

v1.1.2

Colecciones



Kotlin

Arrays

Características de los arrays

- Se representan con la clase `Array`
- Son accesibles mediante `[]`
- Disponen de la propiedad `size`

Arrays

```
val datos = arrayOf(1, 2, 3)
```

```
datos[1] = 5
```

```
println(datos[2])
```

```
datos[3] = 0 // excepción
```

Arrays de tipos primitivos

```
val x: IntArray = intArrayOf(1, 2, 3)
```

```
x[0] = x[1] + x[2]
```

Crear arrays con función factoría

```
// Creates an Array<String> with values ["0", "1", "4", "9", "16"]  
val asc = Array(5, { i -> (i * i).toString() })
```

```
val asc = Array(5) { i -> (i * i).toString() }
```

Crear arrays con función factoría

```
val datos = Array<Int>(4) {0}
```

Matrices y arrays multidimensionales

```
var m = 1

val a = Array<Array<Int>>(2) {
    Array<Int>(2) { m++ }
}

for (i in 0..1) {
    for (j in 0..1) {
        print("%2d".format(a[i][j]))
    }
    println()
}
```

1 2
3 4

Listas, conjuntos y diccionarios

Listas, conjuntos y diccionarios

- Kotlin distingue entre colecciones mutables y no mutables
- Además, podemos tener una vista de solo lectura de una colección mutable

	Inmutable	Mutable
Lista	<code>List<out T></code>	<code>MutableList<T></code>
Conjunto	<code>Set<out T></code>	<code>MutableSet<T></code>
Diccionario	<code>Map<K, out V></code>	<code>MutableMap<K, V></code>

Listas

```
val numbers: MutableList<Int> = mutableListOf(1, 2, 3)
```

```
val readOnlyView: List<Int> = numbers
```

```
println(numbers)           // prints "[1, 2, 3]"
```

```
numbers.add(4)
```

```
println(readOnlyView)      // prints "[1, 2, 3, 4]"
```

```
readOnlyView.clear()       // -> does not compile
```

Operaciones sobre listas

```
val items = listOf(1, 2, 3, 4)
items.first() == 1
items.last() == 4
items.filter { it % 2 == 0 }    // returns [2, 4]

val rwList = mutableListOf(1, 2, 3)
rwList.requireNonNulls()        // returns [1, 2, 3]
if (rwList.none { it > 6 }) println("No items above 6")
val item = rwList.firstOrNull()
```

Conjuntos

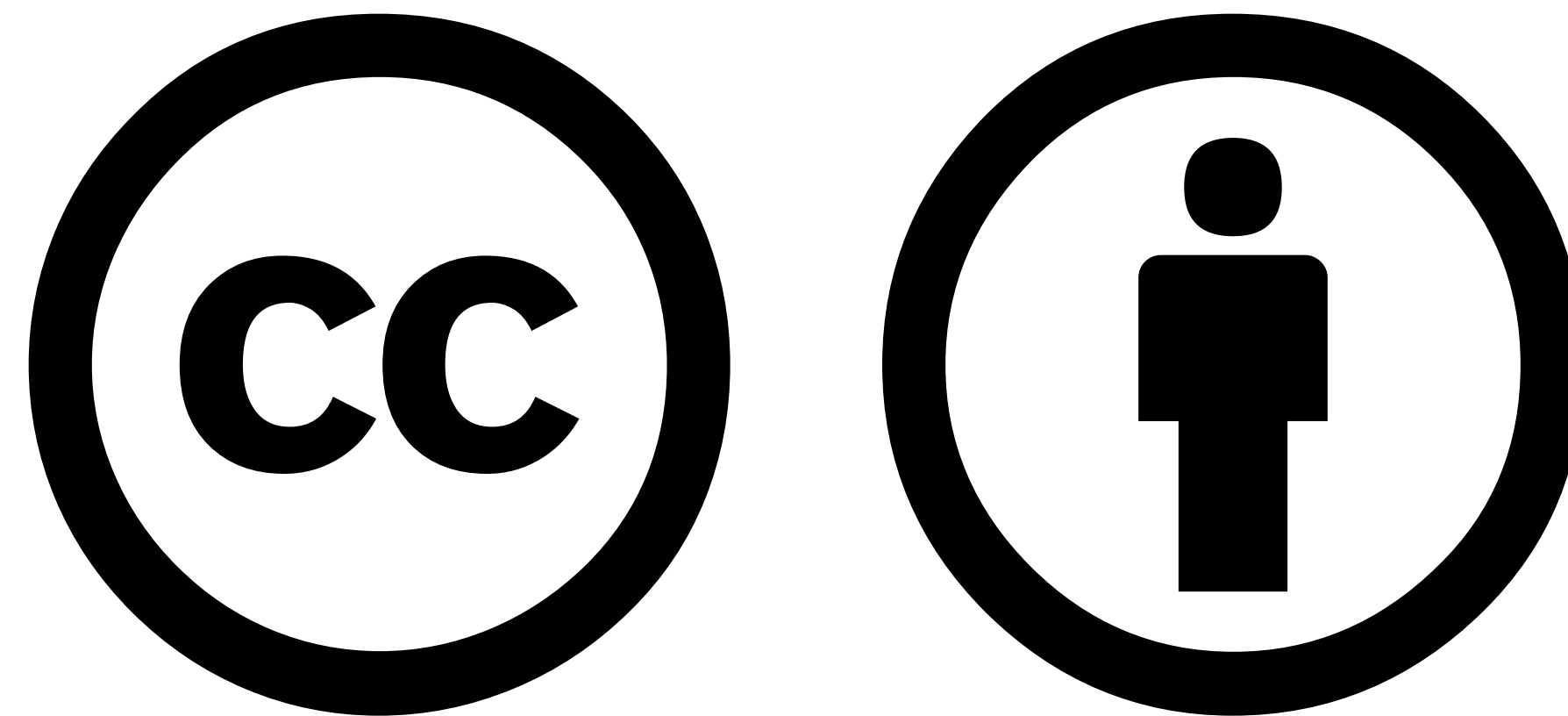
```
val strings = hashSetOf("a", "b", "c", "c")  
println(strings.size == 3)
```

Diccionarios

```
val readWriteMap = hashMapOf("foo" to 1, "bar" to 2)

println(readWriteMap["foo"]) // prints "1"

val snapshot: Map<String, Int> = HashMap(readWriteMap)
```



Excepto si se especifica lo contrario, esta presentación está bajo licencia

<https://creativecommons.org/licenses/by/4.0/>

© 2017 Ion Jaureguialzo Sarasola. Algunos derechos reservados.