Conditions (suite)

Les traitants sont établis avec handler–bind ou handler–case

Pour signaler, on utilise signal, warn, error, cerror, assert, etc

```
* (define-condition zero-found ()
    ())

ZERO-FOUND
* (defun multiply-leaves (l)
    (cond ((consp l) (reduce #'* (mapcar #'multiply-leaves l)))
          ((zerop l) (signal 'zero-found))
          (t l)))

MULTIPLY-LEAVES
* (multiply-leaves '((1 2 (3 4)) (((5 6))) 7))

5040
* (handler-case (multiply-leaves '((1 2 (3 0)) (((5 6))) "hello"))
    (zero-found () 0))

0
*
```