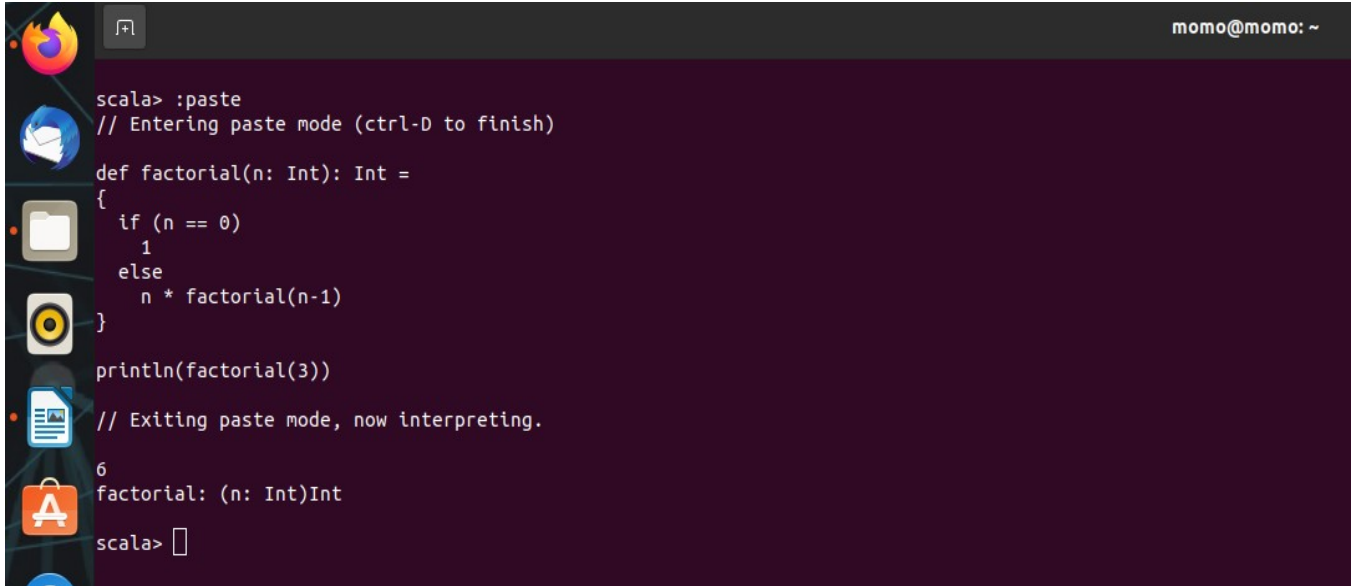


- En Scala la différence entre une fonction et une méthode est qu'une fonction est un objet qui peut être stocké dans une variable alors qu'une méthode appartient toujours à une classe.
- Nous pouvons prendre l'exemple de calcul du factoriel d'un nombre avec la méthode recursive comme le montre l'image ci-dessous:



