]];;
```

実行結果

```
val flatten : 'a list list -> 'a list = <fun>
- : int list = [1; 2; 3]
```