

**Taller Estructuras de Datos en Kotlin**

El objetivo de este taller es que los aprendices sean capaces de comprender y utilizar las principales estructuras de datos en Kotlin, incluyendo arreglos, listas, conjuntos, mapas y pares.

El aprendiz deberá realizar un informe donde se evidencian los siguientes puntos:

1. Introducción a las estructuras de datos en Kotlin
   1. ¿Qué son las estructuras de datos y para qué se utilizan?

R/las estructuras de datos son varias formas de agrupar datos de manera más sencilla

Y óptima ,se pueden usar para almacenar y organizar de manera eficiente y nos facilita

Acceder y manipular estos mismos datos.

* 1. Ventajas de utilizar estructuras de datos en Kotlin

R/ Facilidad de uso: nos proporciona una sintaxis simple para crear estructuras

De datos y nos permite crear estructuras de datos de una manera más fácil de

Entenderla...

Rendimiento Óptimo: al usar estructuras de datos óptimas en kotlin nos asegura

De que kotlin se ejecute de manera más eficiente y sin errores...

Flexibilidad: kotlin nos ofrece varias opciones donde en el momento de crear

Estructuras de datos nos dice o asesora de cual es la mejor estructura para nuestra

Aplicación....

Seguridad:Si la aplicación se ejecuta en entornos diferentes kotlin nos asegura

De mantener nuestros datos protegidos y seguros.

* 1. Diferencias entre las estructuras de datos en Kotlin y Java

R/ 1. En kotlin ya no usamos switch case como lo hacemos en java si no que usamos

El when

2.en kotlin podemos crear estructuras de datos basadas en funciones de esta manera

Se nos facilita el uso de las mismas.

3.kotlin nos permite definir los conjuntos y acceder a ellos de manera individual

mientras que en java no se nos hace posible.

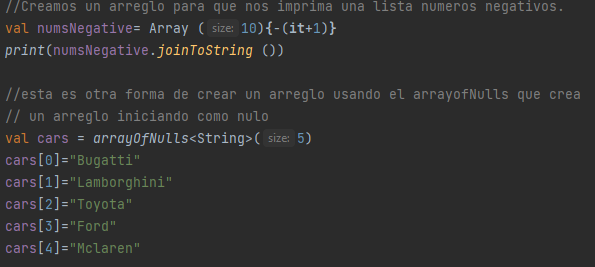
1. Arreglos en Kotlin
   1. ¿Qué es un arreglo?

R/es una estructura de datos que nos permite almacenar elementos de una forma

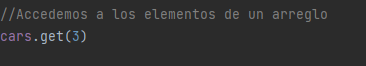
Ordenada y podremos acceder a ellos a través de un indice, y nos facilita operar con ellos

Ya que se almacenan de memoria continua.

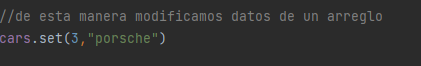
* 1. Creación de arreglos en Kotlin



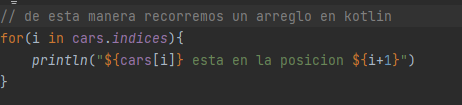
* 1. Accediendo a los elementos de un arreglo



* 1. Modificando los elementos de un arreglo



* 1. Recorriendo un arreglo



* 1. Funciones útiles para trabajar con arreglos en Kotlin

R/las funciones que nos permiten trabajar con arreglos son set y get :

set() sirve para asignar el valor de un índice:

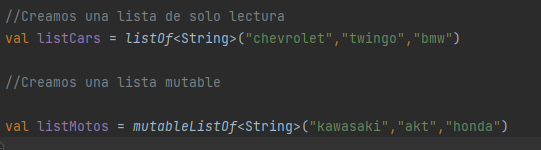
1. get() sirve para obtener el valor de dicho índice:
2. Listas en Kotlin
   1. ¿Qué es una lista?

R/una lista es una colección de elementos que se puede acceder a través de un

Índice.

* 1. Creación de listas en Kotlin

R/



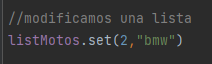
* 1. Accediendo a los elementos de una lista

R/



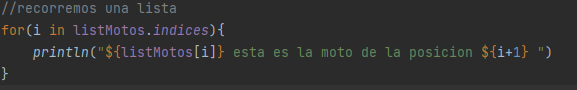
* 1. Modificando los elementos de una lista

R/



* 1. Recorriendo una lista

R/



* 1. Funciones útiles para trabajar con listas en Kotlin

add() // Agregamos algo a la lista

addAll() // Agregamos una lista entera a nuestra lista

remove() // Quitamos un elemento de una lista

removeAt() // Quitamos un elemento a través de un índice

clear() // Eliminamos todo lo de la lista

isEmpty() // Revisamos si la lista está vacía

first() // Muestra el primer elemento de la lista

last() // Muestra el último elemento de la lista

contains() // Muestra una lista nueva en la que los elementos se encuentran en orden inverso

reversed() // Muestra otra en la que los elementos están ordenados en forma ascendente

4.

Conjuntos en Kotlin

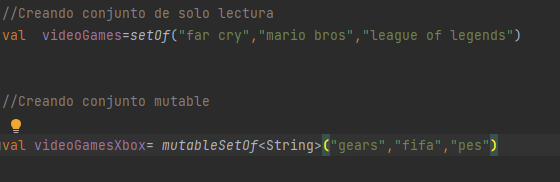
* 1. ¿Qué es un conjunto?

