

# EXCEPTIONS

RAISE / THROW ARE UGLY GOTOS LOST IN TIME OF ASYNC / FUTURE

```
class Weapon {
    Weapon(){
        throw new Exception("NoMoreArrow");
    }
}
class Target {
    Target(){
        throw new Exception("TooMuchFog");
    }
}
class Impacted{}

public class Main {
    Weapon armYouBow = new Weapon();
    Target targetMonster = new Target();
    Impacted hitMonster(Weapon w, Target t) {
        return new Impacted();
    }
    FutureTask<Impacted> attack = new FutureTask(hitMonster(armYouBow, targetMonster));
}
```

# EXCEPTIONS

TRY / CATCH SONT LES RACINES DU MAL

```
class Weapon {  
    Weapon(){  
        throw new Exception("NoMoreArrow");  
    }  
}  
  
public class Main {  
    try {  
        doSomethingThatMayThrow();  
        Weapon armYouBow = new Weapon();  
    } catch (Exception e) {  
  
    }  
}
```

```
let arm_your_bow : weapon = try raise (Failure "NoMoreArrow") with  
Failure s -> let _ = print_endline s in raise (Failure « unarmed")
```