EXCEPTIONS

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

```java
class Weapon {
  Weapon(){
    throw new RuntimeException("NoMoreArrow");
  }
}
class Target {
  Target(){
    throw new RuntimeException("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((Callable) hitMonster(armYouBow, targetMonster));

}
```

EXCEPTIONS

TRY / CATCH SONT LES RACINES DU MAL

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

  }
}
```

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