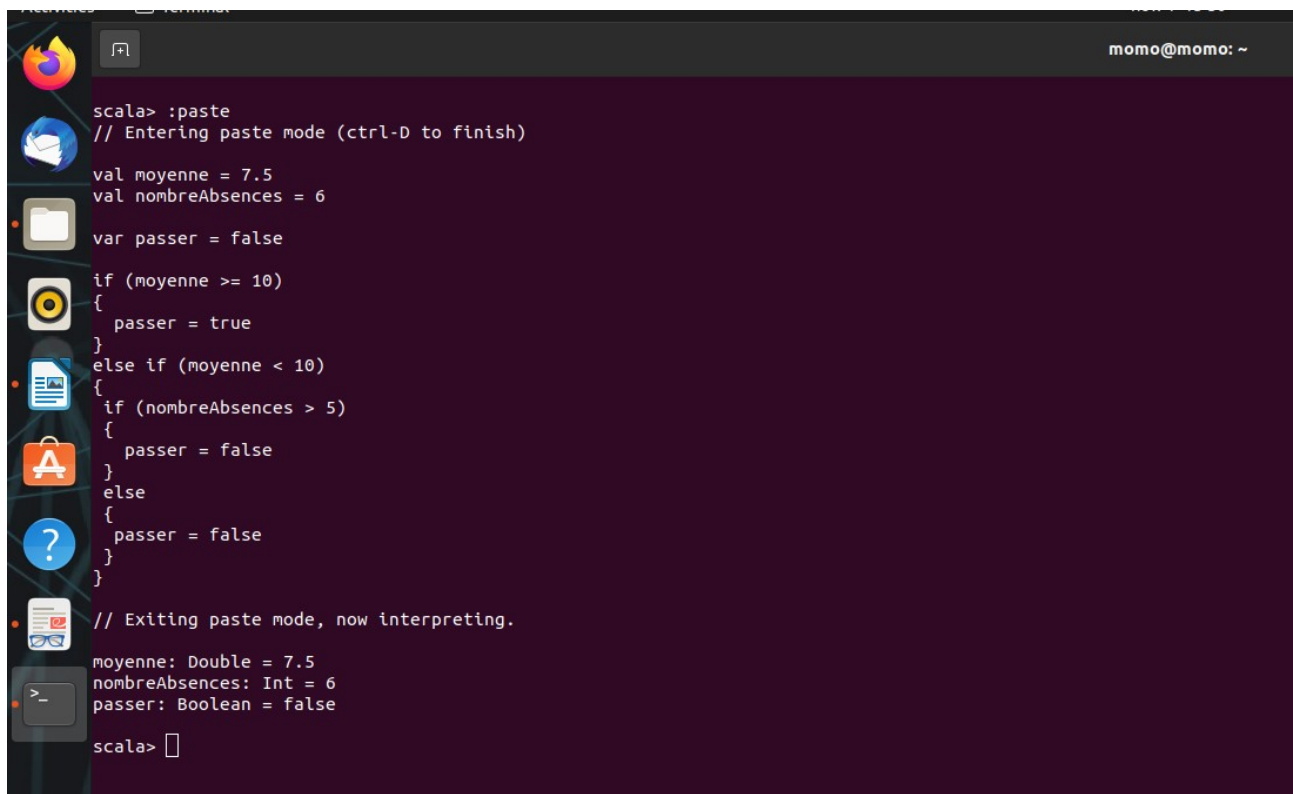


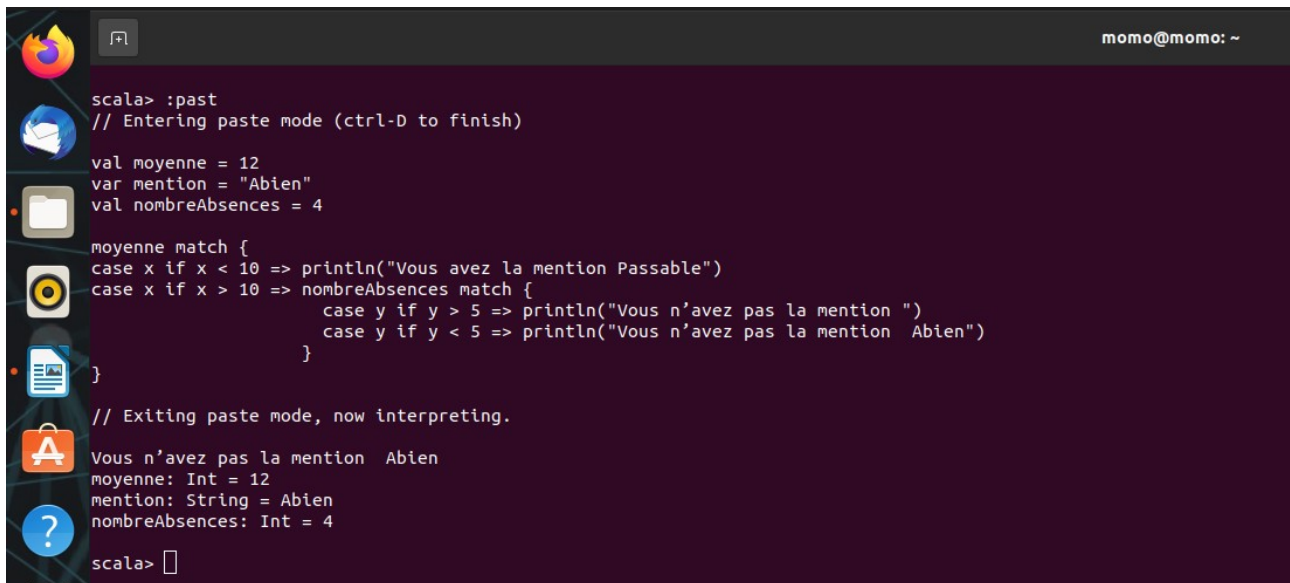
```
momo@momo: ~  
scala> :paste  
// Entering paste mode (ctrl-D to finish)  
  
val moyenne = 7.5  
val nombreAbsences = 6  
  
if (moyenne >= 10)  
{  
  println("Passe en classe supérieure")  
}  
else if (moyenne < 10)  
{  
  if (nombreAbsences > 5)  
  {  
    println("Autoriser à reprendre la classe")  
  }  
  else  
  {  
    println("Autoriser à aller en rattrapage")  
  }  
}  
  
// Exiting paste mode, now interpreting.  
  
Autoriser à reprendre la classe  
moyenne: Double = 7.5  
nombreAbsences: Int = 6  
  
scala> 
```

