Scala Questions

Consider the following Scala List:

```
val list = List("This", "is", "a", "Scala", "List")
```

1. Give and alternative way to create this list using the "cons" operator (`::`)

2. What is the value of the following expressions:

```
list.isEmpty
list.head
list.tail
list.init
list.last
list.length
list.drop(2)
list.filter(s => s.length == 4)
list.map(s => s.length)
list.mkString("[", ":", "]")
list.exists(s => s.length > 5)
list.foreach(s = print(s))
```

3. Decide whether the following statements are True or False.

   - A pure functional language does not allow mutable objects
   - A pure functional language does not allow re-assignment to variables
   - A pure functional language does not allow iteration (while/for loops)
   - A pure functional language does not allow functions as parameters
   - All (functional) list operations can be implemented using recursion and the base list operations isEmpty, head, and tail
   - Prepending an element to the beginning of a list takes constant time
   - Appending an element to the end of a list takes constant time

4. Implement a *length* function for a Scala List that returns the length of the list. Use recursion, if-else statements, and list operations isEmpty, head, and tail.

5. Implement the same *length* function as above, but this time use matching instead of if-else and the base list operations.

6. Implement a *last* function for a Scala List that returns the last element in the list. Use the starter code below and just replace the ellipses (`...`) with the necessary code.

```
def last[T](list: List[T]): T = list match {
     case Nil => ...
     case x :: Nil => ...
     case _ :: xs => ...
}
```

7. Answer the following questions about the last function above.

- Which case statement is matched if you pass in `List("x", "y", "z")`?
- Which case statement is matched if you pass in `List("z")`?
- Which case statement is matched if you pass in `List()`?
- What is the type of `Nil`?
- What is the type of `x`?
- What is the type of `xs`?