```
-- notes sur le cours 9


val p1 = ("Alice", "Paris", 25)
val p2 = ("Bob", "Londres", 20)
val p3 = ("Carol", "Londres", 25)
val p4 = ("David", "Paris", 10)

val p = Seq(p1,p2,p3,p4)



case class Personne(nom: String, ville: String, age: Integer)
val p1 = Personne("Alice", "Paris", 25)
val p2 = Personne("Bob", "Londres", 20)
val p3 = Personne("Carol", "Londres", 25)
val p4 = Personne("David", "Paris", 10)

//liste de personnes
val p = Seq(p1,p2,p3,p4)

//liste des ages
p.map(x => x.age)

//somme des ages
p.map(x => x.age).reduce(_+_)


//les personnes
//-------------
val pers0 = sc.textFile("C:/Users/hubert/Documents/spark/pers.txt")

val pers1 = pers0.map(ligne => ligne.split(","))

// ensemble d'objets de type Personne
val pers2 = pers1.map(t => Personne(t(0), t(1), t(2).toInt))
pers2.take(10).foreach(println)


// ensemble de paires (ville,  (nom,age))
val pers3 = pers1.map(t => (t(1), (t(0), t(2).toInt)))
pers3.take(10).foreach(println)



// les musees
//-----------
val m0 = sc.textFile("C:/Users/hubert/Documents/spark/musee.txt")
val m1 = m0.map(ligne =>  ligne.split(","))
val m2 = m1.map(t => (t(1), t(0)))
m2.take(10).foreach(println)



//jointure entre les personnes et les musées
//------------------------------------------
val visite = pers3.join(m2)
visite.take(10).foreach(println)
```

```
// termes
//--------
val t = sc.textFile("C:/Users/hubert/Documents/spark/terms1000.csv")
term.count

t.take(2)


val t1 = t.map(line => line.split("\\|"))
val t2 = t1.map(t => (t(0), t(1)))
t2.take(10).foreach(println)


// longueur de chaque termes
t2.map{ case (n, e) => e.size}.take(10)



// fichier de grande taille
// ------------------------

val big =
sc.textFile("C:/Users/hubert/Documents/spark/terms.csv").coalesce(1)
big.count


val big =
sc.textFile("C:/Users/hubert/Documents/spark/terms.csv").repartition(10).pe
rsist
big.count
```