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

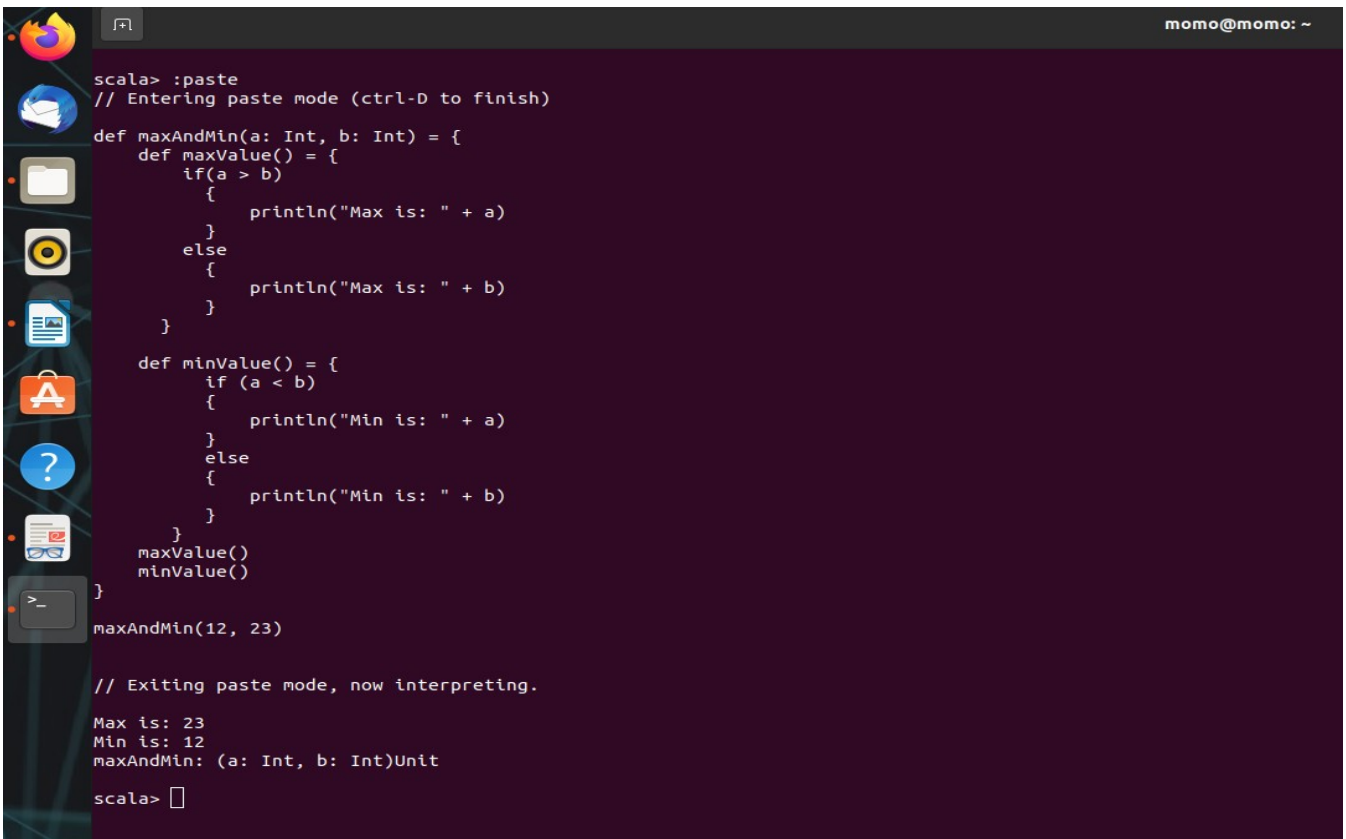
def factorial(n: Int): Int =
{
  if (n == 0)
    1
  else
    n * factorial(n-1)
}

println(factorial(3))

// Exiting paste mode, now interpreting.

6
factorial: (n: Int)Int
scala> 
```

- Exemple d'utilisation de fonctions dans une fonction



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

def maxAndMin(a: Int, b: Int) = {
  def maxVal() = {
    if (a > b)
    {
      println("Max is: " + a)
    }
    else
    {
      println("Max is: " + b)
    }
  }

  def minVal() = {
    if (a < b)
    {
      println("Min is: " + a)
    }
    else
    {
      println("Min is: " + b)
    }
  }
  maxVal()
  minVal()
}

maxAndMin(12, 23)

// Exiting paste mode, now interpreting.

Max is: 23
Min is: 12
maxAndMin: (a: Int, b: Int)Unit
scala> 
```

- Scala utilise exclusivement le passage par valeur, sauf que la valeur est soit une primitive, soit un pointeur vers un objet. Cette méthode est souvent plus efficace que l'appel par nom car il évite le re-calcul répété des expressions d'arguments que l'appel par nom implique.
- Lors de la définition d'une fonction si on constate qu'on a plus une tâche à réaliser, nous pouvons découper cette fonction en plusieurs fonctions de sorte que chaque fonction aura une et une seule tâche à exécuter. Il peut arriver aussi qu'on ait besoin d'appeler une fonction dans une autre fonction si nécessaire.