

- Création de variable de type Int et String et voir les différentes fonctions associées à ces deux types à travers les variables créées



```

(base) momo@momo:~$ scala
Welcome to Scala 2.11.12 (OpenJDK 64-Bit Server VM, Java 11.0.16).
Type in expressions for evaluation. Or try :help.

scala> val nombre = 6
nombre: Int = 6

scala> nombre.
|= + << >> abs compareTo getClass isNaN isValidChar isWhole round to toDegrees toInt toShort underlying
% - <= <= byteValue intValue isValidInt isNegInfinity toBinaryString toDouble toLong toLong unary_+ until
& / == >>> ceil floatValue isFinite isPosInfinity toOctalString toFloat toOctalString unary_- |
* < > ^ compare floor isInfinity isValidByte isValidShort min signum toByte toChar toHexString toRadians unary_~

scala> nombre.

```

```
scala> val chaine = "machaine"
chaine: String = machaine

scala> chaine.
*      capitalize      contentEquals      flatten      indexOfSlice      lengthCompare      patch      reverse      split      takeWhile      toSet
+      charAt           copyToArray      fold          indexOfWhere      lift              permutations    reverseIterator to          to
++     chars           copyToBuffer     foldLeft      indices           lines            prefixLength    reverseMap      toArray      toShort
++:    codePointAt      corresponds      foldRight     init             lines            product         runWith        toBoolean   toStream
+:     codePointBefore count           forall        ints             linesWithSeparators r              sameElements   toBuffer    toString
/:     codePointCount  diff            foreach       intern           map              scan           strip          toByte      toTraversable
+:     codePoints      distinct        format        intersect        matches          scanLeft       stripLeading    toCharArray toUpperCase
:\     collect         drop            formatLocal   isEmpty          max              scanRight      stripLineEnd toDouble    toVector
<     collectFirst    dropRight       genericBuilder isBlank          maxBy           segmentLength  stripMargin   toFloat     transpose
<=    combinations    dropWhile      getBytes      isEmptyOf         min            self           stripPrefix  toFloat     trim
>     companion       endsWith       getChars      isTraversableAgain minBy           seq            stripSuffix   toIndexedSeq union
>=    compare          equals          groupBy       iterator          mkString        size           stripTrailing toInt        unzip
addString compareTo     equalsIgnoreCase grouped      last            nonEmpty        slice          subSequence   toIterable  unzip3
aggregate compareToIgnoreCase exists        lastIndexOf    nonEmpty        sliding         substring      toIterator   updated
apply  compose         filter         lastIndexOfOf  offsetByCodePoints replace         sum            toList       view
applyOrElse concat      filterNot     lastIndexOfWhere replaceAll       sortBy         toLong       withFilter
canEqual contains      find          lastOption     replaceAllLiterally sortWith       toLowerCase  zip
      containsSlice flatMap      head          par             replaceFirst    sorted       toMap        zipAll
      flatMap      indexOf      headOption     repr            span            takeRight    toSeq        zipWithIndex

scala> chaine.[]
```

- Voici un exemple d'utilisation des options :save et :save où nous sauvegardons la session dans un fichier texte pour ensuite charger le fichier:

```
scala> :save test.txt

scala> :load test.txt
Loading test.txt...
nombre: Int = 6
chaîne: String = machaine

scala> 
```

- Nous retrouvons les mêmes commandes disponibles sur spark-shell que sur le shell de scala