



M3 - Programació

Tipus complexos I - Arrays



Index

- Arrays

Array - Definició



- Els **arrays** s'utilitzen per emmagatzemar diversos valors en una sola variable, en lloc de crear variables separades per a cada valor.
- Els **arrays** s'usen per organitzar les dades en la programació de manera que un conjunt relacionat de valors **es puguí ordenar o cercar fàcilment**.

Array - Definició



- Aquí hi ha algunes **propietats bàsiques** dels arrays:
 - S'emmagatzemen en ubicacions de memòria contigües.
 - S'hi pot accedir amb programació mitjançant els seus índexs (array[1], array[0], etc.)
 - Són mutables.
 - La seva mida és fixa.

Array - Creació



- Per crear un array, utilitzeu la funció **arrayOf()** i col·loqueu-hi els valors en una llista separada per comes:

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")
```



Array - Accés a un element

- Podeu accedir a un element de matriu fent referència al número d'índex, entre claudàtors.
- En aquest exemple, accedim al valor del primer element dels cotxes:

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")  
println(cars[0])  
// Imprimeix Volvo
```

Array - Accés a un element



- L'índex del primer element d'un array **és sempre 0**, així doncs, els índexs d'un array complet va sempre de **0...n-1** on **n és a mida de l'array**.
- Si ens passem i intentem accedir a un índex que no existeix, obtindrem un error.



```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")
println(cars[4])
// ERROR
```

Array - Modificar un element



- Per canviar el valor d'un element específic, utilitzeu el número d'índex i reassigneu el valor d'aquella posició:

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")  
cars[0] = "Audi"  
println(cars[0])  
// Imprimeix Audi
```


Array - Modificar un element



- També es pot utilitzar la funció **set()** que ve incorporada a la classe Array:

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")
cars.set(0, "Audi")
println(cars[0])
// Imprimeix Audi
```

Array - Saber la mida



- La classe array incorpora una propietat **size** que et **retorna la mida de l'array**, és a dir, el nombre d'elements que conté aquest.

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")  
println(cars.size)  
// Imprimeix 4
```

Array - Saber la mida



How language creators choose the function to get the size of an array

u/dark-trooper



Array - Conté un element?



- Per saber si un determinat valor està dins d'un array, podem usar la paraula clau **in**, que ja usem en els rangs, **per determinar si aquest element està dins de l'array o no**.

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")  
if ("Volvo" in cars) println("It exists!")  
else println("It does not exist.")  
// Imprimirà existeix si Volvo està dins de l'array cars
```



Array - Iterar

- Sovint, quan treballem amb arrays, heu de recórrer tots els elements.
- Podeu recórrer els elements d'un array amb el bucle for.
- L'exemple següent imprimeix tots els elements de l'array cars:

```
val cars = arrayOf("Volvo", "BMW", "Ford", "Mazda")  
for (x in cars) {  
    println(x)  
}
```

Array - Paràmetre d'entrada



- Ara podem analitzar aquell objecte que ens arribava a la funció main i que inicialment vam descartar.
- Què és això? Doncs ni més ni menys és un array d'Strings que ens ve amb els paràmetres d'entrada del programa (si en té).

```
fun main(args: Array<String>) {  
    for (arg in args) {  
        println(arg)  
    }  
}
```

Array - Paràmetre d'entrada



- Com podem enviar paràmetres d'entrada?

