Scala BitSet

Bitsets are sets of non-negative integers which are represented as variable-size arrays of bits packed into 64-bit words. The memory footprint of a bitset is determined by the largest number stored in it. It extends Set trait.

Scala BitSet Example

```
import scala.collection.immutable._

object MainObject{

   def main(args:Array[String]){

      var numbers = BitSet(1,5,8,6,9,0)

      numbers.foreach((element:Int) => println(element))

   }

}
```

Output:

```
0

1

5

6

8

9
```

Scala BitSet Example: Adding and Removing Elements

You can perform basic operations like adding and deleting in the bitset. In the following example, we have applied these operations.

```
import scala.collection.immutable._

object MainObject{

   def main(args:Array[String]){

      var numbers = BitSet(1,5,8,6,9,0)
```

```
        numbers.foreach((element:Int) => print(element+" "))
        numbers += 20               // Adding an element
        print("\nAfter adding 20: ")
        numbers.foreach((element:Int) => print(element+" "))
        numbers-=0              // Deleting an element
        print("\nAfter deleting 0: ")
        numbers.foreach((element:Int) => print(element+" "))
    }
}
```

Output:

```
0 1 5 6 8 9
After adding 20: 0 1 5 6 8 9 20
After deleting 0: 1 5 6 8 9 20
```