Kotlin Array

Array is collection of similar data types either of Int, String etc.

Array in Kotlinis mutable in nature with fixed size which means we can perform both read and write operations on elements of array.

Constructor of array:

Array constructor is declared with specified size and init function.

The init function is used to returns the elements of array with their index.

```
Array(size: Int, init: (Int) -> T)
```

Kotlin Array can be created using

- arrayOf(),
- intArrayOf(),
- charArrayOf(),
- booleanArrayOf(),
- longArrayOf(),
- shortArrayOf(),
- byteArrayOf() functions.

Kotlin array declaration - using arrayOf() function

```
var myArray1 = arrayOf(1,10,4,6,15)
var myArray2 = arrayOf<Int>(1,10,4,6,15)
val myArray3 = arrayOf<String>("Ajay","Prakesh","Michel","John","Sumit")
var myArray4= arrayOf(1,10,4, "Ajay","Prakesh")
```

Kotlin array declaration - using intArrayOf() function

```
var myArray5: IntArray = intArrayOf(5,10,20,12,15)
```

Modify and access elements of array

Kotlin has set() and get() functions that can direct modify and access the particular element of array respectively.

The set() function is used to set element at particular index location. This is also done with assigning element at array index.

Array get() function is used to get element from specified index.

Kotlin array set() function example

```
fun main(args: Array<String>) {

val array1 = arrayOf(1,2,3,4)
val array2 = arrayOf<Long>(11,12,13,14)
    array1.set(0,5)
    array1[2] = 6
    array2.set(2,10)
    array2[3] = 8
    for(element in array1){
          println(element)
    }
    println()
    for(element in array2){
          println(element)
    }
}
```

Kotlin array get() function example

```
fun main(args: Array<String>) {
     val array1 = arrayOf(1,2,3,4)
     val array2 = arrayOf<Long>(11,12,13,14)
     println(array1.get(0))
     println(array1[2])
     println()
     println(array2.get(2))
     println(array2[3])
}
```

Kotlin Array Example 1:

In this example, we are simply initialize an array of size 5 with default value as 0 and traverse its elements. The index value of array starts from 0. First element of array is placed at index value 0 and last element at one less than the size of array.

```
fun main(args: Array<String>){

     var myArray = Array<Int>(5){0}
     for(element in myArray){
          println(element)
     }
}
```

Kotlin Array Example 2:

We can also rewrite the value of array using its index value. Since, we can able to modify the value of array, so array is called as **mutable** property.

For example:

```
fun main(args: Array<String>){
var myArray = Array<Int>(5){0}

myArray[1]= 10
myArray[3]= 15

    for(element in myArray){
          println(element)
    }
}
```

Kotlin Array Example 3

- using arrayOf() and

- intArrayOf() function:

```
fun main(args: Array<String>){

val name = arrayOf<String>(
          "Ajay","Prakesh","Michel","John","Sumit")
var myArray2 = arrayOf<Int>(1,10,4,6,15)
var myArray3 = arrayOf(5,10,20,12,15)
var myArray4= arrayOf(1,10,4, "Ajay","Prakesh")
var myArray5: IntArray = intArrayOf(5,10,20,12,15)

     for(element in name){
          println(element)
     }
     println()
     for(element in myArray2){
          println(element)
     }
     println()
     for(element in myArray3){
          println(element)
     }
     println()
     for(element in myArray4){
          println(element)
     }
     println()
     for(element in myArray5){
          println(element)
     }
}
```

Kotlin Array Example 4

Suppose when we try to insert an element at index position greater than array size then what will happen?

It will throw an *ArrayIndexOutOfBoundException*.

This is because the index value is not present at which we tried to insert element.

Due to this, array is called fixed size length.

For example:

```
fun main(args: Array<String>){

var myArray5: IntArray = intArrayOf(5,10,20,12,15)
    myArray5[6]=18 // ArrayIndexOutOfBoundsException
    for(element in myArray5){
          println(element)
    }
}
```

Output:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 6
    at ArrayListKt.main(Array.kt:4)
```

Kotlin Array Example 5 - traversing using range:

The Kotlin's array elements are also traversed using index range (minValue..maxValue) or (minValue..maxValue).

Let's see an example of array traversing using range.

```
fun main(args: Array<String>){
var myArray5: IntArray = intArrayOf(5,10,20,12,15)
    for (index in 0..4){
         println(myArray5[index])
    }
    println()
    for (index in 0..myArray5.size-1){
         println(myArray5[index])
   }
}
```