Scala Stream

Stream is a lazy list. It evaluates elements only when they are required. This is a feature of scala. Scala supports lazy computation. It increases performance of your program.

Scala Stream Example

In the following program, we have created a stream.

```
object MainObject{

   def main(args:Array[String]){

      val stream = 100 #:: 200 #:: 85 #:: Stream.empty

      println(stream)

   }

}
```

Output:

```
Stream(100, ?)
```

In the output, you can see that second element is not evaluated. Here, a question mark is displayed in place of element. Scala does not evaluate list until it is required.

Scala Stream Example: Applying Predefined Methods

In the following example, we have used some predefined methods like toStream, which is used to iterate stream elements.

```
import scala.collection.immutable._

object MainObject{

   def main(args:Array[String]){

      var stream = 100 #:: 200 #:: 85 #:: Stream.empty

      println(stream)

      var stream2 = (1 to 10).toStream

      println(stream2)
```

```
        var firstElement = stream2.head

        println(firstElement)

        println(stream2.take(10))

        println(stream.map{_*2})

    }

}
```

Output:

```
Stream(100, ?)

Stream(1, ?)

1

Stream(1, ?)

Stream(200, ?)
```