

# Programmation applicative – L2

## TD 3 Correction : Variables, liaison et portée

G.Artignan {artignan@lirmm.fr}      M.Lafourcade {lafourcade@lirmm.fr}  
S.Daudé {sylvain.daude@univ-montp2.fr}      A.Chateau {chateau@lirmm.fr}  
B.Paiva Lima da Silva {bplsilva@gmail.com}      C.Dony {dony@lirmm.fr}

### Exercice 1 *Valeur des expressions let*

```
> (let ((a 2) (b (* 2 2)) (c 10))
    (* c (- a b)))
⇒ -20

> (let ((x 5))
    (* x x)
    (let ((x 10))
        (* x x)))
⇒ 100

> (let ((a 2)
        (b (let ((a 3) (b 4))
            (- a b)(+ a b))))
    (* a b))
⇒ 14
```

### Exercice 2 *Ré-écriture des expressions let avec lambda*

```
((lambda (a b c) (* c (- a b))) 2 (* 2 2) 10)

((lambda (x) (* x x)) ((lambda (x) (* x x)) 10)) 5)
```

### Exercice 3 *Portée et liaison*

```
> (define a 10)
⇒ rien

> (let ((x a)) (* 2 x))
⇒ 20

> (let ((a a)) (* 2 a))
⇒ 20

> (let ((a (* 2 a))) (* 2 a))
⇒ 40

> (let ((a (* 2 a)) (b 5) (c (+ 1 a)))
    (+ a b c))
⇒ 36
```

### Exercice 4 *Let imbriqués : affichage*

```
a : b
b : 3
a : 4
b : b
```

### Exercice 5 *Valeur des expressions let\**

```
(define a 10)
⇒ rien
```

```
(let ((a 2)(b a))
  (* a b))
⇒ 20
```

```
(let* ((a 2)(b a))
  (* a b))
⇒ 4
```

```
(let* ((a (* 2 a)) (b 5) (c (+ 1 a)))
  (+ a b c))
⇒ 46
```

```
(letrec ((a (* 2 a)) (b 5) (c (+ 1 a)))
  (+ a b c))
⇒ erreur
```

```
(let ((x 5))
  (let* ((y (+ x 10))
        (z (* x y)))
    (+ x y z)))
⇒ 95
```

### Exercice 6 *Valeur des expressions*

```
(define a 10)
⇒ rien
```

```
(define (foo x) (* a x))
⇒ rien
```

```
(let ((a 0)) (foo 5))
⇒ 50
```

```
(define a 100)
⇒ rien
```

```
(let ((a 0)) (foo 5))
⇒ 500
```

```
((lambda (a)
  (let ((a 1)
        (foo (lambda (x) (+ a x))))
    (foo a))) 5)
⇒ 6
```

```
((lambda (a)
  (let* ((a 1)
         (foo (lambda (x) (+ a x))))
    (foo a))) 5)
⇒ 2
```