

INTRODUCCIÓN A KOTLIN

Kotlin es un lenguaje de programación multiplataforma, tipado, de alto nivel y de propósito general con inferencia de tipos. Está diseñado para ser totalmente interoperable con **Java**, hasta el punto que puedes traducir un código de un lenguaje a otro automáticamente en casi cualquier IDE. La versión JVM de la biblioteca estándar de Kotlin depende de la Java Class Library, pero la inferencia de tipos permite que su sintaxis sea más concisa. Kotlin se dirige principalmente a la JVM, pero también compila a JavaScript para aplicaciones web como React. Además, desde el lanzamiento de **Android Studio 3.0** en octubre de 2017, Kotlin se ha incluido como alternativa al compilador Java estándar.

... y con el paso del tiempo se ha hecho evidente que es más eficiente programar en Kotlin para Android que en Java.

Sabiendo Java, es posible aprender las bases de **Kotlin** con cierta facilidad. La sintaxis de Kotlin puede parecernos rara, pero en el fondo, es como si fuese Java resumido. Simplemente hay que tener en cuenta los dos o tres detalles que diferencian a ambos lenguajes, y practicar.

Vamos a aprender Kotlin como todo buen programador: mirando tutos en Internet.

<https://www.w3schools.com/KOTLIN/index.php>

No necesitas nada más que esta página y un IDE con Kotlin para hacer pruebas. Aconsejo el **IntelliJ IDEA** porque la forma que tiene de organizar los proyectos y el código se asemeja mucho a la de **Android Studio**. Basta con bajarse la versión de prueba, o si se desea, crearse una cuenta con licencia de estudiante (para tu uso personal).

Cosas que diferencian a Kotlin de Java

Kotlin no es 100% igual a Java, aunque puedes leer su código sin tener ni idea si conoces las cuatro cosillas siguientes. Te en cuenta que esto es un pequeño resumen...

Nullable / No-Nullable

Kotlin diferencia las variables que pueden y no pueden ser nulas. Todas las variables que definimos son siempre **no-nullables**, a menos que especifiquemos lo contrario. Es decir, que no pueden ser nunca nulas. Si quieres que sean nulas, tienes que usar **“?”**. Esto también se utiliza para hacer comprobaciones de nulo, etc.

Var y Val

Se utiliza **var** y **val** a la hora de declarar variables. La diferencia entre ellas es que **var** permite que el valor de la variable cambie, mientras que las declaradas con **val** no pueden cambiar. Pero cuidado, que no es lo mismo que static de Java. Kotlin no usa variables estáticas.

```
val name: String = "John"  
val birth = 1975
```

Tipos de datos

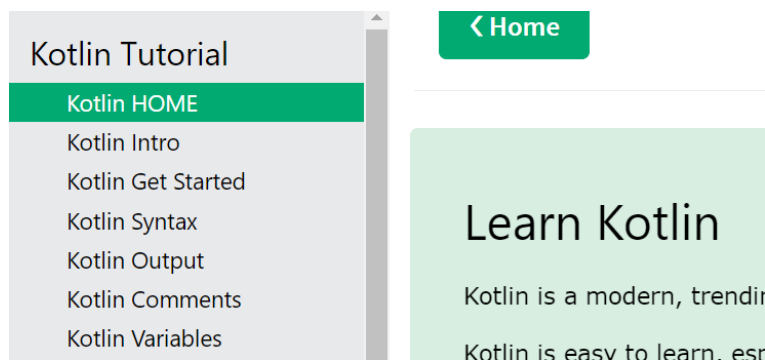
Kotlin es capaz de adivinar por sí mismo el tipo de datos de las variables. Esto permite resumir bastante el código. Es decir, no necesitas especificar absolutamente todo como en Java. Mira por ejemplo la imagen de arriba. No le estamos diciendo en ningún momento el tipo de dato de birth, pero el IDE no te da error. Simplemente ‘adivina’ que es un String. Cuidado que este tipo de cosas, acostumbrado a Java, confunden un poco.

Definir clases, constructores, getters y setters, etc.

Esto es un jaleo. Kotlin permite hacer cosas realmente sorprendentes como definir en una sola línea de texto la clase y su constructor. Puede incluso que se definan clases vacías pero que en realidad tengan montones de movidas. Esto es mejor ir encontrándoselo. Por el momento, recuerda solamente que no hace falta hacer getters y setters en una clase. Kotlin ya asume que existen.

Nuestro Tutorial

Vamos a trabajar partiendo de la página mencionada más arriba. Vamos a ir punto por punto ejecutando los ejemplos que vienen en la propia página. Tienes código comentado a parte a modo de ejemplo



Bloque prácticas 1

Accede a la página del tutorial y revisa el primer bloque, desde **Kotlin Home** hasta **Kotlin Functions**. Estos puntos te resumen los elementos básicos del lenguaje de programación: variables, bucles, funciones...

A continuación, lee y ejecuta las clases **Main01.kt** hasta **Main07.kt** para información adicional y ver ejemplos funcionales. Prueba tú mismo a crear tus propios ejemplos.

Ejercicio 1

Crea un programa Kotlin que solicita un usuario y un password. Si ambos son 'admin', le mostramos un mensaje de bienvenida. Si no, le mostramos un mensaje de error.

Ejercicio 2

Lo mismo, pero le dejamos tres intentos.

Ejercicio 3

Lo mismo, pero en lugar de bienvenida, mostramos un menú de opciones:

- 1) Nuevo nombre
- 2) Mostrar nombres
- 3) Buscar nombre
- 4) Modificar nombre
- 5) Borrar nombre

Guardaremos los nombres en un array de tamaño 10.

Ejercicio 4

Lo mismo, pero el array tiene que estar siempre ordenado de forma ascendente por nombre. Por lo tanto, al insertar, borrar o modificar un nombre, será preciso reordenar el array.

Ejercicio 5

[Resuelto - Main08.kt] Realiza de nuevo el Reto 1 de la asignatura de Programación. El enunciado está detallado en un documento aparte.