- Assigner le résultat d'un bloc if/else à une variable



```
momo@momo: ~  
scala> :paste  
// Entering paste mode (ctrl-D to finish)  
  
val moyenne = 7.5  
val nombreAbsences = 6  
  
var passer = false  
  
if (moyenne >= 10)  
{  
  passer = true  
}  
else if (moyenne < 10)  
{  
  if (nombreAbsences > 5)  
  {  
    passer = false  
  }  
  else  
  {  
    passer = false  
  }  
}  
  
// Exiting paste mode, now interpreting.  
  
moyenne: Double = 7.5  
nombreAbsences: Int = 6  
passer: Boolean = false  
  
scala> 
```

- Imbrication de différents cas avec match



```
scala> :past
// Entering paste mode (ctrl-D to finish)

val moyenne = 12
var mention = "Abien"
val nombreAbsences = 4

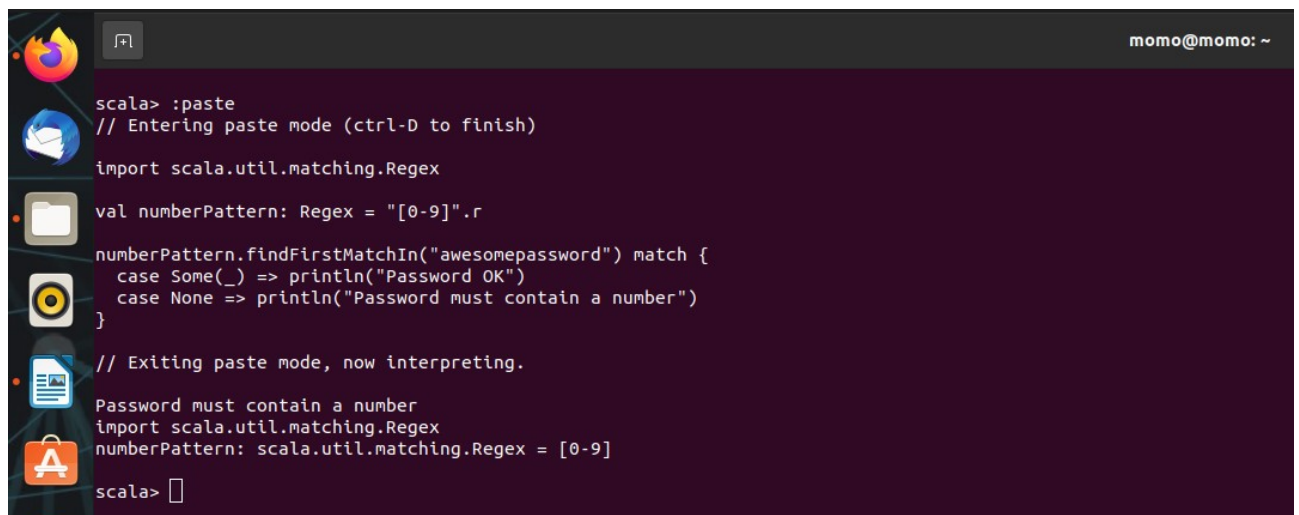
moyenne match {
  case x if x < 10 => println("Vous avez la mention Passable")
  case x if x > 10 => nombreAbsences match {
    case y if y > 5 => println("Vous n'avez pas la mention ")
    case y if y < 5 => println("Vous n'avez pas la mention Abien")
  }
}

// Exiting paste mode, now interpreting.

Vous n'avez pas la mention Abien
moyenne: Int = 12
mention: String = Abien
nombreAbsences: Int = 4

scala> 
```

- Les expressions régulières sont des chaînes de caractères qui peuvent être utilisées pour trouver des motifs (ou leur absence) dans des données. En Scala toute chaîne peut être convertie en expression régulière à l'aide de la méthode `.r`. Nous en avons un exemple comme le montre l'image ci-dessous:



```
scala> :paste
// Entering paste mode (ctrl-D to finish)

import scala.util.matching.Regex

val numberPattern: Regex = "[0-9]".r

numberPattern.findFirstMatchIn("awesomepassword") match {
  case Some(_) => println("Password OK")
  case None => println("Password must contain a number")
}

// Exiting paste mode, now interpreting.

Password must contain a number
import scala.util.matching.Regex
numberPattern: scala.util.matching.Regex = [0-9]

scala> 
```