

List class:

Useful methods

- **head:** This method returns the first element of the scala list.
- **tail:** This method returns all the elements except the first.
- **isEmpty:** This method checks if the list is empty in which case it returns True else False.
- **fill():** This method creates a uniform list consisting of zero or more copies of the specified element.
- **reverse:** This method reverses the element of the list.
- **::** or **:::** or **.concat():** Concatenates two or more lists.
- **tabulate():** This method can tabulate a list. It's similar in spirit to fill() method, but unlike the fill() method, it takes two arguments. The first argument gives the dimensions of the list to create while the second describes the elements of the list, which are computed from a function.
- **::** Add an element x at the beginning of this list.

Example:

```
1 :: List(2, 3) = List(2, 3) :: (1) = List(1, 2, 3)
```

Example of task:

To recursively sum all elements of the list

```
def sum(xs: List[Int]): Int = {
  xs match {
    case x :: tail => x + sum(tail) // if there is an element, add it to
the sum of the tail
    case Nil => 0 // if there are no elements, then the sum is 0
  }
}

val l1 = List(1, 2, 3, 4, 5, 7)
sum(l1)
```