

Exceptions OCaml

Ce sont des erreurs, lorsqu'une exception est lancée, cela interrompt tout le programme.

- `failwith` lance une exception
- `Not_found`, `Invalid_argument`, `Failure...` sont des exceptions

On peut définir ses propres exceptions :

```
1 exception Nadine;;
```

ocaml

Ce sont des constructeurs !

On peut lancer une exception avec `raise` :

```
1 raise Nadine;;
```

ocaml

On peut donner des arguments à nos exceptions :

```
1 exception Nadine of int;;  
2  
3 raise (Nadine 12);;
```

ocaml

`Failure` et `Invalid_argument` prennent en argument une *string*.

En fait, la fonction `failwith` est codée de cette manière :

```
1 let failwith S = raise (Failure S);;
```

ocaml

On peut rattraper les exceptions :

```
1 try:  
2   (* instructions renvoyant du 'a *)  
3   foo ();  
4   ....  
5 with  
6   (* Ce pattern matching renvoie également du 'a *)  
7   | Failure s -> bar ()  
8   | Nadine s -> nad ()
```

ocaml