R/es una colección de elementos no ordenada y que son únicos sin repetición

Y son útiles cuando necesitamos crear una colección sin elementos duplicados

* 1. Creación de conjuntos en Kotlin

R/

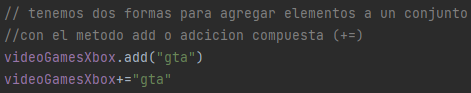


* 1. Accediendo a los elementos de un conjunto

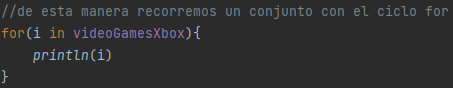
R/No se puede acceder al elemento de un conjunto,ya que sus diferentes elementos cuentan como una sola posicion.



* 1. Modificando los elementos de un conjunto



* 1. Recorriendo un conjunto



* 1. Funciones útiles para trabajar con conjuntos en Kotlin

add() // agregar un elemento en el conjunto

remove() // quitar un elemento del conjunto

unión() // toma como argumentos dos colecciones y retorna en un conjunto con todos los elementos que pertenezcan a ambas.

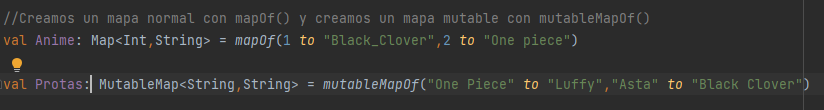
intersect() // aislar solo los elementos que estén presentes en dos colecciones

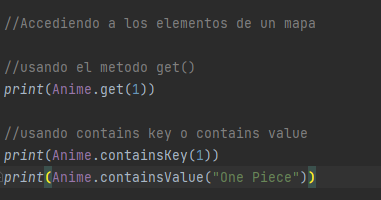
subtract() // calcular la diferencia entre dos colecciones

1. Mapas en Kotlin
   1. ¿Qué es un mapa?

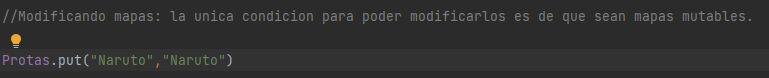
R/es una colección de datos pares que se definen en su llave y su valor

Y podemos recorrerla de manera ordenada según su llave

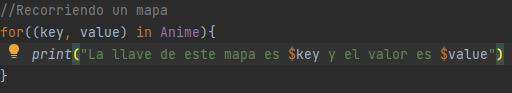
* 1. Creación de mapas en Kotlin 
  2. Accediendo a los elementos de un mapa



* 1. Modificando los elementos de un mapa



* 1. Recorriendo un mapa



* 1. Funciones útiles para trabajar con mapas en Kotlin

get(K) // obtiene el valor de la clave

isEmpty() // Revisa si el mapa está vacío

containsKey(K) // Revisa si existe la clave en el mapa (devuelve un boolean)

containsValue(V) // Revisa si el valor está en el mapa (devuelve un boolean)

keys // devuelve un Set inmutable con todas las claves en el mapa.

values // Collection inmutable de todos los valores en el mapa.

remove(K) // borra la clave y su valor enlazado. (Solo en mutables

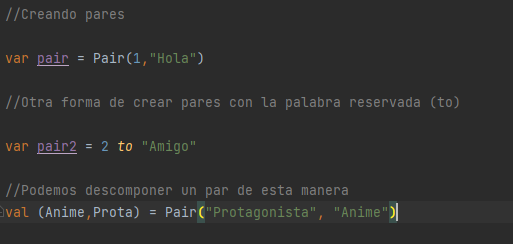
clear() // elimina todos los elementos del mapa.

1. Pares en Kotlin
   1. ¿Qué es un par?

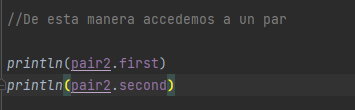
R/es una estructura de datos que tiene 2 valores uno a la izquierda

Y otro a la derecha.

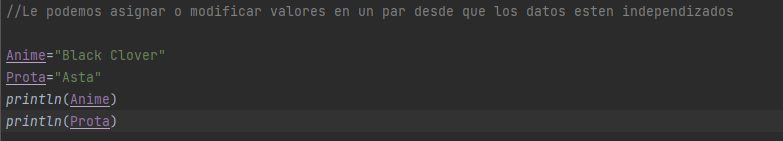
* 1. Creación de pares en Kotlin



* 1. Accediendo a los elementos de un par



* 1. Modificando los elementos de un par



* 1. Recorriendo un par
  2. Funciones útiles para trabajar con pares en Kotlin

1. Prácticas de estructuras de datos en Kotlin
   1. Ejercicios prácticos para aplicar los conceptos aprendidos

* Ejercicio que permita la suma de números enteros que estén en un arreglo
* Ejercicio que pida al usuario la cantidad de numeros a ingresar y haga su promedio

usando listas en kotlin

* Ejercicio donde use dos conjuntos de números y que en un tercer conjunto queden los números que

coinciden en las otras dos listas

* Ejercicio que guarde en un mapa diferentes animes y el usuario pueda seleccionarlo con su llave
* Ejercicio que le permita al usuario ingresar una cantidad de animales y ingresar su especie y saber que especie es, usando pares

Recursos adicionales:

Taller Estructuras de Datos en Kotlin 2