

Definitions locales \rightarrow avec des variables qui ne sont utilisables que localement (Pas en dehors de cette définition)

Définition locale simple : `let ident = expression1 in expression2`

```
# let a = 1 + 2 in a * 3 ;;
- : int = 9
```

Définitions locales emboîtées et multiples :

```
let ident1 = expr1 and ... and identi = expri
in let identi+1 = expri+1 and ... and identj = exprj
...
in expressionn
```

```
# let a = 1 and b = 3 in
  let x = a + b and y = a - b in
    x * y ;;
- : int = -8
```

Une expression contenant une ou plusieurs définitions locales est une expression !

3 Les fonctions

3.1 Fonctions à un seul paramètre

Définir une fonction : `let f x = expression` \rightarrow Paramètre formel

```
# let succ x = x + 1 ;;
val succ : int -> int = <fun>
```

Application de la fonction f à la valeur x : `f x`

```
# succ 3 ;;
- : int = 4
```

⚠ L'application de fonction est prioritaire sur tout autre opérateur.

`f x + y` \equiv `(f x) + y` `f x + g y` \equiv `(f x) + (g y)`

```
# succ 3*2;;
- : int = 8
```

```
# succ (3*2) ;;
- : int = 7
```

⚠ Si le paramètre n'est pas une expression atomique, il faut donc le parenthéser.

\rightarrow Paramètre effectif

```
# succ -1 ;;
Characters 0-4: Error: This expression has type int -> int
but an expression was expected of type int
```

```
# succ (-1) ;;
- : int = 0
```

3.2 Fonctions à plusieurs paramètres

Définition : `let f x y = expression`

```
# let average a b = float_of_int(a + b) /. 2.;;
val average : int -> int -> float = <fun>
```

Application à x et y : `f x y` \equiv `(f x) y`

```
# average 2 3 ;; # same as (average 2) 3
- : float = 2.5
```

L'application de fonction est "associative à gauche"

```
# average (2 3) ;;
Error: This expression has type int
This is not a function; it cannot be applied.
```

Donc : `f (g x) \neq f g x`

```
# succ succ 2 ;;
Error: This function has type int -> int
It is applied to too many arguments; maybe you forgot a ';;'.4
```

```
# succ (succ 2) ;;
- : int = 4
```

Attention : `let f (x, y) = expression` est une fonction à 1 seul paramètre (le couple (x, y)).

3. L'indication des caractères concernés par l'erreur sera omise dans les exemples suivants.
4. Ceci (le ';;') sera expliqué plus tard...