

Auteur du rapport :  
**Soulié Maxime**

## **TP 1d**

programmation fonctionnelle, OCaml

## Table des matières

1. Premiers pas.....	4
1.1. Expressions fonctionnelles simples.....	4
1.2. expressions bizarres.....	4
3. premières fonctions.....	6

**Index des figures**

**Index des tableaux**

# 1. Fonctions

## 1.1. exception « failwith »

### 1.1.1.

```
let rac x = if x >= 0 then  
sqrt(float_of_int x)  
else failwith "x < 0";;
```

## 1.2. définitions locales dans des fonctions

### 1.2.1. Un jeu

```
let deviner x =  
let rand = Random.int(11) in  
if x < 0 then  
failwith "x < 0"  
else x = rand;;
```

### 1.2.2. un calcul

```
let f x =  
let c = cos(x) in  
exp((c**2.)+.2.)-.10.;;
```

## 1.3. un problème de représentation

### 1.3.1.

```
let nombre_sol a b c =  
let delta = int_of_float(b**2.)-4*a*c in  
if delta > 0 then 2  
else if delta < 0 then 1  
else 0;;
```

### 1.3.2.

```
let tarif masse reco =  
match reco with  
| 1 when masse <= 20. -> 0.5  
| 1 when masse <= 50. -> 0.78  
| 1 when masse <= 100. -> 1.18  
| 2 when masse <= 20. -> 3.33  
| 2 when masse <= 50. -> 3.61  
| 2 when masse <= 100. -> 4.02  
| _ -> failwith "error";;
```

## 2. stratégies d'évaluation

### 2.1.

La fonction d1 affiche le texte "on est dans d1 !", saute une ligne et affiche le résultat de  $x*2$ .

Son type est `int -> int = <fun>` car  $x*2$  est de type `int`, et `print_string` n'est pas renvoyé, juste affiché.

### 2.2.

d renvoie true si :

- b et  $(x*2 < 10)$

false sinon.

Son type est `bool -> int -> bool = <fun>` car b doit être un `bool` à cause de l'opérateur `&&`, a est `int` car paramètre de d1. Le résultat est `bool` car renvoie le résultat de `b && ((d1 x) <= 10)`.

- d true 4;;
  - true car `b=true` et  $(4*2=8) \leq 10$
- d true 6;;
  - false car  $(6*2=12) > 10$
- d false 4;;
  - false car `b=false`

### 2.3.

retourne y. son type est polymorphe car aucune operation.

Je m'attends :

On est dans d1!

- : `int = 3`